

Rapport de stage
DEA Réseaux et Systèmes Distribués

Services de protection du contenu

Guillaume TAMBOISE

juin 01

Entreprise :	Bouygues Telecom
Responsable Entreprise :	Maël RAFFIN
Responsable Eurécom :	Prof. Refik MOLVA
Rapport confidentiel :	non

Communications d'Entreprise
Institut Eurécom

Remerciements

Je tiens en premier lieu à adresser mes plus vifs remerciements à Refik Molva pour son encadrement, son aide et sa présence dans toutes les phases du stage, de la bibliographie à la relecture de mes différents documents. Nul doute qu'il m'a fait grandement apprécier le monde de la recherche en réseaux, si des considérations autres ne m'avaient pas éloigné de l'idée d'une thèse c'est encadré par lui et par personne d'autre que j'aurais voulu poursuivre.

Je remercie Alain Pannetrat, avoir travaillé avec lui a grandement contribué à me forger une opinion très positive d'une thèse à Eurécom. Son approche pragmatique lors de la phase de réflexion sur l'architecture a été des plus bénéfiques, ses connaissances sur le sujet en ont fait un partenaire de discussion hors pair.

Je n'oublie pas Maël Raffin et sa coopération à distance. Les exigences du monde industriel qu'il incarnait m'ont permis de me remotiver à chaque baisse de régime et obligé à un effort de présentation bénéfique *a posteriori*, et dont je lui suis particulièrement reconnaissant.

Je tiens également à remercier tous les autres membres de NSTEAM, Guillaume Urvoy et les discussions que j'ai eues avec lui sur les thèses (il m'aurait presque fait changer d'avis !), Laurent Bussard (encore toutes mes félicitations) et sa documentation sur CPSA, Yves, Pietro et Stefano, les gens du département informatique de l'Eurécom, en particulier Jean-Christophe Delaye et Christophe Lonziano pour leur efficacité et leur disponibilité.

Table des matières

<i>Chapitre I - Introduction</i>	6
<i>Chapitre II - État de l'art</i>	7
1 Modèles	7
1.1 Besoins	7
1.2 Moyens	7
1.3 Modèles	7
2 Sécurité	8
3 Cryptographie	8
3.1 Mécanismes usuels	8
3.2 Algorithmes asymétriques	9
4 Le tatouage	10
4.1 Services possibles.....	10
4.2 Copie conditionnelle.....	10
5 Protocoles à base de tatouage	11
5.1 Protection d'une image par une signature externe.....	11
5.2 Intervention du TTP en ligne.....	12
5.3 Communication directe entre les deux entités	15
6 Architecture CPSA	18
6.1 SDMI.....	19
6.2 CPRM.....	19
6.3 CSS.....	21
6.4 DTCP.....	21
6.5 HDCP	23
7 Approches ouvertes	23
7.1 OPIMA	23
7.2 IETF	24
7.3 3GPP.....	24
7.4 MPEG.....	24
8 Mobile Music Forum	27
9 Synthèse	28
9.1 Approches.....	28
9.2 Comparaison des deux approches.....	28
<i>Chapitre III - Proposition</i>	30
1 Contexte	30
2 Formalisation	31
3 Cryptage multicast	31
3.1 Cryptage symétrique.....	31
3.2 Cryptage asymétrique	33
3.3 Application à un réseau GSM	34
<i>Chapitre IV - Article</i>	36
1 Introduction	36
1.1 Définitions	36
1.2 Scenario	36
2 Multicast access control	37
2.1 The Data Distribution Layer.....	38

2.2	The Key Distribution Layer.....	39
2.3	The Membership Management Layer.....	40
2.4	Entities and Trust.....	40
3	(CST) Cipher sequence Trees	41
3.1	Preliminary Definitions	41
3.2	Key Distribution Trees	42
3.3	Key Distribution in a Dynamic Group.....	44
3.4	Membership Management	45
3.5	Reliability and Trust.....	46
4	Content Distribution.....	47
4.1	Background	47
4.2	Framework.....	48
5	Mobility.....	50
5.1	Land Mobile Network	50
5.2	Key Distribution	50
5.3	Handover	52
	Chapitre V - Évaluation de performances	53
1	Modèle.....	53
1.1	Position du mobile.....	53
1.2	État du mobile	53
2	Probabilités stationnaires.....	55
2.1	Chaîne de Markov	55
2.2	Probabilités stationnaires.....	55
3	Résolution	56
4	Coût de mise à jour de l'appartenance au groupe	57
4.1	Mise à jour par le gestionnaire local.....	57
4.2	Mise à jour par le gestionnaire global.....	58
4.3	Perspectives	58
5	Traitement par renormalisation d'un modèle de percolation dirigée.....	59
5.1	Modèle de percolation	59
5.2	Traitement par renormalisation	63
	Chapitre VI - Conclusion.....	67
	Calculs d'Évaluation de Performances.....	68

Table des figures

<i>fig. 1.3.1 – Modèles opérationnels</i>	8
<i>fig. 5.2.1 – Protocole de protection d'une image par une signature externe</i>	13
<i>fig. 5.2.2 – Protocole à base de TTP</i>	14
<i>fig. 5.3.1 – Communication directe</i>	15
<i>fig. 6.2.1 – Content Protection for Recordable Media : écriture sur le DVD</i>	20
<i>fig. 6.2.2 – Content Protection for Recordable Media : lecture du DVD</i>	20
<i>fig. 7.4.1 – Gestion des événements dans MPEG-21</i>	26
<i>fig. 7.4.1 – Processus de diffusion de contenu présenté par le Mobile Music Forum</i>	27
<i>fig. 3.1.1 – Cryptage multicast à logarithme discret</i>	32
<i>fig. 3.3.1 – Diffusion multicast dans un réseau GSM</i>	34
<i>fig. 1.1.1 – Marche du mobile</i>	53
<i>fig. 1.2.1 – État du mobile</i>	54
<i>fig. 5.1.1 – Longueurs de corrélation parallèle et perpendiculaire</i>	62
<i>fig. 5.2.1 – Renormalisation d'un facteur 2</i>	63
<i>fig. 5.2.2 – Résolution du pont de Wheatstone</i>	65

Liste des tableaux

<i>tab. 3.1.1 – Services vs. mécanismes</i>	8
<i>tab. 3.2.1 – Chiffrement à clef publique</i>	9
<i>tab. 5.2.1 – Table des symboles</i>	14

Chapitre I - Introduction

Ce stage de DEA est le fruit de la coopération entre Bouygues Telecom Recherche et Développement et l'équipe NSTEAM de l'institut Eurécom.

Le service à valeur ajoutée qu'un opérateur fournirait dans le cadre de cette étude serait, au-delà de l'acheminement des données d'un fournisseur de contenu vers un utilisateur final, l'assurance du respect de la propriété intellectuelle vis-à-vis du fournisseur de contenu. Ceci dans une approche visant à

- garantir l'accès conditionnel aux données ;
- limiter la diffusion de données ne respectant pas la propriété intellectuelle de leurs auteurs.

Nous nous sommes placés dans le cadre d'un système de rendu des données multimédia relativement ouvert, avec une capacité de traitement en environnement sécurisé du côté de l'utilisateur équivalent à celle d'une carte à puce.

Les données concernées sont essentiellement du son, des images et de la vidéo. Il convient donc de tenir compte des spécificités de ces données (en terme de volume ou le cas échéant de temps réel). Nous avons tâché de prendre en compte les progrès récents en matière de tatouage, et de l'utiliser sous une forme générique à travers les services de sécurité qu'il peut rendre. Il a alors été nécessaire de déterminer ses limites, la pertinence de son utilisation comme seul garant d'un accès ou d'une copie conditionnelle ainsi que son apport (ou son inutilité) vis-à-vis des approches classiques de chiffrement.

Dans un premier temps, nous nous sommes attachés à définir différents modèles de références d'une architecture de protection du contenu puis à étudier les critères de conception qui ressortent de ces modèles, avant d'examiner leur mise en œuvre dans des architectures existantes. Nous ne nous sommes pas particulièrement limités aux approches favorisant la mobilité de l'utilisateur final mais en avons tenu compte dans les critères de conception applicables.

L'état de l'art nous permis de mettre en évidence un défaut de passage à l'échelle dans l'architecture la plus prometteuse pour un opérateur de télécommunications. Nous avons tenté de modifier cette approche pour gommer ce défaut. D'abord en formalisant d'une manière originale les relations entre les différents acteurs de distribution du contenu, ensuite en appliquant ce modèle dans la conception d'une architecture de diffusion par-dessus un réseau multicast existant.

Chapitre II - État de l'art

1 Modèles

1.1 Besoins

Les besoins auxquels doit répondre une architecture visant à protéger le contenu sont principalement :

- la gestion et la mise en œuvre d'un droit de copie modulable ;
- le respect d'une certaine *privacy* ;
- la possibilité d'agir en temps réel ;
- la gestion des périphériques dynamiquement ou par révocation.

1.2 Moyens

Les moyens à notre disposition sont

- la cryptographie,
- la stéganographie via le tatouage,
- les matériels sécurisés (TPH1),
- les tiers de confiance (TTP2).

1.3 Modèles

Les différents modèles qui répondent à ces besoins, compte tenu des moyens que l'on se donne sont représentés fig. 1.3.1. Nous développerons par la suite ces différents modèles à travers les architectures qui les ont adoptés :

- la première par CPSA, le distributeur de périphérique est une entité intervenant dans le schéma de distribution,
- la seconde par le *Mobile Music Forum*, le distributeur de contenu est présent au plus près de l'utilisateur *via* sa carte SIM3,
- la troisième et la quatrième mandatent un tiers de confiance dans le but d'ajouter des services de non-répudiation.

¹ *Tamper-Proof Hardware*, matériel sécurisé

² *Trusted Third Part*, tiers de confiance

³ *Subscriber Identity Module*, représentant de l'opérateur dans le mobile GSM

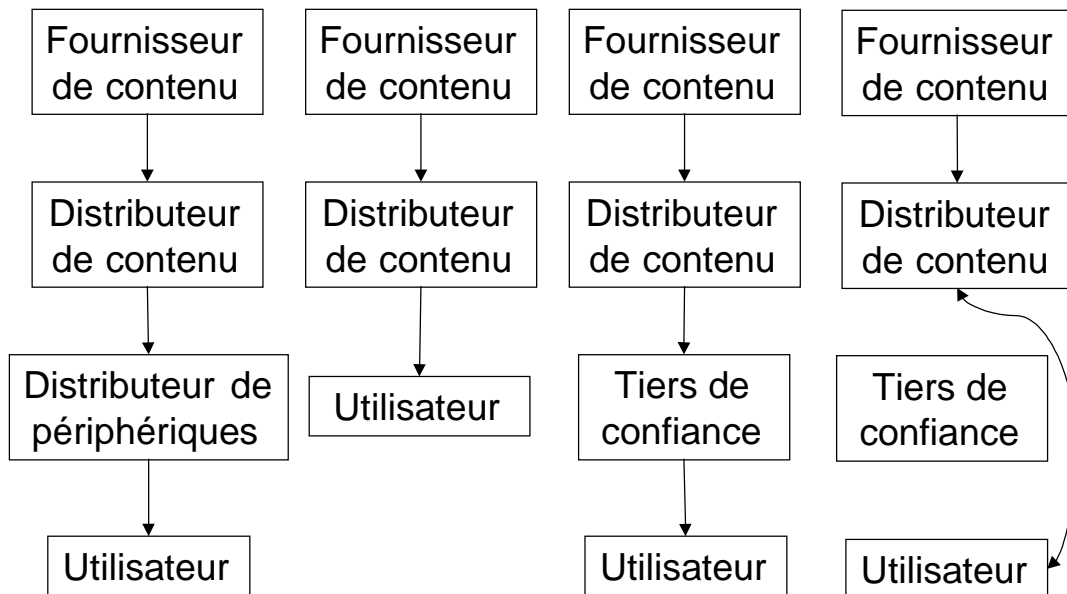


fig. 1.3.1 – Modèles opérationnels

2 Sécurité

Les services de sécurité usuels répondent aux questions suivantes :

- authentification : mon interlocuteur est-il bien celui qu’il prétend être ?
- contrôle d’accès : a-t-il bien le droit d’accéder aux données qu’il réclame ?
- confidentialité : suis-je assuré qu’aucune tierce partie écoutant notre conversation ne puisse rien en déduire sur son contenu ?

3 Cryptographie

3.1 Mécanismes usuels

Le tab. 3.1.1 donne une vue d’ensemble des services de sécurité que peuvent apporter les mécanismes de chiffrement usuels [I]. Pour envoyer un message confidentiel avec accusé de réception, il convient ainsi de combiner les mécanismes cités.

Service	cryptage	Hachage simple	hachage avec clef	signature numérique
confidentialité	✓			
intégrité		✓	✓	✓
authentification			✓	✓
preuve d’origine				✓

tab. 3.1.1 – Services vs. mécanismes

Notons qu'aucune combinaison de ces mécanismes ne fournit le service de non-répudiation d'origine : il est nécessaire d'introduire la notion de tiers de confiance.

3.2 Algorithmes asymétriques

Le tab. 3.2.1 donne les éléments clefs du chiffrement par les algorithmes RSA et El Gamal.

L'algorithme de chiffrement de El Gamal repose sur la difficulté de calculer le logarithme discret. Pour accéder au message x à partir de $x\beta^k \bmod p$, un attaquant doit accéder à k qu'il ne peut obtenir que via $y \equiv \alpha^k \bmod p$. Chercher un tel k revient alors à calculer $\ln_\alpha(y)$, ce qui est connu comme étant un problème « difficile ». L'algorithme de chiffrement RSA repose lui sur la difficulté de factoriser de grands nombres. En effet, une formulation du théorème d'Euler-Fermat (conséquence directe formulée par Euler du petit théorème de Fermat) est que si on travaille dans un groupe cyclique modulo n produit de nombres premiers distincts, les exposants dans les calculs dans ce groupe se réduisent modulo $\varphi(n)$ ($\varphi(n)$ est l'indicatrice d'Euler, le nombre de nombres premiers avec n et plus petits que n), ce qui se formule « si $r \equiv s \bmod \varphi(n)$, alors $a^r \equiv a^s \bmod n$ pour tous les entiers a ». Ce résultat reste valable si n est premier [II]. Avec pour applications directes :

- dans El Gamal, quiconque ne possédant aucun secret est capable de travailler non seulement
 - *en bas*, c'est à dire par exemple effectuer des produits car il le fait modulo p qui est public, mais aussi
 - *en haut* c'est à dire au niveau de l'exponentiation car il doit le faire modulo $\varphi(p)$ qui est connu et vaut $p-1$ car p est premier ;
- dans RSA, tout le monde peut travailler en bas mais pour travailler en haut, c'est à dire effectuer correctement l'exponentiation pour inverser e modulo $\varphi(n)$, il faut connaître $\varphi(n)$ qui ne l'est que de celui qui a formé $n = pq$ (et vaut $\varphi(p)\varphi(q) = (p-1)(q-1)$).

Algorithme	Cryptage	Décryptage	Fondements	Public	Privé
RSA	$x^e \bmod n$	$x^{en} \bmod n$	$ed \equiv 1 \bmod \varphi(n)$ $n = pq$ $\varphi(n) = (p-1)(q-1)$	$\{e, n\}$	$\{d, p, q\}$
El Gamal	$\alpha^k \bmod p$ $x\beta^k \bmod p$	$x\beta^k (\alpha^{ka})^{-1} \bmod p$	$\beta \equiv \alpha^a \bmod p$	$\{\alpha, \beta, p\}$	$\{k, a\}$

tab. 3.2.1 – Chiffrement à clef publique

4 Le tatouage

4.1 Services possibles

Il convient de s'interroger sur l'adéquation des services fournis par les mécanismes de chiffrement usuels avec la transmission de données multimédia comme une vidéo ou une bande sonore. Une image dont la taille a doublé ou la luminance a varié n'est-elle pas identique à l'original ? Pourtant, une fonction de hachage ne reverra pas la même valeur pour les deux fichiers. Les techniques de tatouage se servent des spécificités des données multimédia pour d'une part offrir des services de sécurité similaires à ceux déjà connus, mais surtout ajouter des briques élémentaires à l'édifice [III, IV] :

- insertion d'empreinte numérique pour identifier les sources de copies illégales ;
- protection contre la copie ;
- surveillance d'une diffusion large à des fins de facturation ;
- intégrité au sens de données multimédia ;
- authentification des données ;
- étiquetage visant à informer l'utilisateur final.

[V] donne le cahier des charges d'un protocole qui mettrait en œuvre le tatouage couplé au cryptage et répondrait à la problématique « Alice vend à Bob – qui n'est pas digne de confiance – des données numériques qui ne peuvent être lues que sur un médium contrôlé » :

- le contenu numérique doit être crypté sur toutes les interfaces « ouvertes » ;
- le cryptage en tant que tel n'est pas suffisant ;
- chaque interface doit être munie d'un mécanisme de génération de clefs de session ;
- le contenu du médium de stockage doit être protégé contre la copie de bit à bit, par exemple *via* une clef de médium unique ;
- le cryptage de bout en bout est inadapté bien que préférable d'un point de vue de sécurité, on s'oriente plus aujourd'hui vers un cryptage de lien en lien ;
- le signal analogique ne doit pas pouvoir revenir dans le monde « sécurisé », **c'est ici que le tatouage devient nécessaire.**

[VI] donne sous la forme d'un théorème une condition fondamentale à l'utilisation du tatouage : le processus de vérification d'un tatouage doit prendre en compte l'image originale (sous une forme ou sous un autre), sinon le tatouage est inversible.

4.2 Copie conditionnelle

Le tatouage peut ajouter un service de copie conditionnelle si :

- le matériel de copie est conforme ;
- les données circulent en clair dans le monde numérique non sécurisé (une simple en-tête de fichier fournirait le même service dans le cas contraire).

5 Protocoles à base de tatouage

On trouve dans littérature [VI, VII, VIII] des mécanismes mettant en œuvre des techniques de tatouage, nous les résumerons dans cette partie.

5.1 Protection d'une image par une signature externe

[VII] propose un protocole de protection d'une image par une signature externe représenté fig. 5.2.1.

Le processus de dépôt ou de protection d'une image se fait en trois grandes étapes :

- Acquisition et insertion d'un tatouage fragile (temporaire) au niveau de l'appareil numérique ;
- Connexion et identification auprès d'une entité d'authentification ;
- Vérification et protection de l'image déposée.

Ce protocole assez simple assure donc l'authentification de l'image : l'identité de son propriétaire est inscrite dans l'image elle-même.

Les deux protocoles suivants sont plus spécifiquement orientés vers la protection des droits du client, ce qui nous le verrons protège encore d'avantage le propriétaire. On peut imaginer plusieurs cas qui rendent cette protection du client nécessaire pour **les deux parties** :

- Le propriétaire O vend par erreur la même version de l'image tatouée V_w aux clients B et C . Supposons que B donne illégalement V_w à une autre personne D . Du fait que B et C ont la même version, qui est responsable de la fuite ? Même si O donne à B et C des images tatouées différentes, B pourrait prétendre que O a fait une erreur et a donné à C la même version. B pourrait aussi prétendre que O a donné la même V_w à D . Comment C et/ou O peut prouver que B est responsable de la transaction illégale ?
- Le propriétaire O mandate A pour distribuer l'image V pour son compte. Légalement, A devrait donner aux différents clients B_1, B_2, \dots, B_{99} et C des images tatouées différentes $V_{w_{b_1}}, V_{w_{b_2}}, \dots, V_{w_{b_{99}}}$ et V_{w_c} , et reverser à O les royalties correspondantes. Cependant, A pourrait vendre illégalement la même version tatouée en utilisant le tatouage de C W_c pour C et pour B_1, B_2, \dots, B_{99} et alors prétendre qu'il n'a vendu qu'une copie à C . C serait alors accusé d'avoir revendu illégalement 99 copies parce qu'elles portent son tatouage.

Même si C achète de O les droits de V et obtient ainsi V_{w_c} , il ne peut pas s'en servir publiquement et le mettre par exemple sur une page web (au risque de se faire accuser de revente illicite). Existe-t-il une solution pour que C puisse utiliser V_{w_c} publiquement et simultanément interdire d'autres utilisations qui elles seraient illégales ?

5.2 Intervention du TTP en ligne

[VI] propose deux protocoles : l'un avec une intervention directe du TTP¹, l'autre où il n'intervient que pour la gestion des clefs et en cas de conflit.

La fig. 5.2.2 montre un des rôles que peut avoir le TTP. Le tab. 5.2.1 recense les symboles utilisés.

¹ *Trusted Third Part*, tiers de confiance

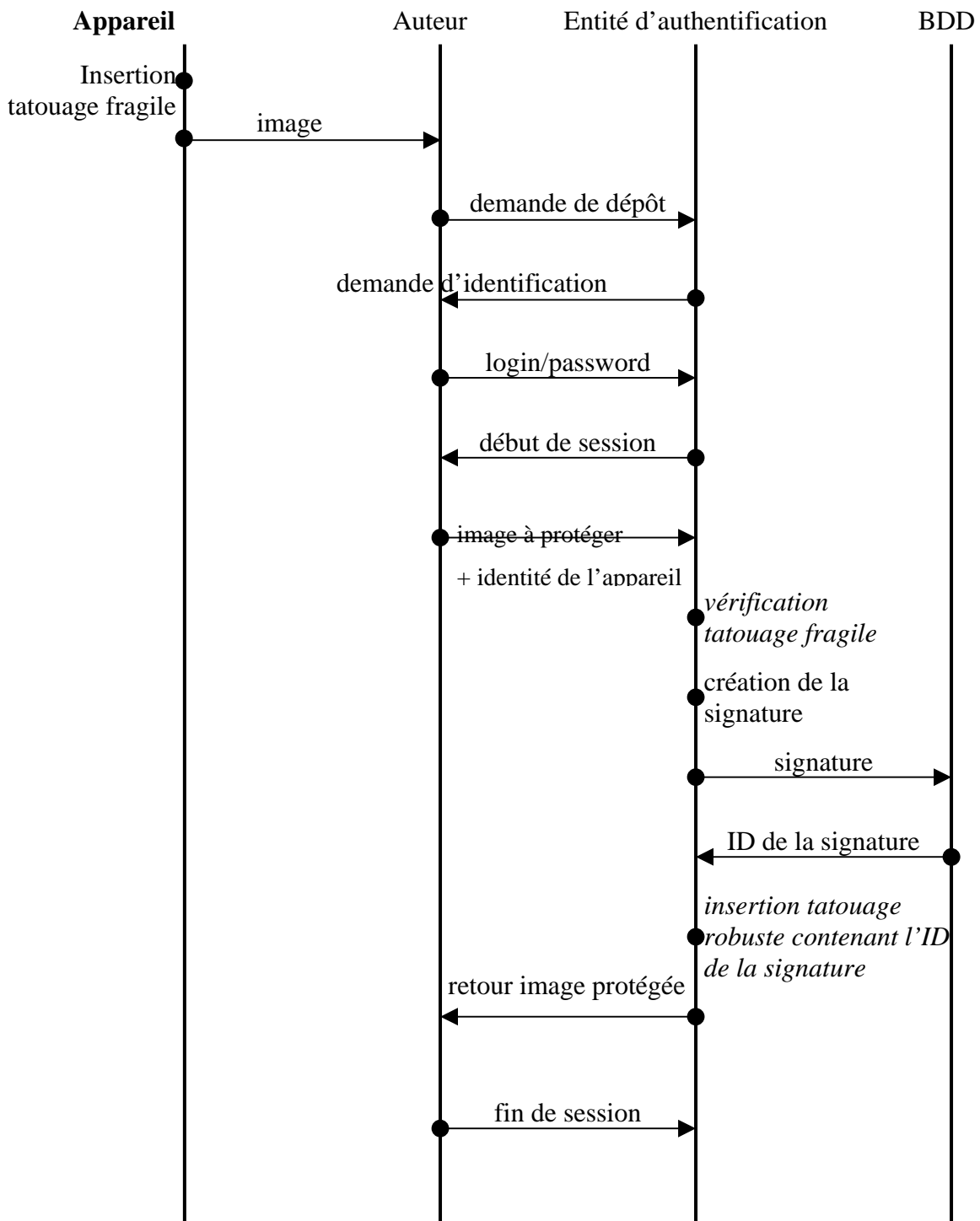


fig. 5.2.1 – Protocole de protection d'une image par une signature externe

TTP ou S	Tiers de confiance
O, A	Propriétaires de la donnée multimédia
K_a	Clef secrète connue uniquement par A, d'inverse K_a^{-1}
KP_a	Clef asymétrique publique, d'inverse KS_a , clef secrète
K_{sa}	Secret partagé entre Set A
$\{M\}_K$	Message M crypté par K
$V \otimes K \rightarrow V_w$	Donnée originale V tatouée par W pour former l'image tatouée V_w

tab. 5.2.1 – Table des symboles

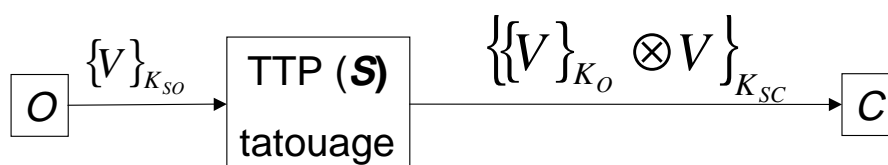


fig. 5.2.2 – Protocole à base de TTP

Les protocoles de tatouage à base de TTP utilisent un tiers de confiance entre le propriétaire O et le client C . O et C ne communiquent pas directement. Si O veut envoyer des données V_w tatouées à C , O envoie l'original V au tiers de confiance. le TTP crée alors V_w pour C en utilisant un algorithme de tatouage non inversible, s'assure que le tatouage utilisé est unique pour C , et envoie V_w à C . L'envoi d'une image tatouée de O vers C se fait alors comme suit :

- O envoie au TTP S : $\{V\}_{K_{SO}}$;
- S calcule : $V_w = \{V\}_{K_O} \otimes V$;
- S envoie à C : $\{V_w\}_{K_{SC}}$

Les avantages de ce protocole de tatouage sont :

- Le TTP peut s'assurer que chaque copie qui est distribuée possède un tatouage unique en son sein. Pour vérifier si un client est dans son droit ou non, il cherche dans ses enregistrements la présence du nom de l'utilisateur (le nom est authentifié par un autre canal) ;
- Il assure l'utilisation de schémas de tatouage standardisés.

Les défauts de ce protocole sont :

- la centralisation autour du TTP, qui devient un goulet d'étranglement ;

- le TTP a tous les originaux, s’il est compromis les droits des clients ne sont plus assurés et les originaux sont exposés ;
- la taille de la base de donnée du TTP sera rapidement trop grande.

5.3 Communication directe entre les deux entités

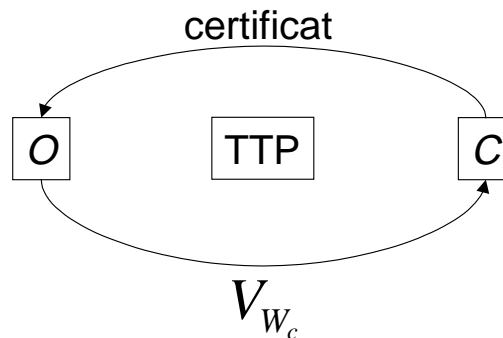


fig. 5.3.1 – Communication directe

Le TTP n’est présent que dans un but de vérification, comme représenté fig. 5.3.1. Pour assurer l’unicité de l’image tatouée V_w , l’utilisateur final C doit fournir au propriétaire A

(ou son représentant) la preuve de sa propre identité. Le problème est qu’alors A peut voir l’information fournie par C et donc A peut modifier, détruire ou remplacer l’information d’identité de C . Cet aspect est résolu comme suit :

- C calcule : $C_1 = \{C_0\}_{K_c}$
- C envoie à A : $C_1 = \{\{C_1\}_{K_{S_c}}\}_{K_{P_a}}$
- A calcule : $V_w = \{C_1\}_{K_a} \otimes V$
- A envoie à C : $\{\{V_w\}_{C_1}\}_{K_{P_a}}, \{K_a, V, C_1\}_{K_{S_a}}\}_{K_{P_c}}$

Reprenons avec un peu plus de détails cette procédure. On suppose que l’échange initial des clefs adéquates a déjà été réalisé.

i. Création du certificat

C utilise DES pour créer son propre certificat C_1 qui est une séquence de bits aléatoire. Sans connaître K_c , personne ne peut créer le même certificat C_1 , c’est DES qui nous le garantit.

La taille de C_0 devrait être la même que celle de l'image V . C_0 ne devrait pas être une séquence aléatoire, car sinon B pourrait choisir n'importe quel K_{b_0} , calculer $B_0 = \{C_1\}_{K_{b_0}^{-1}}$ et détruire l'unicité de C_1 (qui vérifierait alors $C_1 = \{B_0\}_{K_{b_0}}$). Le choix de C_0 pourrait être par exemple les n premiers chiffres de π .

K_c est conservée soigneusement et ne doit être produite au TTP qu'en cas de vérification.

ii. Envoi du certificat

De multiples algorithmes asymétriques permettent de mener à bien et de manière sécurisée cette transaction (cf. tab. 3.1.1).

iii. Tatouage

A utilise C_1 comme base de tatouage pour C et crée V_w . K_a est connue de A seul et va être envoyé au TTP uniquement en cas de vérification.

iv. Envoi de l'image tatouée

A envoie $\left\{ \left\{ V_w \right\}_{C_1} \right\}_{K_{P_a}}, \left\{ K_a, V, C_1 \right\}_{K_{P_c}} \right\}$ à C . En cryptant V_w par C_1 ,

- A atteste que c'est bien lui qui a créé le tatouage ;
- C atteste qu'il connaît C_1 et peut ainsi obtenir V_w .

Le premier point est nécessaire car sinon, en étape 2, A pourrait utiliser ζ'_1 que C ne connaît pas et ne pourrait ainsi pas prouver ses droits sur l'image.

Le second point est nécessaire car sinon, C pourrait nier avoir envoyé C_1 à A . La seconde partie du message est présente à des fins de vérification.

v. Vérification

Deux points peuvent être vérifiés :

- C doit vérifier que V_w a bien été créé par le bon protocole, pour vérifier qu'il dispose bien des droits sur la donnée ;
- le propriétaire des droits sur la donnée doit pouvoir être désigné sans ambiguïté.

Le TTP entre alors en jeu. C envoie au TTP $\left\{ \left\{ \left\{ V_W \right\}_{C_1} \right\}_{K_{S_a}}, \left\{ K_a, V, C_1 \right\}_{K_{S_a}}, K_c, C_0 \right\}_{K_{S_c}}$.

On note que C_0 est connu de A et C .

- pour vérifier que V_W a été créé avec le bon protocole, le TTP vérifie la formule de l'étape 3
- pour vérifier que C est détenteur des droits sur V_W , le TTP vérifie que
 - C_1 est la graine convenue entre A et C ;
 - $C_1 = \{C_0\}_{K_c}$. Clairement, si B détient V_W pour une raison quelconque mais ne connaît pas K_c , il lui est impossible de générer une clef K_b telle que $C_1 = \{C_0\}_{K_b}$, ce qui établit l'unicité des droits de C .

6 Architecture CPSA

CPSA est une architecture visant à assurer la protection du contenu, en utilisant la cryptographie et le tatouage [VIII]. Elle a été conçue par la 4C-entity¹ en collaboration avec le SDMI² et le CPTWG³ pour rester cohérent avec ces approches et les englober.

CPSA donne en 11 axiomes un cadre de développement de matériels normalisés :

- le propriétaire du contenu fixe son CMI⁴ ;
- l'intégrité du CMI doit être assurée ;
- le tatouage peut être utilisé, il contiendra alors le CMI ;
- le contenu des média préenregistrés doit être crypté ;
- les copies autorisées doivent être cryptées ;
- si le support n'est pas crypté et en cas de présence d'un tatouage, la restitution doit être interdite ;
- les transmissions doivent être cryptées et se faire en fonction des CMI entre tous les modules CPSA, à l'exception des DVD audio ;
- un contenu non crypté doit subir une vérification de son CMI tatoué avant d'être crypté, le CMI numérique ainsi défini doit être une copie du CMI tatoué ;
- le CCI⁵ (qu'il soit numérique ou tatoué) doit être examiné avant la copie ;
- le CCI (tatoué ou numérique) doit être mis à jour avant la copie ;
- le CCI (tatoué ou numérique) ne doit pas être mis à jour pour la création d'une image temporaire et localisée.

En particulier, CPSA reconnaît plusieurs technologies comme étant compatibles avec ses spécifications (liste non exhaustive) :

- CPRM⁶ pour la protection des média enregistrables comme les DVD, la mémoire flash ou les disques dur ;
- CPPM⁷ pour la protection du contenu audio d'un DVD préenregistré ;
- CSS⁸ pour la protection du contenu vidéo d'un DVD préenregistré ;
- DTCP¹ pour la protection du contenu lors des transmissions numériques ;

¹ Intel, IBM, Matsushita et Toshiba

² *Secure Digital Music Initiative*, <http://www.sdmi.org>

³ *Content Protection Technology Working Group*, groupe de travail sur les technologies de protection du contenu

⁴ *Content Management Information*, informations de gestion du contenu

⁵ *Copy Control Information*, informations de contrôle de copie

⁶ *Content Protection for Recordable Media*, protection du contenu des supports enregistrables

⁷ *Content Protection for Pre-recorded Media*, protection du contenu des supports préenregistrés

⁸ *Content Scrambling System*, système de chiffrement du contenu

- HDCP² pour la protection du contenu se déplaçant à travers des interfaces à haut débit vers un affichage numérique ;
- l'accès conditionnel pour la distribution protégée du contenu par câble ou satellite ;
- le tatouage 4C du contenu audio [IX] ;
- un tatouage vidéo qui doit encore être choisi par le DVD CCA.

6.1 SDMI

SDMI [X]³ propose un modèle de référence, dans lequel les modules sous licence (LCM⁴) interfacent les applications qui délivrent la musique avec les périphériques portables (PD⁵) chargés de restituer le son à destination du client. SDMI reprend l'ensemble des notions définies par CPSA et n'utilise aucune notion différente de celles manipulées par HDCP ou de CPRM. L'apport de SDMI sera significatif lorsqu'une technologie efficace de tatouage du son sera normalisée et incluse dans l'architecture, ce qui n'est pas encore le cas.

6.2 CPRM

CPRM [XI] fait partie intégrante de CPSA et définit

- un ensemble de fonctions cryptographiques (cryptage, hachage, modes de chaînage, génération de nombres aléatoires) ;
- la gestion des clefs de cryptographie.

Chaque périphérique reçoit à sa fabrication un ensemble de n clefs qui doivent absolument rester secrètes. Elles servent à fabriquer la clef de médium K_m unique à partir du bloc de clef de média MKB. L'écriture et la lecture d'un DVD sont schématisés fig. 6.2.1 et fig. 6.2.2.

L'objet du bloc de clef de média MKB est d'assurer le renouvellement du système. Si un ensemble de clefs de périphérique est compromis, le 4C peut mettre en circulation un MKB mis à jour qui, combiné avec les clefs de périphérique compromis, forme une clef K_m différente (en fait 0000000000000016) : c'est ainsi qu'est assurée la révocation des clefs de périphériques. Les extensions aux blocs de clef de média, quant à elles, permettent de restreindre la validité des MKB : elles en modifient progressivement le comportement jusqu'à les rendre inutilisables par les clefs de périphériques valides (et non valides).

¹ *Digital Transmission Content Protection*, protection du contenu lors de la transmission numérique

² *High-bandwidth Digital Content Protection*, protection du contenu lors de la transmission numérique à haut débit

³ *Secure Digital Music Initiative*

⁴ *Licensed Compliant Module*, modules sous licences SDMI

⁵ *Portable Device*, périphériques portables

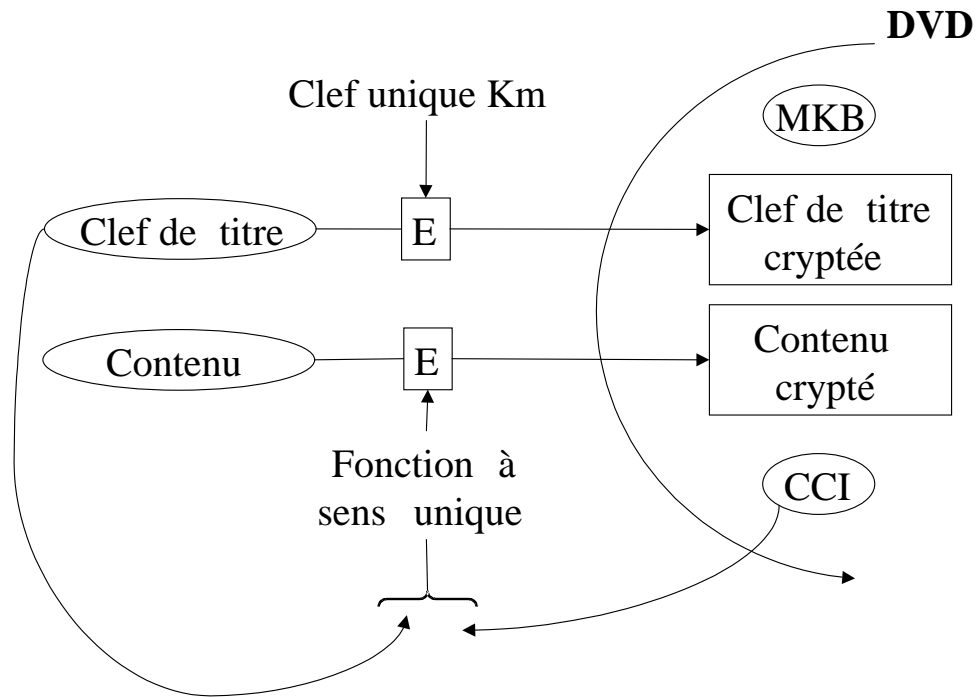


fig. 6.2.1 – Content Protection for Recordable Media : écriture sur le DVD

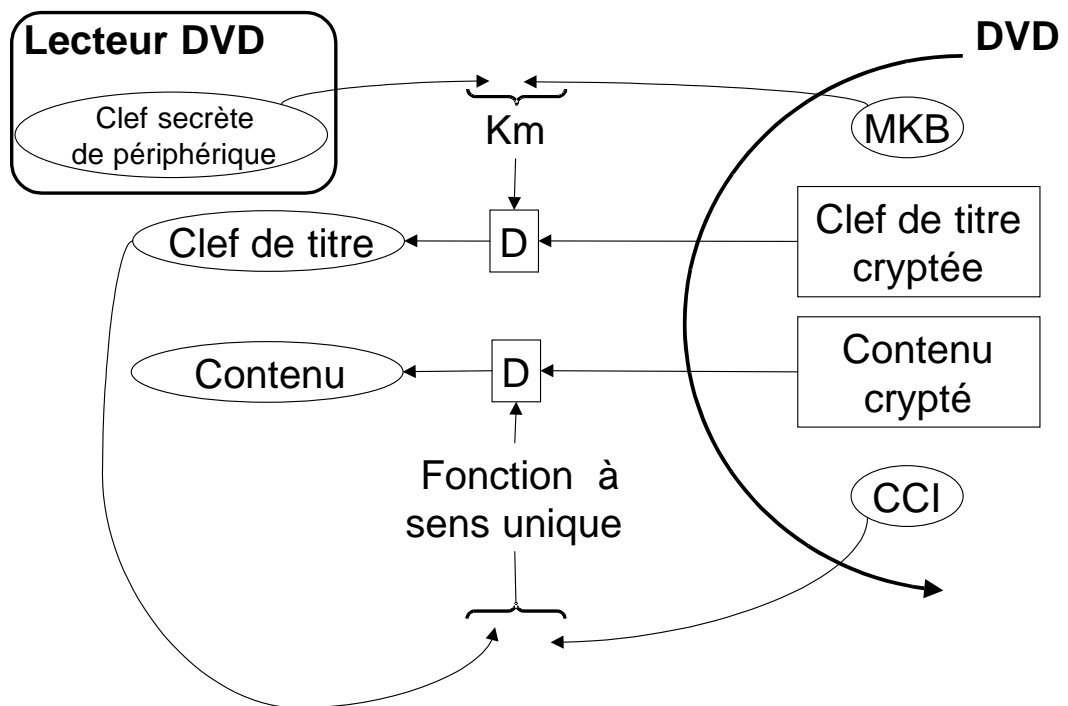


fig. 6.2.2 – Content Protection for Recordable Media : lecture du DVD

6.3 CSS

Cette norme de protection a été analysée dans le détail [XII] et plusieurs failles de taille exploitées. La première grande faille est qu'elle est mise en œuvre de manière logicielle, typiquement dans les pilotes des lecteurs de DVD. Le chemin qui mène alors les données chiffrées jusqu'au haut de la couche application est long, et aucun système d'exploitation ne peut interdire à un utilisateur de dévier le flux non chiffré et en faire ce que bon lui semble.

Frank STEVENSON a quant à lui montré que la facette cryptographique de CSS est largement défailante :

« My main interest in this is purely cryptographical, so I have little or no knowledge of the problems associated with CSS. What I have done is devise an attack that will recover a CSS key with a complexity of 2^{16} and as little as 6 known output bytes. This should reduce the key recovery time from about 17 hours to a fraction of a second. The CSS algorithm is fatally flawed. A divide and conquer attack is possible by guessing the 16 unknown bits of LFSR1. LFSR1 is then clocked 4 times, and the known keystream bytes are then used to reconstruct the state of LFSR2. The whole cipher is then clocked another 2-6 times to validate the key. If the key is correct LFSR2 is clocked backward 4 times to retrieve the initial state. The fine details can be found in the source code below. »

Des programmes d'une dizaine de lignes écrits en langage C ou en Perl circulent sur l'Internet et montrent les faiblesses de ce schéma de cryptage.

6.4 DTCP

DTCP [XIII, XIV] définit un protocole cryptographique pour protéger le contenu audio et vidéo à la traversée de mécanismes comme les bus série à haute performance (IEEE 1394-1995). Seuls les contenus fournis à un périphérique source *via* un autre système de protection de copie approuvé – par le 4C – comme le CSS (cf. Paragraphe 6.3) seront protégés par ce système.

DTCP agit à quatre niveaux :

- Information de contrôle de copie (CCI) ;
- Authentification de périphérique et échange de clefs (AKE¹) ;
- Cryptage du contenu ;
- Mise à jour du système.

¹ *Authentication and Key Exchange*, authentification et échange de clefs

i. Authentification

DTCP prévoit deux types d'authentification initiée par le récepteur, tous deux conduisent à l'établissement d'une clef d'authentification qui sera utilisée par la suite.

– *L'identification complète*

Elle se fait en utilisant la signature numérique (signature et vérification) et l'algorithme d'échange de clefs de type Diffie-Hellman. Cette identification doit être utilisée pour le contenu de type « ne jamais copier », et tient compte des listes de révocation de certificat.

– *L'identification restreinte*

Elle est utilisée pour les périphériques à puissance de calcul limitée, essentiellement à base de secrets partagés (réponse à un challenge aléatoire en le modifiant *via* le secret partagé puis hachage). Elle permet la protection de copies de type « une seule génération » et « plus de copie ».

ii. Gestion du canal de contenu

– *Clef d'échange*

Après l'authentification (complète ou restreinte), la source établit la clef d'échange K_x de manière aléatoire. La clef de contenu K_c est calculée à partir de

- K_x , la clef d'échange ;
- N_c , nombre aléatoire généré par le périphérique source et envoyé en clair à tous les périphériques concernés dans des paquets asynchrones ;
- C_a , C_b ou C_c des constantes liées au mode EMI indiqué dans l'en-tête du paquet.

La clef K_x est ensuite cryptée en utilisant K_{auth} mise au point en phase d'authentification, pour donner K_{SX} suivant des spécifications protégées par une licence. La source envoie K_{SX} au périphérique intéressé qui le décrypte *via* K'_{auth} (toujours mis au point en phase d'authentification). Finalement, la source met à jour le SRM¹ si nécessaire.

– *Clef de contenu*

La première clef de contenu est calculée à partir de la clef d'échange et d'une graine, un nombre aléatoire généré par la source et envoyé en clair à la destination. Cette clef de contenu change toutes les de 30 à 120 secondes. La nouvelle clef est calculée avec la même méthode que la première clef mais en prenant comme graine $N_c + 1 \text{ mod } 2^{64}$, la source est prévenue du changement de graine par un bit de parité.

¹ System Renewability Message, message de mise à jour du système

iii. Mise à jour du système

Les périphériques permettant une authentification complète (et eux seuls) peuvent recevoir et interpréter des messages de mise à jour (SRM) créés par le DTLA¹. Un SRM est composé entre autres d'un numéro de version strictement croissant (connu comme X_{SRM} pendant l'authentification totale), la liste de révocation de certificats (CRL) et la signature numérique de 320 bits de ce contenu.'

Le passage à l'échelle est assuré par un rajout successif de révocations de génération ultérieure à la fin et de plus basse priorité. Les SRM peuvent être mis à jour par

- d autres périphériques normalisés
- le contenu d'un médium préenregistré
- les flux de contenu de périphériques qui peuvent communiquer de manière externe (*via* l'Internet, la ligne téléphonique, les réseaux câblés, les satellites de broadcast etc.).

6.5 HDCP

HDCP serait un maillon final dans une solution de cryptage de bout en bout. Les solutions de cryptage de DVD actuels tels que CSS (cf. paragraphe 6.3) sont très vulnérables à une interception du flux décrypté à haut niveau. HDCP permet de repousser le décryptage aux périphériques d'affichage nouvelle génération connectés *via* une DVI² [XV]. Les éléments clef du système HDCP sont

- l'authentification pour vérifier qu'un périphérique d'affichage est habilité à recevoir le contenu protégé ;
- le cryptage de la vidéo transmise pour éviter l'écoute clandestine du contenu protégé et
- la possibilité de mise à jour pour révoquer les périphériques compromis.

Le chiffrement de HDCP se fait par bloc pendant la phase d'authentification, puis par flux pendant le cryptage et décryptage (ou exclusif bit à bit avec un flux fournit par HDCP).

7 Approches ouvertes

7.1 OPIMA

OPIMA³ [XVI] a pour but de créer un marché ouvert de livraison de contenu et assurer ainsi l'interopérabilité des systèmes de protection avec les terminaux multimédia.

¹ Digital Transmission Licensing Administrator, administrateur des licences de transmission numérique

² Digital Visual Interface, Interface Visuelle Numérique

³ Open Platform Initiative for Multimedia Access, initiative de plate-forme ouverte pour l'accès multimédia

OPIMA opère *via* une machine virtuelle, OVM¹, qui garanti la sécurité des ajouts (plug-ins) représentant un système de gestion et de protection de la propriété intellectuelle spécifique (IPMP²), et qui typiquement serait propriétaire. Le rôle d'OVM est essentiellement d'implémenter deux interfaces de programmation d'applications (API),

de service à une application (par exemple, un logiciel de rendu vidéo demande un accès à un contenu identifié par une URL) ;

de service à un module IPMP (on trouve ainsi comme méthode *extractWatermark* [XVII]). SSL a été choisi comme plug-in de téléchargement, l'authentification se fait entre le serveur de contenu et le tiers OPIMA.

7.2 IETF

Le site de l'IETF³ et en particulier l'éditeur de RFC⁴ [XVIII] ne font état d'aucun travail avancé en matière de protection du contenu : une recherche du type `(c|C)ontent.*(p|P)rotection` sur l'ensemble des RFC ne renvoie que ceux sur S/MIME et AKE¹.

7.3 3GPP

Une recherche sur `content AND protection` sur le site du 3GPP⁵ [XIX] fournit 121 réponses, qu'on peut résumer à cette phrase trouvée dans l'une d'entre elles :

The protection of copyrighted content (pictures, sound, video, trademark symbols, etc.) should be seen as an issue beyond the scope of SMS/EMS. There are currently several different organizations defining copyright protection mechanisms.

Les documents présents sur le site ne font en effet allusion qu'à la protection du copyright des SMS *via* des consignes binaires en en-tête de message. Le 3GPP semble donc considérer pour l'instant les architectures de protection du contenu comme étant hors de son champ de visibilité.

7.4 MPEG

i. MPEG-4

MPEG⁶[XX] a défini IPMP¹ pour donner une utilisation pratique du standard MPEG-4. Deux extensions de MPEG-4 vont dans ce sens :

¹ *OPIMA Virtual Machine*, machine virtuelle OPIMA

² *Intellectual Property Management and Protection*, protection et gestion de la propriété intellectuelle

³ *Internet Engineering Task Force*

⁴ *Request For Comments*, documents définissant les standards Internet

⁵ *Third Generation Partnership Project*, projet de partenariat pour les mobiles de troisième génération

⁶ *Moving Picture Experts Group*, Groupe d'experts en images animées

- IPMP-Ds² : une partie des descripteurs d'objet de MPEG-4 qui décrit la manière d'accéder et de décoder un objet. Une autorité d'enregistrement indépendante est utilisée pour identifier un système IPMP particulier et éviter les collisions ;
- IPMP-ES³ : les objets MPEG sont représentés par des flux élémentaires qui peuvent se référencer l'un l'autre. Certains peuvent être utilisés pour transporter des données IPMP. Ni leur syntaxe ni leur sémantique ne sont précisés dans le standard.

MPEG-4 traite ainsi le problème des droits de propriétés intellectuelles par insertions dans les objets d'un code d'identification IPI⁴ donnant des informations sur le contenu, le type du contenu et les droits attenants à l'objet en question. Les données contenues dans l'IPI et associées à chaque objet peuvent différer même pour des objets appartenant à une même image (par ex: droits libres sur le fond, mais restreint sur le personnage). L'insertion de l'IPI au moment du codage implique également l'insertion des mécanismes de protection équivalent aux droits sur l'image (protection contre les copies, facturation,...).

ii. MPEG-7

MPEG-7 définit les éléments suivants: les descripteurs (Ds), les organisations de descripteurs (DSs), un langage de définition de description (DDL – fondé sur le XML) et une couche système.

MPEG-7 devrait fournir un mécanisme pour désigner les droits de contenu (propriétaire des droits, informations contractuelles).

iii. MPEG-21

Un niveau d'abstraction supplémentaire est atteint dans MPEG-21, où l'interaction entre deux utilisateurs est modélisée sur la base d'événements [XXI] comme représenté fig. 7.4.1. Ni l'architecture dans laquelle s'inscrivent ces événements ni les spécifications des événements eux-mêmes ne sont encore définis.

¹ *Intellectual Property Management and Protection*, Protection et gestion de la propriété intellectuelle

² *IPMP-Descriptors*, Descripteurs IPMP

³ *IPMP-Elementary Streams*, Flux élémentaires IPMP

⁴ *Intellectual Property Identification*, Identification de la propriété intellectuelle

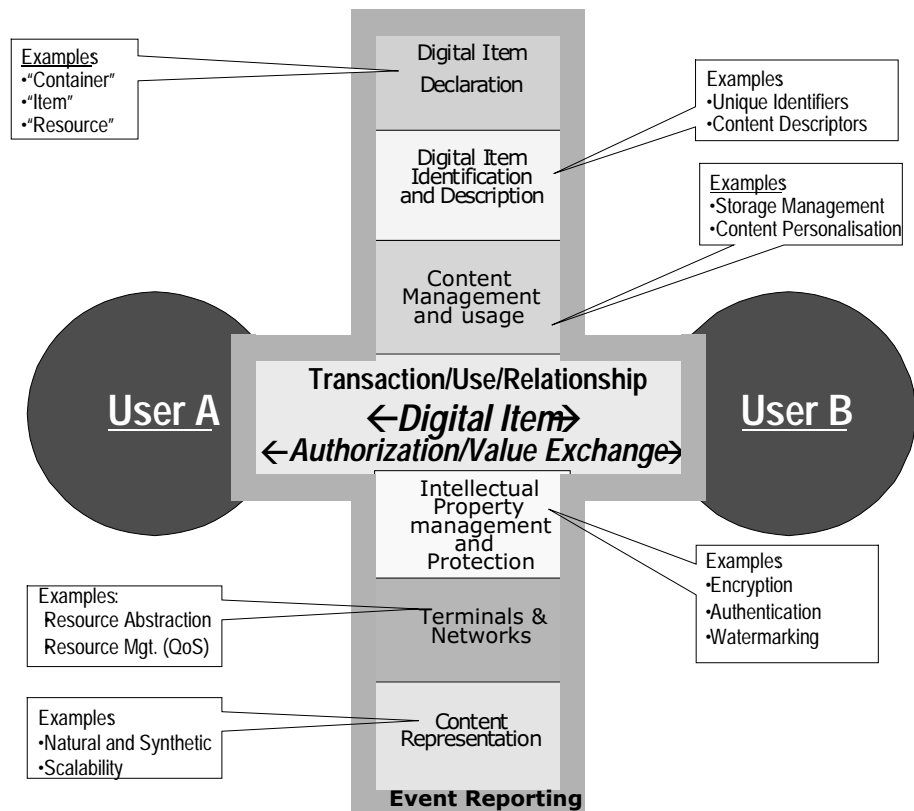


fig. 7.4.1 – Gestion des événements dans MPEG-21

MPEG prévoit donc dans ses normes le contenant multimédia de toutes les informations propres à assurer la protection du contenu, sans préciser le contenu à y apporter ni l'architecture globale mettant en œuvre ces différentes briques.

8 Mobile Music Forum

La fig. 7.4.1 donne le processus de diffusion de contenu tel que présenté par le Mobile Music Forum. La carte SIM est considérée comme un environnement d'exécution sur à la différence du mobile. À cause de sa puissance de calcul limitée, son action se limite au décryptage après vérification des droits d'accès de la clef de chiffrement des données multimédia. Cette approche a l'avantage de tenir compte de la mobilité de l'utilisateur et de l'infrastructure existante dans le monde GSM.

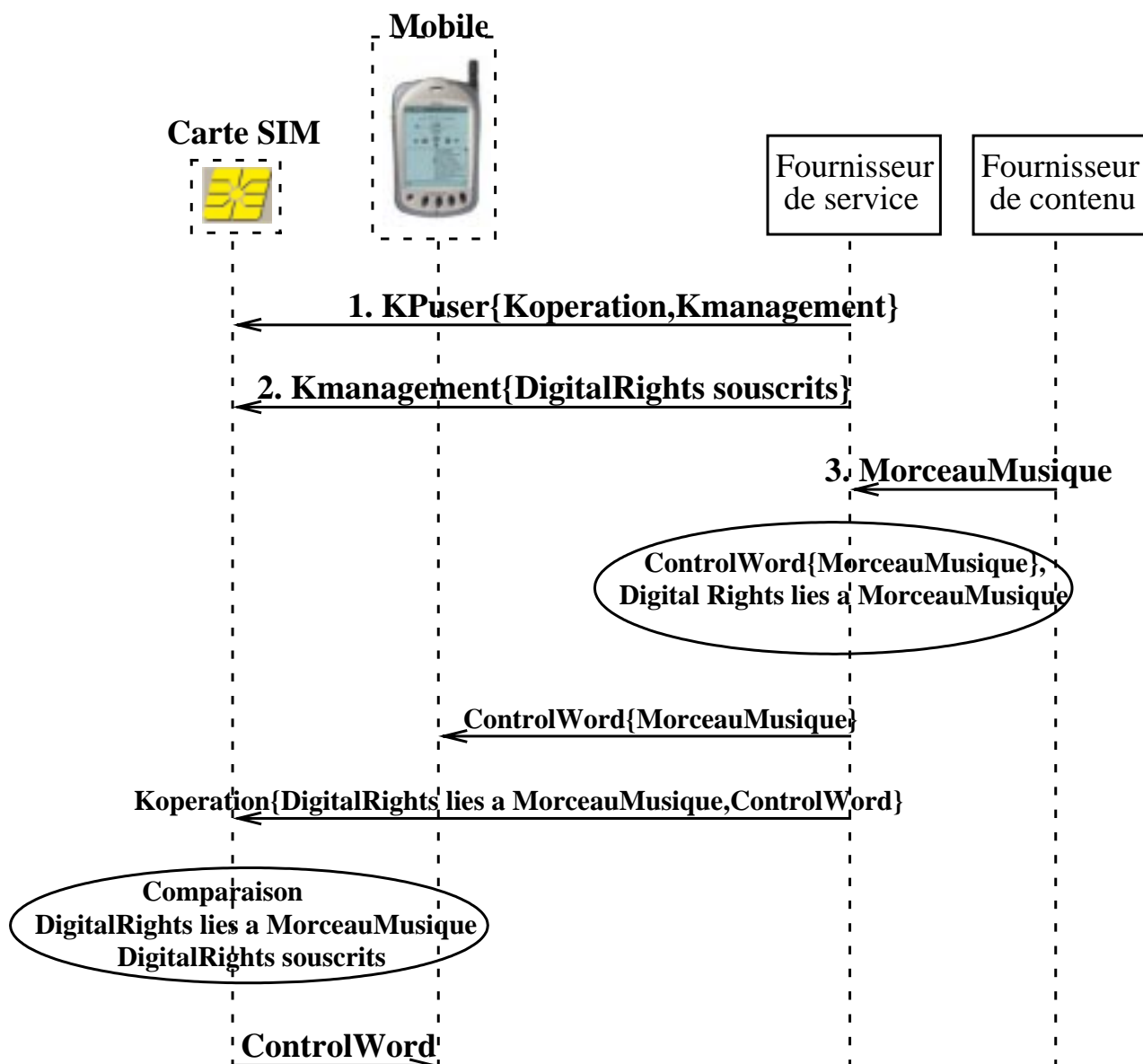


fig. 7.4.1 – Processus de diffusion de contenu présenté par le Mobile Music Forum

La granularité de la définition de la propriété intellectuelle et des droits de copie est *a priori* très fine, tant que le morceau de musique reste dans le portable de l'utilisateur. La difficulté réside ensuite à interfacier cette architecture de protection du contenu avec d'autres

périphériques peut-être moins sous contrôle. Pour finir, ce modèle a le même défaut que CSS : les données sont transmises en clair au périphérique final, que le système ne contrôle pas. Il est alors aisé d'imaginer de reprogrammer de manière abusive le périphérique final, pour qu'il ne se contente plus de restituer le son, mais aussi le retransmette hors du monde sécurisé.

9 Synthèse

La grande variété d'architectures que l'on a pu rencontrer peut se synthétiser de manière quasiment caricaturale en deux approches.

9.1 Approches

Les différentes approches existantes répondent donc à des besoins très différents :

i. protection exclusive des droits du propriétaire

Cette approche a été adoptée par les 4C avec CPSA ainsi que par le Mobile Music Forum, dans le cadre d'un modèle de cryptage de lien par lien (décryptage puis cryptage à chaque interface), qui vise à ne transmettre que des données cryptées depuis la source jusqu'à l'ultime interface contrôlée, l'écran numérique dans le cas des données visuelles. Ses différentes composantes sont suffisamment découplées pour pouvoir être adoptées individuellement.

ii. protection des droits du client

Cette approche introduit nécessairement la notion de tiers de confiance (ce qui n'était pas le cas auparavant), essentiellement avec l'espoir d'une répartition plus équitable des responsabilités, ce modèle serait plus apte à être reconnue par une autorité légale.

9.2 Comparaison des deux approches

Des critiques virulentes ont accueilli l'annonce de la demande d'introduction de CPRM dans la norme ATA¹. Il est en effet très séduisant de considérer CPRM comme une tentative des éditeurs de contenu de se protéger, à leur seul bénéfice et au détriment des utilisateurs. On peut ainsi s'inquiéter de la complexification qu'engendrerait un cryptage systématique des disques durs : les sauvegardes de ce disque seraient-elles aussi cryptées avec la clef du disque original, comment accéder aux données en cas de défaillance physique du matériel ? On ne peut de plus que difficilement imaginer l'application restreinte de CPRM à certaines données sur le disque dur, le cryptage se faisant au niveau physique il ne pourrait se limiter qu'à des partitions de taille fixe, et pas à des fichiers particuliers.

Une autre critique vise la modularité de CPRM elle-même : si une seule composante de CPSA n'est pas déployée (la télévision haute définition acceptant la norme HDCP par

¹ AT Attachment, l'une des interfaces de périphérique les plus répandues

exemple), le reste est caduc car le décryptage des données et l'envoi en clair vers un périphérique non contrôlé seront alors inévitables et feront s'écrouler la chaîne de confiance. Le cryptage doit donc se faire jusqu'au bout, toute demi-mesure paraît bancale. Le tatouage peut venir au secours d'un déploiement partiel car il permet aux CMI de résister à un passage éventuel par le monde sécurisé, mais il ne peut être contraignant que dans le monde CPSA.

Vient alors la question du respect de la *privacy* des utilisateurs, qui se voient d'une part dépossédés du contenu des données qu'ils manipulent et d'autre part identifiés par des clefs de périphérique à la merci des autorités de certification et de révocation.

On peut ensuite s'interroger sur la gestion de la cohérence des SRM : les spécifications actuelles prévoient que chaque périphérique présente sa liste de révocation et que la plus récente est choisie. Ceci suppose une gestion très centralisée de la génération des SRM pour assurer la cohérence de la base de données. La taille sans cesse croissante de la liste des SRM dans chaque périphérique est aussi un motif d'inquiétude.

L'approche CPSA est initiée par des acteurs suffisamment majeurs du secteur de la distribution et du matériel pour que l'on souhaite rester compatible avec elle. Il est même envisageable, et nous le verrons dans le chapitre suivant, d'apporter des améliorations substantielles à CPSA tant sur la gestion des SRM que dans la gestion des clefs en général et ainsi disposer d'une architecture fonctionnelle en tant que telle, tout en restant compatible avec ce standard qui semble émerger.

Le Mobile Music Forum apporte la gestion de la mobilité aux notions déjà présentes dans CPSA *via* le découplage de la gestion cryptographique des clefs et droits d'accès de celle du décryptage brutal des données.

Enfin, pour certains services de distribution dans lesquels l'acheteur est personnalisé, l'introduction d'un tiers de confiance est nécessaire. La notion de propriété est alors à granularité plus fine, on disposerait d'un droit de copie et de diffusion modulable et des moyens de faire reconnaître ce droit. La grande difficulté de cette approche réside dans le passage à l'échelle tant du nombre d'utilisateurs que du volume des données certifiées.

Chapitre III - Proposition

1 Contexte

L'État de l'art a résumé les propositions du *Mobile Music Forum* concernant la gestion des secrets et des droits de copie dans la diffusion d'un morceau de musique. L'approche proposée a pour le moins le défaut de mal passer à l'échelle, dans le sens où elle suppose le cryptage par une clef différente pour chaque destinataire du contenu.

Nous nous plaçons dans le cas de la diffusion de documents changeant régulièrement et de mise à jour aléatoire. [XXII] a prouvé que dans ce cas, le multicast était plus adapté qu'un système de caches.

Nous tenterons de proposer ici une diffusion scalable des secrets dans le cadre d'une diffusion multicast d'un contenu multimédia. Nous introduirons un modèle original de gestion des clefs de contenu, propre à être mise en œuvre dans une architecture de distribution.

L'hypothèse de travail est le changement régulier de la clef de contenu, comme cela est le cas dans la majorité des protocoles de cryptage. La compromission de la clef de contenu a ainsi une portée limitée dans le temps et dans l'espace.

Vient alors le problème de la clef qui crypte cette clef de contenu. On peut considérer deux approches extrêmes :

- Tous les MS¹ décryptent la clef de contenu avec la même clef. Cette approche nécessite de changer toutes les clefs dans le cas d'une révocation (un MS quitte le groupe de réception des données multimédia). La difficulté est alors localisée au niveau de l'ensemble des MS qui sont affectés à chaque changement de groupe ;
- Le serveur de contenu crypte la clef de contenu avec une clef différente pour chaque MS. La difficulté est déplacée vers le serveur qui devrait gérer tout seul l'envoi de ces secrets.

Pour gérer le problème de base qu'est la révocation de la clef de contenu, on va tenter de lier la localisation du mobile (la cellule où il se trouve) avec la clef utilisée. Nous avons comme effet de bord le changement de la clef de contenu avec la cellule en cas de *Handover*².

Ce schéma nécessite deux primitives :

- abonnement d'un utilisateur à un groupe ;
- désabonnement de l'utilisateur du groupe.

¹ *Mobile Station*, station mobile

² *Transfert de compétence*

Ainsi, si l'utilisateur change de cellule, il se désabonne puis se réabonne dynamiquement.

2 Formalisation

Nous avons montré dans l'état de l'art que la distribution de contenu faisait nécessairement intervenir le fournisseur de contenu, le distributeur de contenu et l'utilisateur. Les relations entre ces trois entités se doivent d'être strictes pour assurer une distribution de contenu sécurisée :

- l'utilisateur final ne doit avoir accès qu'au contenu chiffré et à la clef de déchiffrement ;
- le distributeur de contenu peut (et doit) chiffrer, mais il ne doit pas garder le contrôle sur ses clefs de chiffrement, au moins parce qu'il peut exister *plusieurs* distributeurs ;
- le fournisseur de contenu doit garder le contrôle des clefs de chiffrement du contenu, mais sans posséder une infrastructure de distributeur de contenu.

Nous avons proposé une organisation de la gestion des clefs en trois plans :

- le plan de distribution des données où circulent
 - les données cryptées par une clef de contenu ;
 - la clef de cryptage cryptée par une autre clef ;
- le plan de gestion des clefs où l'on trouve les nœuds du réseau qui cryptent les données ;
- le plan de gestion de l'appartenance au groupe, où le fournisseur de contenu est souverain.

Une description plus précise de ces différents plans ainsi que de leurs relations est présente dans l'article inséré Chapitre IV -.

3 Cryptage multicast

3.1 Cryptage symétrique

Une première forme de cryptage multicast peut se faire de manière symétrique [XXIII] en utilisant une extension de l'algorithme de El Gamal qui est expliqué paragraphe 3.2 .

La fig. 3.1.1 montre le cheminement de la clef de contenu et ses cryptages successifs.

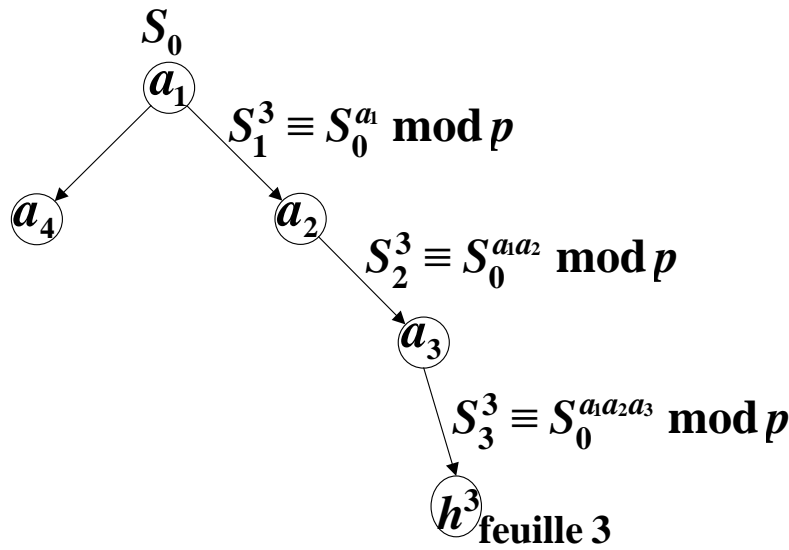


fig. 3.1.1 – Cryptage multicast à logarithme discret

i. Initialisation

La source choisit un grand nombre **premier** p , un générateur g du groupe cyclique Z_p^* et une valeur aléatoire $r \in Z_{p-1}^*$ (ainsi, les $g^r \bmod p$ sont générateurs de Z_p^* , ce qui assure la sécurité de l'ensemble des transformations à venir). Les nœuds et la racine reçoivent des valeurs $a_{i>0} \in Z_{p-1}^*$. La valeur initiale de la séquence est fixée à $S_0 = g^r \bmod p$.

On désigne par $\{S_{ik>0}\}$ les éléments de la séquence. La fonction inverse distribuée aux feuilles s'écrit $h_{(0,n_k)}^k(x) = x^{(a_{i_1} \cdot a_{i_2} \cdot a_{i_3} \cdots a_{i_{n_k}})^{-1}} \bmod p$.

La fonction f appliquée à chaque nœud s'écrit $f(x, a_{i_k}) = x^{a_{i_k}} \bmod p$.

ii. Distribution des clefs

Soit K la clef que la source veut distribuer aux feuilles de l'arbre multicast. La source envoie à chacun de ses fils $S_1 = (S_0)^{a_1} \bmod p$ et $T = K \oplus S_0$. Les nœuds intermédiaires appliquent $S_{i_{k-1}}$ à leur entrée et envoient S_{i_k} et T à leurs fils, sachant que $S_{i_k} \equiv f(S_{i_{k-1}}, a_{i_k}) \equiv (S_{i_{k-1}})^{a_{i_k}} \bmod p$.

iii. Clef suivante

La clef suivante \tilde{K} requiert le changement de $S_0 = g^r \bmod p$ en $\tilde{S}_0 = g^{\tilde{r}} \bmod p$ dans Z_p^* .

La symétrie de ce schéma vient de ce qu'il peut être parcouru dans les deux sens : un nœud peut diviser l'exposant par son coefficient pour faire remonter l'information car il sait avec quel modulo le faire comme expliqué Annexe A3.2 (diviser en arithmétique se fait en multipliant par l'inverse qui s'obtient par l'algorithme d'Euclide).

3.2 Cryptage asymétrique

i. Initialisation

La source choisit deux grands nombres p et q et forme le produit $n=pq$. Les nœuds et la racine reçoivent des valeurs $a_{i>0}$ vérifiant $a_i \wedge \varphi(n)=1$ de sorte que A , produit d'un ensemble non vides d'éléments a_i vérifie aussi $A \wedge \varphi(n)=1$. La fonction appliquée aux nœuds s'écrit $f(x, a) \equiv x^a \pmod n$.

La fonction inverse distribuée aux feuilles s'écrit $h_{(0, n_k)}^k(x) = x^{(a_{i_1} \cdot a_{i_2} \cdot a_{i_3} \cdots a_{i_{n_k}})^{-1}} \pmod p$ étant entendu que $(a_{i_1} \cdot a_{i_2} \cdot a_{i_3} \cdots a_{i_{n_k}})^{-1} (a_{i_1} \cdot a_{i_2} \cdot a_{i_3} \cdots a_{i_{n_k}}) \equiv 1 \pmod{\varphi(n)}$, calcul qui ne peut être effectué que par celui qui possède la décomposition $n=pq$.

ii. Distribution des clefs

Soit K la clef que la source veut distribuer aux feuilles de l'arbre multicast. La source envoie à chacun de ses fils $S_1 = K^{a_1} = S_0^{a_1} \pmod n$. Les nœuds intermédiaires appliquent S_{i-1} à leur entrée et envoient S_i à leurs fils, sachant que $S_i \equiv f(S_{i-1}, a_i) \equiv (S_{i-1})^{a_i} \pmod n$.

iii. Clef suivante

La clef suivante \tilde{K} requiert le changement de $S_0 = g^r \pmod p$ en $\tilde{S}_0 = g^{\tilde{r}} \pmod p$ dans Z_p^* .

La symétrie de ce schéma vient de ce qu'il peut être parcouru dans les deux sens : un nœud peut diviser l'exposant par son coefficient pour faire remonter l'information car il sait avec quel modulo le faire comme expliqué Annexe A3.2 (diviser en arithmétique se fait en multipliant par l'inverse qui s'obtient par l'algorithme d'Euclide).

iv. Choix de l'asymétrie

Comme nous l'avons expliqué au paragraphe 3.1 l'algorithme symétrique a pour fonctionnalité de ne pouvoir régénérer la nouvelle fonction h à partir de la fonction précédente à partir du paramètre de l'un des nœuds présents sur le trajet.

L'algorithme asymétrique est dans ce sens plus sûr, car seuls les détenteurs de l'indicatrice d'Euler sont à même d'effectuer ces calculs.

3.3 Application à un réseau GSM

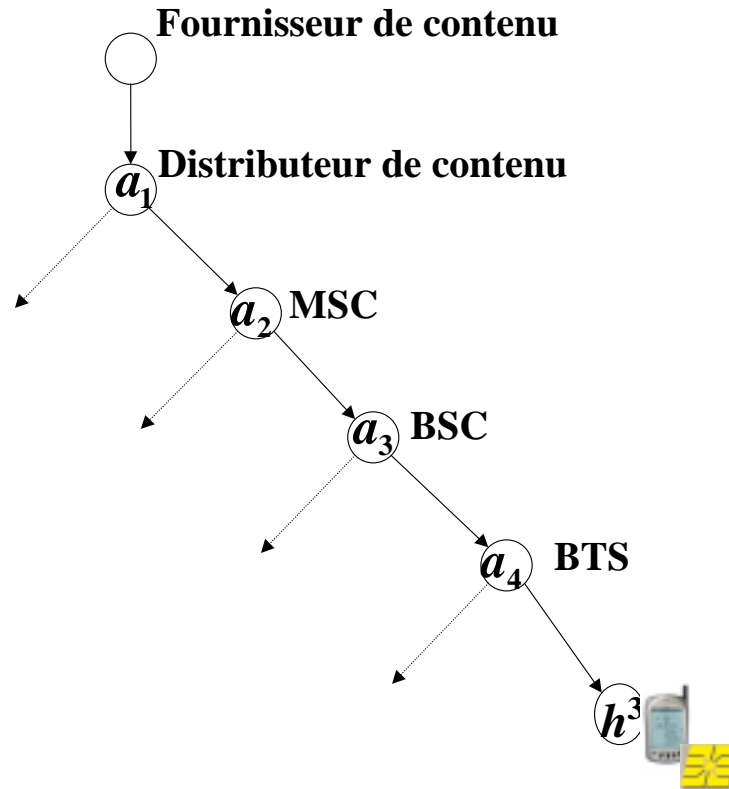


fig. 3.3.1 – Diffusion multicast dans un réseau GSM

La figure fig. 3.3.1 montre l'adaptation du système de cryptage multicast à un réseau GSM. La norme GSM prévoit le cryptage sur l'interface air (entre le BTS¹ et le MS²), mais laisse le choix sur les interfaces A entre le MSC³ et le BSC⁴ et A bis entre le BSC et le BTS. L'opérateur du mobile n'est pas nécessairement responsable du chemin de bout en bout des données vers le mobile, c'est en tenant compte de cette contrainte qu'ont été conçus les protocoles d'authentification à base de HLR⁵ et VLR⁶ : le secret conduisant à l'authentification du mobile et que connaît le HLR n'est jamais transmis au VLR. Nous allons calquer cette gestion de la mobilité pour offrir un service de gestion des secrets qui passe à l'échelle.

¹ Base Transceiver Station, station transducteur de base

² Mobile Station, station mobile

³ Mobile Switching Center, centre de commutation du réseau mobile

⁴ Base Station Controller, contrôleur de la station de base

⁵ Home Location Register, centre d'enregistrement fixe

⁶ Visitor Location Register, centre d'enregistrement délocalisé

i. Abonnement à un groupe

Un mobile qui souhaite s'abonner à un groupe fait changer le paramètre a_4 dans l'exemple de la figure fig. 3.3.1. Il reçoit la fonction h^3 lui permettant de décrypter la clef de contenu. Envoyer cette fonction de déchiffrement à des mobiles ne dépendant pas de la même station de base serait caduque, car elle ne sert pas à décrypter les données que ces mobiles reçoivent. Les seuls mobiles affectés par l'abonnement à un groupe dépendant d'une station de base sont donc ceux rattachés à la station de base. Ils doivent tous recevoir la nouvelle fonction de décryptage h^3 qui, dans le cas d'une séquence paramétrée symétrique dans laquelle nous nous sommes placés, peut se déduire de la précédente par $h^3 \equiv (h_0^3)^{\frac{a_4}{a_4}} \pmod{p}$.

ii. Désabonnement d'un groupe

De même, un mobile souhaitant se désabonner provoque le changement du même paramètre a_4 , il ne reçoit pas la nouvelle fonction h^3 qui permet de décrypter les données suivantes.

iii. Gestion de la mobilité

Les deux primitives présentées paragraphes i. et ii. permettent d'envisager sereinement la gestion de la mobilité. L'abonnement et le désabonnement étant n'ayant un impact que local, le *handover* d'un mobile peut alors être considéré comme un désabonnement suivi immédiatement d'un abonnement.

Chapitre IV - Article

Dans le Chapitre III -, nous avons donné une vue globale de notre architecture de distribution du contenu en fonctionnement. Ce chapitre reprend l'article rédigé sur le modèle de gestion de clefs en première partie et l'architecture conçue pendant le stage en deuxième partie.

Multicast Access Control

AP, GT, RM

Part 1

1 Introduction

1.1 Definitions

We call *recipients*, the network entities that receive IP-multicast packets, such as PC's or mobile devices. A *recipient group* describes all the recipients that have subscribed to a particular multicast group. It is only limited by multicast routing restrictions such as the TTL field.

In the context of secure multicast content distribution, our goal is to allow only selected recipients to access the content of the IP-multicast packets. We call these selected recipients *members* and together they form a *member group*. Thus the recipient group is not limited by IP-multicast protocol but rather by cryptographic mechanisms. We say that a recipient *joins* the member group when he becomes a member and we say that a member *leaves* the group when he loses the privilege to access the distributed content.

Finally we will call *source* an entity which multicasts content over the network.

1.2 Scenario

Multicast access control is a generic term that encompasses many possible scenarios each with their constraints and possible solutions. Here we will consider multicast groups with one or very few sources and a very large member group where members join and leave throughout the lifetime of the group. This setting fits many envisioned commercial large-scale applications [XXIV], the most striking example of which is pay-TV.

Furthermore, as opposed to satellite broadcasting, we don't assume that the recipients have tamper proof hardware that is trusted by the content distributor. Since we will be distributing security parameters to members to allow them to access the distributed content,

this poses a specific problem. If the member group grows to be large enough there are likely to be security exposures. These exposures may be done intentionally by a member: he could send his security parameters to another recipient, or they may be unintentional: a hacker may steal the parameters held by a member and publish them in a newsgroup or on a web page. Thus, one of the key ideas of this work is that we need to limit the impact of security exposures, a property that we call *containment*.

2 Multicast access control.

Providing access control mechanisms in the context of multicast groups as described in the scenario above requires a scalable key distribution algorithm. The goal of a multicast key distribution algorithm is distribute or update the access parameters held by members to enforce:

- Restricted access: only members of the group can access the distributed data.
- Join and leave secrecy: new members should not access past data and former members should not access future data.

The main difficulty is to provide a key distribution that scales well, even in a large dynamic multicast group [XXV].

While a good key distribution algorithm is necessary, it is not sufficient. In fact, we highlight 3 layers in a multicast access control scheme:

- Data Distribution (**DD**): this layer describes how the protected data is sent to the multicast group.
- Key Distribution (**KD**): this layer describes how the access parameters are used by the entities in the network to provide multicast access control.
- Membership Management (**MM**): this layer describes the entities that compute the access control parameters and manage the member group.

The description of these three layers, summarized on Figure 1, is the focus of this section. It is followed by a discussion about the trust level that we expect from the different entities that participate in a multicast access control scheme.

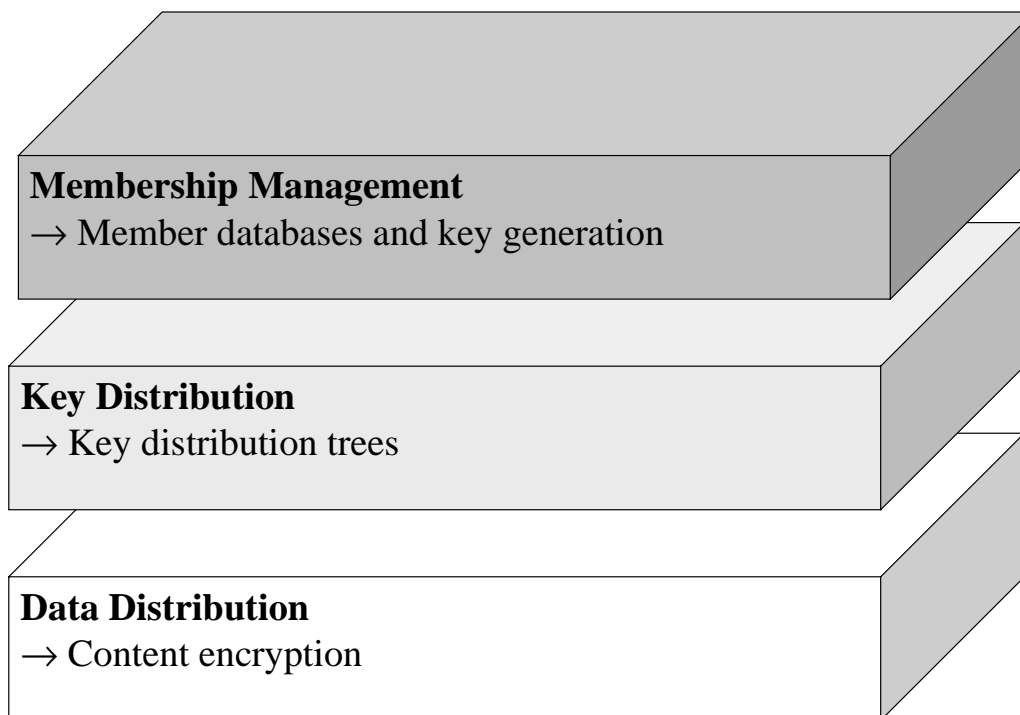


Figure 1

2.1 The Data Distribution Layer

The Data Distribution (DD) layer describes how multicast data is protected to restrict access to the set of current legitimate members. Data is thus encrypted with a key, usually called *TEK* for *Traffic Encryption Key*. The TEK must only be known by the set of current members. If a recipient joins the set of recipients he should not have access to past data, this requirement is sometimes referred to as join secrecy. Similarly, we define leave secrecy: a member leaving the member set should not have access to future data. As a consequence, the TEK must be changed each time the member set is modified to preserve join and leave secrecy.

Moreover, as said earlier, a large group is likely to introduce security exposures. It is important to limit the impact of TEK exposures, by providing containment measures. The extreme scenario occurs when legitimate members retransmit their TEK to non-members. There are at least two ways to deal with this problem:

- Limit the lifetime of the TEK: if the TEK is changed often it limits the damage generated by the exposure of a TEK.

- Limit the scope of the TEK: members receive data that is encrypted with a TEK that depends on their location in the network.

While the second method offers higher security than the first, in practice, it is also more difficult to implement. If the source multicasts a packet how can the members receive the same packet with a different encryption? The solution is to put the intermediate elements in the network to contribution such as proxies or intelligent routers. Indeed, if the intermediate elements in the network change the TEK used to encrypt the data, each cluster or sub-net of recipients will require a different TEK to access the data. Since this solution can turn out to be expensive, we will propose another alternative in the next section.

2.2 The Key Distribution Layer

The key distribution (KD) layer of an access control framework in a large multicast group covers the mechanisms and entities used to distribute and update the TEK used in the DD layer. Often, the TEK is itself encrypted with one or more auxiliary keys, usually called KEK, or Key Encryption Keys[XXVI].

i. The two main characteristics of the KD layer.

The main challenge is to provide a scalable key distribution algorithm which guarantees leave secrecy. This problem has been extensively described in all major proposals for multicast key distribution [XXV][XXVI][XXVII][XXVIII]. Briefly, it's easy to understand the problem if we imagine a naïve key distribution algorithm where an entity distributes the TEK to each member of the group directly. When a recipient joins the group, it's possible to provide join secrecy if the new TEK is encrypted with the old one and multicast to the group (the old TEK serves as a KEK). The key distribution entity simply sends the new TEK to the new member with a secure unicast connection. Providing leave security, once a member leaves is much more cumbersome because it would require the key distribution entity to encrypt the new TEK with the keys of each individual remaining member in the set. This would require a different KEK for each individual member: such an approach does not scale at all. Thus our first need is a scalable key distribution algorithm.

The second characteristic of key distribution is that it requires a certain degree of reliability. If a member cannot update his TEK successfully then he will not be able to access the data layer until the TEK is updated again. The consequences of failures in the delivery of key update messages should be as small as possible.

ii. Containment

Notice that if we have a scope limited TEK, as described previously, we also have reduced reliability needs: if a member leaves the group then only the members in the same network scope will require an immediate TEK update while other members will remain unaffected. However, data layer scope limitation is potentially costly since it requires re-encryption of all the data. A less expensive alternative is to provide KEK scope limitation at the KD layer, combined with lifetime TEK limitation at the DD layer. This approach still

requires the use of intermediate elements in the network, however the computational cost is much lower since we only operate on keys instead of the data itself. Even if the TEK is common to all members of a group, it does not provide a means of long term access to the group: the membership is represented by the long term KEK(s) held by a member.

2.3 The Membership Management Layer

The Membership Management (MM) layer describes the entities that manage the list of members, and use the key distribution layer to enforce membership policies.

MM requires the storage of the subscription information of each member thus it requires a storage capacity that is essentially an order of magnitude of the group size. Being able to distribute membership over several entities thus brings a scalability benefit in computational terms. Having many independent MM entities allows them to be placed closer to the members and also limits the impact of a MM entity failure. Grouping members under the authority of a MM entity may also materialize political or legal boundaries. For example a MM manager could manage the members in a country, in a region or under a certain ISP. On the other hand, as opposed to members themselves, it should be clear that MM entities require the highest degree of trust. Thus, we cannot multiply the number of MM entities beyond a certain point without weakening the security of the access control scheme. When a group of entities reaches a certain size there are likely to be intentional or accidental security exposures.

2.4 Entities and Trust

There are different groups of entities that have distinct roles in a multicast access control framework. These groups of entities vary in size and are assigned different levels of trust:

- *The source or content provider:* there is only one or a few of these entities. The source is trusted to encrypt the data with the right keys provided by a *TEK generator* and multicast it.
- *The members:* the number of these entities is potentially huge (any multicast group). As said previously, security exposures should be taken into account.
- *The membership manager(s):* there should be a manageable number of these entities (one for each region, ISP, etc.) We need to trust these elements.

Additionally, to provide containment, we also need:

- *Intermediate elements* (intelligent routers, proxies).

The trust we put in the intermediate elements is an interesting question. If the content provider owns the infrastructure, then perhaps we can trust these intermediate elements almost as much as the source or the membership manager. But if a complete proprietary

infrastructure is required for each content distributor, this renders the cost of multicast content distribution prohibitive. Even in more traditional environments, such as telephone company networks, where the infrastructure can be considered secure, it is hard to imagine that the source and all recipients will be using the same network in today's global business model. The alternative is to imagine a different model where intermediate elements are much less trusted, and perhaps managed by third parties that are not controlled by the source. This is the approach we take here, where we trust the intermediate elements to make the transformations on the data, yet we do not give these intermediate elements access to the content nor means to distribute membership rights.

3 (CST) Cipher sequence Trees

This section is based on work presented in [XXX]. For further details we refer the reader to that work.

3.1 Preliminary Definitions

Definition 1: (Asymmetric cipher groups) Let N be the product of two carefully selected large prime numbers as in RSA. We define the set of trapdoor permutations $G_N = \{f_a : Z_N^* \rightarrow Z_N^*, x \rightarrow x^a \bmod N, a \in Z_{\phi(N)}^*\}$. For convenience we will index the functions of this set by the exponential $a \in Z_{\phi(N)}^*$ that describes them. We will assume that these sets have the following security properties:

- Given a random index a , it is computationally infeasible to find an index b such that $f_a \circ f_b = Id$ with any non-negligible probability.
- Let $x \xleftarrow{Rand} Z_N^*$. Given a set $A = \{a_i, 0 < i \leq k \mid \forall i \neq j, \gcd(a_i, a_j) > 1\}$ and the set $Y = \{y_i, 0 \leq i \leq k \mid y_i = f_{a_i}(x)\}$ it is computationally infeasible to compute x with any non-negligible probability.

This set forms a an Abelian group through the composition operation, since:

- For all $(f, g) \in (G_N)^2$, $(f \circ g) = (g \circ f) \in G_N$.
- For any $f \in G_N$ there exists a g such that $f \circ g = g \circ f = Id$.

We call such a group, an (RSA based) *asymmetric cipher group*.

Definition 2: (Asymmetric cipher sequences and reversing functions) Let $(F)_1^k = (f_{a_1}, \dots, f_{a_k})$ define a sequence of functions in an asymmetric cipher group. For any $x \in Z_N^*$ we can define a secondary sequence $(F[x])_0^k = (S_0, \dots, S_k)$ in Z_N^* as follows:

$$S_0 \leftarrow x$$

$$S_i \leftarrow f_{a_i}(S_{(i-1)})$$

We call (F) an *asymmetric cipher sequence* in G_N . To describe the secondary sequence $(F[x])$ we will simply say that (F) is instantiated by x . An interesting property of any asymmetric sequence (F) is that it can be associated to a *reversing function* h such that $h \circ (f_{a_1} \circ \dots \circ f_{a_k}) = Id$. For any instance $(F[x])_0^k = (S_0, \dots, S_k)$ of (F) , the knowledge of the reversing function $h \in G_N$ allows to compute x from S_k , since $h(S_k) = S_0 = x$.

Given a sequence of function in an (RSA based) asymmetric cipher group $(F)_1^k = (f_{a_1}, \dots, f_{a_k})$ indexed by their exponents, the reversing function $h \in G_N$ can be computed as $h(x) = x^B \bmod N$ where $B \cdot (\prod_1^k a_i) = 1 \bmod \varphi(N)$

3.2 Key Distribution Trees.

Definition 3: We say that 2 finite sequences $(S_i)_1^l, (T_i)_1^k$ have a *common prefix* of length c if there exists a $k > 0$ such that $S_i = T_i$ for all $i \leq c$.

Definition 4: We say that 2 finite sequences $(S_i)_1^l, (T_i)_1^k$ are *distinct* if there exists a $d > 0$ such that $S_d \neq T_d$.

Assume we have a set of asymmetric cipher sequences (F_1, \dots, F_n) which verifies both the following properties:

- For all (i, j) such that $i \neq j$, F_i and F_j are distinct.
- For all (i, j) such that $i \neq j$, F_i and F_j have a common prefix.

Then we can represent these asymmetric cipher sequences as a tree with each node representing a function and which branches each time at least two sequences differ. This tree has as many leaves as they are cipher sequences, and naturally we can associate a distinct reversing function to every leaf of the tree as shown on Figure 2.

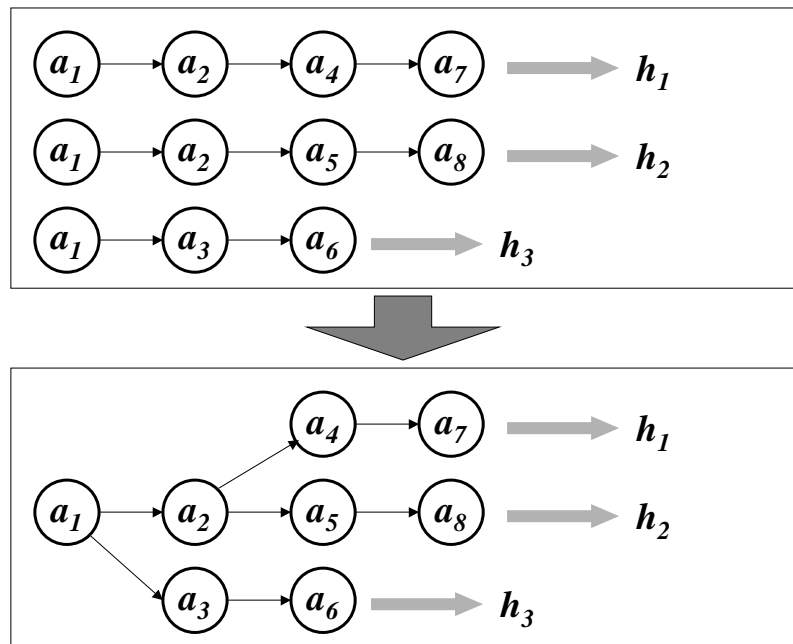


Figure 2

Such a tree can be mapped over a multicast network to provide the basis of a multicast key distribution framework with KD level scope limitation:

- The root of the tree is a short term TEK generator.
- The other vertices of the tree represent intermediate elements in the network such as proxies or intelligent routers.
- The members are divided in subgroups and each one of these subgroups is associated to a leaf in the tree. For example a subgroup could be a set of hosts on the same sub-net, and the leaf vertex could be an IGMP router.

Given an asymmetric cipher group G_N , we will assume that each vertex v_i in the key distribution tree securely receives a randomly chosen function $f_{a_i} \in G_N$, whereas each member in a subgroup receives the reversing function corresponding to the leaf they are associated to. To distribute a key K , we proceed as follows:

- The root encodes K in a special format we will describe later denoted $[K]$ for now. It sets $S_0 = [K]$ and computes $S_1 = f_{a_0}([K])$ and sends it to its children in the tree.
- When a vertex v_i receives a value S it computes $S' = f_{a_i}(S)$ and forwards the new value to its children.

- Leaf vertices proceed the same way but instead they forward the result to the members in the attached local subgroup. These members then use the reversing function they hold to recover $[K]$, and finally K .

The reversing function can thus be considered as a long term KEK that is used to distribute short term TEK. The reason why we don't use this framework to distribute data directly instead of a TEK is that the individual RSA computations are too expensive to be applied to data packets. This rules out the use of this framework to provide DD level scope limitation. However, this framework provides KD level containment because each subgroup uses a different (with very high probability) reversing function to access the TEK. The exposure of a TEK is thus limited to a subgroup and provides no practical means to access the TEK for other members outside that subgroup.

The distributed TEK is used to encrypt the protected content at the DD level. The root of the tree is the TEK generator and it should distribute the generated to the content producer. The content producer will then encrypt the data and use traditional multicasting to send it through the network. It is important to realize that the multicast network tree is potentially completely different from the KD tree described above.

The KEK requires some formatting, denoted $[K]$ above, because the use of n bit RSA trapdoor permutations is not sufficient to guaranty semantic security of the generated k bit key. We suggest that the TEK generator generates a random n bit value r and sets $K=H(r)$ and $[K]=r$ where $H : \{0,1\}^n \rightarrow \{0,1\}^k$ is a one way hash function assumed to be 'ideal' [XXIX].

3.3 Key Distribution in a Dynamic Group.

The previous description did not take member group dynamics into account neither did it describe how the cipher group elements were distributed to the nodes in the tree. Distributing these elements is the role of the Membership Managers (MM). Members are added and removed from a group, nodes are created and updated by interacting with a MM. As we will show later, these MM entities can be arranged in order to each manage a sub-tree of the key distribution tree.

When a recipient wishes to join the group it connects to the local MM which manages the sub-tree in which the recipient is to be located. (Perhaps there is a central MM or directory, which redirects the recipients to the right local MM). There are two possible scenarios:

- (1) If there is no leaf close to the joining recipient, the tree is expanded by adding one or several nodes. The MM securely distributes a randomly chosen cipher group element to each one of these new nodes. Concretely the local KM will send both the index a_i of the function in G_N as well as N itself, which is sufficient to describe $f_{a_i} \in G_N$.
- (1) If the joining recipient is already in a leaf (example: a sub-net) of the tree, then the MM sends a message to the leaf node to update the cipher group element it holds to a new

randomly chosen cipher group element in G_N . Concretely it only needs to transmit the new index a_i (the new RSA exponent).

With the knowledge of the cipher elements in the sub-tree it manages and some auxiliary information we will describe later, the MM computes the new reversing function corresponding to the leaf in which the recipient is attached. This reversing function h is sent with secure unicast to all the members of the leaf. Concretely the MM will send the index of $h \in G_N$ and eventually the value N itself. Once the joining recipient receives h he becomes a member and uses the pseudo-KEK h to retrieve subsequent TEK. Because the new value of h is independent of the previous one this operation provides join secrecy.

When a member in a leaf L is required to leave the group, we proceed similarly to a join operation. The function in L is changed by the MM to a new randomly chosen one and the new corresponding reversing function is sent with secure unicast to the remaining members in the subgroup associated to L , thus providing leave secrecy.

The size of a subgroup is not a scalability issue in our framework, as we can always set an upper limit M on the size of a subgroup by splitting the subgroup and expanding the tree as necessary.

3.4 Membership Management.

The cipher sequence tree construction allows a form of distributed key management where each MM manages its own sub-tree. To compute the reversing function each MM needs to know at least 3 things:

- (1) The index of the cipher group element corresponding to the product of all nodes from the root of the main tree down to the root of the managed sub-tree. This is the auxiliary value referred above.
- (2) An image of the sub-tree with all the associated cipher group element indexes.
- (3) The values N and $\varphi(N)$ needed to compute reversing functions.

The MM can be arranged in a tree, most likely a depth 2 level tree, as shown on Figure 3. The root MM manages the top nodes that are not in any sub-tree and distributes the auxiliary values to the local MM.

Thus our framework fulfils the objectives we listed in section 2 .

Part 2

We showed in first part the advantages of our three-plane model. Its features make it particularly well adapted to content distribution services:

- Content distributor and content provider are distinct entities in real life, which most of architectures do not address unlike our.
- Revocation process is reliable, dynamic and local as required by the distribution of large amount of data to wide-scale groups.

Moreover, such distributed responsibilities adapt well to large-scale mobile networks, where different entities cooperate.

4 Content Distribution

4.1 Background

Several projects address the issue of protected content distribution services. Two of them summarise two different approaches, a proposition of the Mobile Music Forum and the Content Protection System Architecture (CPSA). They use different operational models:

- The Mobile Music Forum basically defines three entities, the content provider, the content distributor and the end user.
- CPSA adds a fourth entity, the device distributor.

In the latter, the user has little control on the device, which raises moral questions as addressed in [XXXI].

i. CPSA

CPSA provides several rules to enforce a content protection policy:

- Content is encrypted on a link-by-link basis on all open interfaces, and no clear text content should be sent.
- Content may only be decrypted by certified tamper-proof hardware, Certificate Revocation Lists (CRL) are issued to revoke certificates that have been compromised.

- Digital rights are written in digital or watermarked Content Management Information (CMI), sent along with data. Only certified tamper-proof hardware may decrypt data and will decrypt the CMI and the CRL altogether.

The main advantage of CPSA over a Public Key Infrastructure (PKI) is the tied link between access to payload and access to CRL. Once the device is granted the right to access data, it updates its CRL. No third part can separate these two streams, and the relevance of a CRL is not subject to a doubt (as it is in PKI).

CPSA suffers from the same main drawback as PKI, the lack of an efficient management of revocation. Revocation through CRL means a growing database of revoked certificates in each device, which is not scalable.

ii. Mobile Music Forum

This project addresses the problem of content distribution in a mobile environment e.g. in a GSM infrastructure. The content distributor manages trusted Single Identifier Modules (SIM) cards in users' mobiles. Content keys and digital rights are sent to SIM cards through a secure channel. After comparing digital rights subscribed by the user and digital rights relating to the content, the SIM card decides or not to send content keys to the mobile, that will decrypt data.

This approach is centralised and relies on a content distribution server able to manage all keys and encrypt data with a different content key for each user, which is not scalable.

4.2 Framework

i. Model

Large-scale content distribution requires separating content and service providers, at least because each service provider may operate many content providers, and each content provider may use many service providers. Moreover, traffic is natively encrypted only on air interface and not on private land networks that are shared among service providers.

Content provider must know secrets that protect content; service provider should not access content. Thus, content keys are to be managed by content provider. All of this makes our model particularly adapted to content distribution.

ii. Data and Key distribution

As no efficient cryptographic algorithm in a multicast environment has been designed yet, we will use symmetric cryptography for bulk encryption.

The source encrypts data with a common secret key K for all members of a group, and changes the key on a regular basis. Belonging to a group in a period of time grants you access to this key during that period.

Each node has its parameter affected by either a local G_l or global G_0 membership manager. Local membership managers will represent the content provider on a local scale, and global membership manager will represent it on large scales. To join a group, a leaf sends a join request to his local membership manager G_l . G_l sends G_0 the product of all the parameters it knows modulo $\varphi(n)$. G_0 multiplies it by the product of all remaining parameters, computes the reversing function h and sends it back to the leaf.

iii. Confinement

Sending enciphered content to large multicast groups has inherent limits. There will always be a final interface that data crosses in clear text. Thus, each member of the group has eventually access to content, and remains *free* to redistribute it. Encrypting data itself instead of content key would only shift this issue. If a group member decides to burn CDs containing the keys he receives and send them to a non-member who had stored the encrypted data, he could straight away burn CDs with clear text content if data is encrypted on a per-member basis.

We will assume that no node has the capacity to efficiently diffuse data to some group or leaves.

We showed in first part that confinement is achieved within a cell i.e. all members that depend on the same leaf apply the same decrypting function $h_{(i,j)}$, but this function is of no use to receivers of other cells since it does not decrypt the encrypted content key they receive.

If a member of the group sends the content key K outside of the cell, he is bound to do it as regularly as K changes, which would mean he owns an efficient diffusion network.

iv. Node Compromise

To compute a reversing function $h_{(i,j)}$, the knowledge of the intermediate parameters $\{a_{i+1} \dots a_j\}$ would not be sufficient as $\varphi(n)$ is also required to compute any exponentiation. Hence, even compromising all nodes from father to son would not give access to the reversing function.

Each local membership manager G_l must be as trusted as the global membership manager G , since they all know $\varphi(n)$. Compromising a leaf and a membership manager would give infinite access to a group since if h^3 defined by $h^3 \cdot a_1 a_2 a_3 a_4 \equiv 1 \pmod{\varphi(n)}$ is

known, if a_4 changes into a_4' , then $h^3 a_4 (a_4')^{-1} \bmod \varphi(n)$ gives access to the content key. The computation of $(a_4')^{-1} \bmod \varphi(n)$ can easily be made through extended Euclidean algorithm.

5 Mobility

The architecture we presented in first section well adapts to content distribution in a GSM-wise network.

5.1 Land Mobile Network

i. Trusting Nodes

Authentication of a Mobile Station (MS) in a GSM network is achieved through a challenge-response scheme. The MS is statically attached to a Home Location Register (HLR) depending on its operator. When it roams, it depends on an other location register that becomes its Visitor Location Register (VLR). To achieve authentication of the MS without comprising secrets, the HLR sends the VLR challenge response pairs. The MS authenticates against the network when it gives the VLR the right answers to its challenge.

Similarly, only few and well chosen entities manage keys in our framework, content distribution nodes are only given partial secrets that do not grant them access to global secret nor to any content. This adapts well to the topology of GSM networks where content may cross many global operators' networks from source to destination.

ii. Trusting Network

Our architecture requires trusting land network to efficiently send messages. We make no particular additional assumption with regards to GSM. The air interface is not required to be reliable to achieve revocation, since no message is sent to the Mobile Station (MS).

This is a main strength of our scheme, since data first reaches the (unsafe) MS before the SIM card.

5.2 Key Distribution

We will use the asymmetric scheme presented in first section. All membership managers know a common secret, $\varphi(n)$, that enables them to compute exponentiations i.e. to manage keys.

i. Architecture

As shown on Figure 4, we designed the architecture to focus on handovers – the way GSM networks handles mobility. Mobile Switching Centres (MSC) represent the network subsystem, Base Station Controllers (BSC) and Base Transceiver Stations (BTS) the Base Station Subsystem. When an MS is about to leave its current cell (the area covered by a BTS), its BSC decides to trigger the change of BTS in charge and if relevant change the BSC and/or MSC.

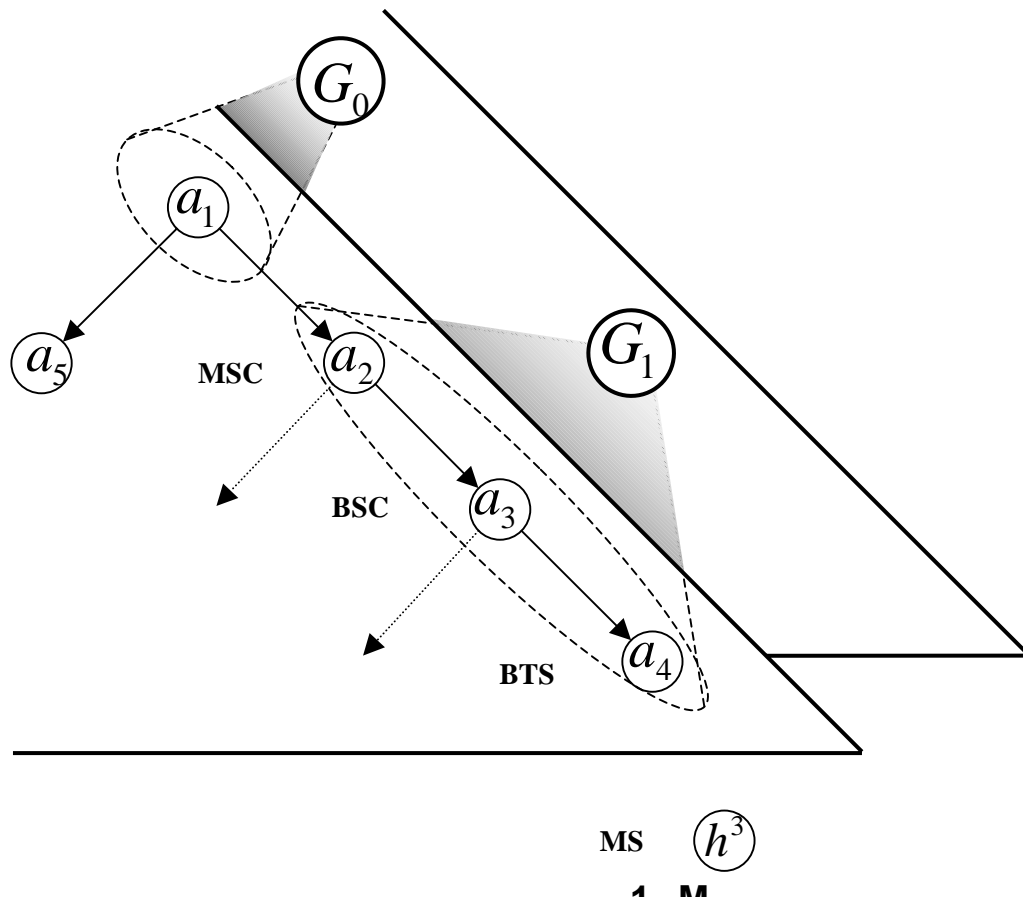


Figure 4

Both local and global membership managers are equally trusted and know $\varphi(n)$, but they store different (distributed) data about group members.

ii. Setup

Each node of the multicast tree is given a RPS parameter. A membership manager is given keys of all nodes below its level in its branch of the multicast tree, down to and excluding the level of the next membership manager.

iii. join and leave Primitives

We suppose that the MS has been authenticated against the network through usual GSM procedure. To join a group, an MS sends a join request to his local membership manager G_l . G_l sends G the product of all the parameters it knows modulo $\varphi(n)$. G multiplies it by the product of all remaining parameters and the reversing function h it sends back to the MS.

The MS has nothing particular to do to leave the group. If it does not receive any update of h , it will not be able to decrypt the group key anymore.

5.3 Handover

Although the very first join involves the global membership manager, most of next subsequent join and leave use only the local membership manager. The GSM protocol has made the same assumption about mobility patterns of users, since the BSC (and not the MSC) gather data relevant to trigger handovers.

i. Intra BSC and intra MSC handover

The BSC informs the local membership manager that a handover will occur and gives the identity of the next BTS the MS will depend on. The local manager locally computes the new reversing function (it is able to do so since it knows the former reversing function, each relevant parameter and $\varphi(n)$) and sends it to the MS.

ii. Inter MSC handover

The MSC initiates a new join procedure that involves the global membership manager.

iii. Performance Evaluation

Each handover requires the exchange of the following messages:

- 2 messages sent to update the value in each last node on the path,
- At most $2(M - 1)$ messages sent to the current members in each leaf,
- 1 message sent to the moving MS, built from scratch in case of an intra BSC or intra MSC handover, requires at most 3 messages otherwise.

A handover operation thus requires at most $2M + 4$ message exchanges.

Further detailed studies of the performance evaluation and of real life constraints are necessary (e.g. frequency of handovers, proportion of inter MSC handovers).

Chapitre V - Évaluation de performances

Dans ce chapitre, nous nous attacherons à évaluer l'efficacité de notre architecture de distribution du contenu. Nous nous placerons dans le cas d'un réseau mobile et tâcherons d'évaluer le trafic engendré par l'impact des changements de fonction de déchiffrement au cours du temps. Nous utiliserons un modèle Markovien [XXXII] pour rendre compte des déplacements du mobile.

1 Modèle

1.1 Position du mobile

On considère un sous-système radio constitué de cellules placées selon une topologie linéaire. Chaque cellule est représentée par son BTS (note 1 page 34). L'ensemble des cellules est partitionné en sous-ensemble de $2D+1$ cellules placées sous l'autorité d'un gestionnaire local d'appartenance au groupe, lui-même dépendant d'un gestionnaire global comme représenté fig. 1.1.1.

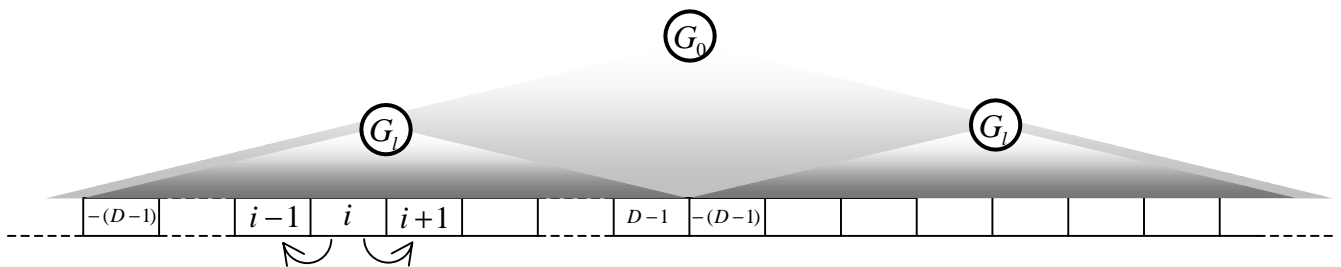


fig. 1.1.1 – Marche du mobile

Deux cellules sont considérées comme voisines si un mobile peut se déplacer de l'une vers l'autre sans en traverser une troisième. Nous travaillons dans le cadre d'une topologie linéaire dans laquelle les cellules i et $i+1$ sont voisines. Ainsi, un mobile présent dans la cellule i ne peut que se déplacer vers les cellules $i-1$, $i+1$ ou rester dans la cellule i .

Nous travaillerons en temps discrétisé et considérerons qu'un mobile ne peut effectuer plus d'un déplacement en un slot de temps. On supposera aussi que les mouvements sont stochastiques et indépendants d'un mobile à l'autre.

1.2 État du mobile

Les états accessibles par le mobile sont représentés fig. 1.2.1 :

- état stationnaire (S) ;
- mouvement à droite (R) ;

- mouvement à gauche (L).

Supposons qu'un mobile est dans la cellule i au début d'un slot. Son mouvement pendant ce slot peut se faire comme suit :

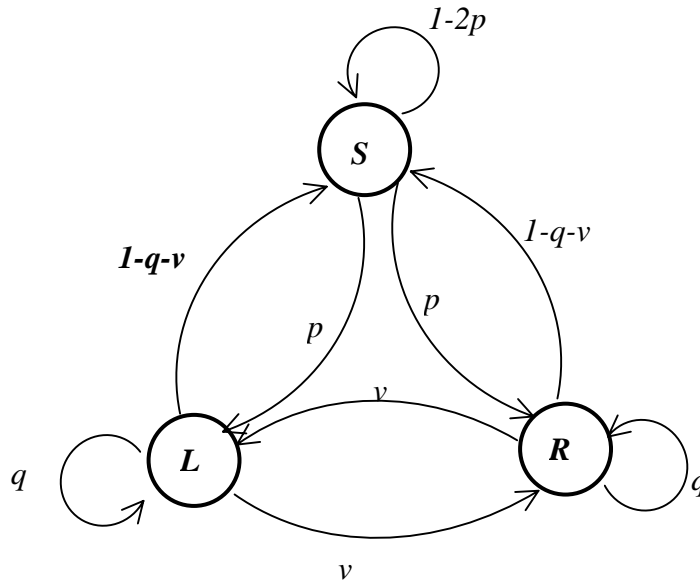


fig. 1.2.1 – État du mobile

- si le mobile est dans l'état S, il reste dans la cellule i ;
- s'il est dans l'état R, il se déplace dans la cellule $i+1$;
- s'il est dans l'état L, il se déplace dans la cellule $i-1$.

Soit $X(t)$ l'état du mobile pendant le slot t . Nous supposons que $\{X(t), t = 0, 1, 2, \dots\}$ est une chaîne de Markov à temps discret de probabilités de transition $p_{k,l} = P[X(t+1) = l | X(t) = k]$ définis comme suit : $p_{R,R} = p_{L,L} = q$, $p_{L,R} = p_{R,L} = v$, $p_{S,R} = p_{S,L} = p$, $p_{L,S} = p_{R,S} = 1 - q - v$ et $p_{S,S} = 1 - 2p$.

Nous disposons donc d'un jeu de paramètres $\{p, q, v\}$ propres à rendre compte des schémas de mobilité du mobile et donc de la nécessité de mettre à jour sa fonction de décryptage de la clef de contenu h .

Chaque changement de cellule entraîne une mise à jour de sa fonction de décryptage h . Le changement est local à l'intérieur du domaine du gestionnaire local, global sur sa frontière : on note D la distance du centre de cette zone à une de ses frontières, par commodité on numérotera donc les cellules de 0 à $D-1$. L'évaluation de performance porte sur les deux points suivants :

- un mobile d'abscisse inférieure en valeur absolue à $D-1$ et qui change de cellule entraîne une mise à jour de sa fonction de décryptage h au niveau du gestionnaire local ;
- un mobile d'abscisse $D-1$ (resp. $-(D-1)$) voulant se déplacer vers la droite (resp. vers la gauche) restera à $D-1$ (resp. $-(D-1)$) mais provoquera une procédure de mise à jour de h au niveau du gestionnaire global.

2 Probabilités stationnaires

2.1 Chaîne de Markov

Soit $Y(t)$ l'abscisse de la cellule dans laquelle le mobile est à l'instant t .

$\{(Y(t), X(t)), t = 0, 1, 2, \dots\}$ est une chaîne de Markov irréductible apériodique :

- la connaissance de l'état et de la position du mobile à l'instant t est suffisante pour déterminer celle à $t+1$;
- tous les états sont accessibles à partir de n'importe quel état ;
- il existe des cycles de longueur unité (tous ceux pour lesquels $X(t) = S$), le plus grand commun diviseur de toutes les longueurs de cycle vaut donc un.

Ce modèle est donc justiciable des équations de Chapman-Kolmogoroff en temps discret pour établir sa distribution de probabilités stationnaires.

2.2 Probabilités stationnaires

On désigne par $Q_{d,x} = \lim_{t \rightarrow \infty} P[Y(t) = d, X(t) = x]$, $d = 0, 1, \dots, D-1, x \in \{S, R, L\}$ la distribution de probabilités stationnaires de cette chaîne de Markov.

Prenons le mobile dans l'état (d, S) et considérons les moyens pour lui d'arriver dans cet état. Il pouvait venir du même état (d, S) et avec une probabilité $1-2p$ rester S . Il pouvait aussi venir de l'état $(d-1, R)$ ou $(d-1, L)$ et devenir S avec une probabilité $1-q-v$. On fait le même type de raisonnements avec les états (d, R) et (d, L) pour aboutir aux équations d'équilibre :

$$\begin{cases} 2pQ_{d,S} = (1-q-v)(Q_{d-1,R} + Q_{d+1,L}) & 0 \leq d \leq D-2 \\ Q_{d,R} = pQ_{d,S} + qQ_{d-1,R} + vQ_{d+1,L} & 0 \leq d \leq D-2 \\ Q_{d,L} = pQ_{d,S} + vQ_{d-1,R} + qQ_{d+1,L} & 0 \leq d \leq D-2 \end{cases}$$

Equation 1

sachant que par raison de symétrie, $\begin{cases} Q_{d,R} = Q_{-d,L} \\ Q_{d,S} = Q_{-d,S} \end{cases}$.

Les conditions aux limites peuvent s'énoncer

- si le mobile est dans l'état $(D-1, S)$, c'est qu'il vient
 - ou bien de ce même état,
 - ou bien de $(D-2, R)$, il vient de se déplacer vers la droite,
 - ou bien de $(-(D-1), L)$, il a franchi la frontière de la zone sous contrôle de son gestionnaire local ;
- si le mobile est dans l'état $(D-1, L)$, il vient
 - ou bien de $(D-1, S)$, il n'a pas bougé
 - ou bien de $(D-2, R)$, il vient de se déplacer vers la droite,
 - ou bien de $(-(D-1), L)$, il a franchi la frontière de la zone sous contrôle de son gestionnaire local ;
- idem avec l'état $(D-1, R)$.

Ce qui s'écrit de manière plus précise :

$$\begin{cases} 2pQ_{D-1,S} = (1-q-v)(Q_{D-2,R} + Q_{-(D-1),L}) \\ Q_{D-1,R} = pQ_{D-1,S} + qQ_{D-2,R} + vQ_{-(D-1),L} \\ Q_{D-1,L} = pQ_{D-1,S} + vQ_{D-2,R} + qQ_{-(D-1),L} \end{cases}$$

Equation 2

La normalisation s'effectue en rajoutant comme condition $\sum_{d \in [-(D-1), D-1]} (Q_{d,S} + Q_{d,R} + Q_{d,L}) = 1$.

3 Résolution

On travaille pour $d \geq 0$. Soustraire les deux premières lignes de Equation 1 donne $Q_{d,R} - Q_{d,L} = (q-v)(Q_{d-1,R} - Q_{d+1,L})$. Une combinaison linéaire des deux premières lignes fournit rapidement $Q_{d+1,L} = \frac{2}{1-q+v} Q_{d,R} - \frac{1+q-v}{1-q+v} Q_{d-1,R}$ et donc $Q_{d,L} = \frac{2}{1-q+v} Q_{d-1,R} - \frac{1+q-v}{1-q+v} Q_{d-2,R}$. En combinant ces trois équations et en tenant compte des valeurs interdites de q et v , on obtient finalement $Q_{d,R} - 2Q_{d-1,R} + Q_{d-2,R} = 0$ et donc une

solution générale de la forme $Q_{d,R} = \alpha d + \beta$. Une équation précédente fournit alors

$$Q_{d,L} = \alpha d + \frac{(1+q-v)(\alpha-\beta)+2\beta}{1-q+v} - \alpha \quad \text{et} \quad \text{une des équations de départ}$$

$$Q_{d,S} = \frac{1-q-v}{p} \left(\alpha d + \frac{(\alpha-\beta)(v-q)-\beta}{1-q+v} \right).$$

Les deux premières lignes de Equation 2 et la condition de symétrie donnent les

$$Q_{D-1,S} = \frac{(1-q-v)}{p} (\alpha(D-2) + \beta)$$

conditions aux limites pour $Q_{d,x}$: $Q_{D-1,L} = \alpha(D-2) + \beta$

$$Q_{D-1,R} = \alpha(D-2) + \beta$$

On exprime les équations de Chapman-Kolmogoroff en 0, cellule la plus à même de refléter la symétrie du processus : $2pQ_{0,S} = (1-q-v)(Q_{1,L} + Q_{-1,R}) = 2(1-q-v)Q_{1,L}$ qui nous fournit directement $\alpha = 0$. L'équation de normalisation donne alors

$$\beta = \frac{p}{(2D-1)(1+2p-q-v)}. \quad \text{En régime stationnaire, on a donc, pour } 0 \leq d \leq D-1 :$$

$$Q_{d,S} = \frac{1-q-v}{(2D-1)(1+2p-q-v)}$$

$$Q_{d,R} = \frac{p}{(2D-1)(1+2p-q-v)}$$

$$Q_{d,L} = \frac{p}{(2D-1)(1+2p-q-v)}$$

4 Coût de mise à jour de l'appartenance au groupe

4.1 Mise à jour par le gestionnaire local

La probabilité de mise à jour de h par le gestionnaire local peut alors s'évaluer à

$$\sum_{d \in [-(D-2), D-2]} (Q_{d,R} + Q_{d,L}) + Q_{-(D-1),R} + Q_{D-1,L} \quad \text{soit} \quad P_l(D) = \frac{4(D-1)p}{(2D-1)(1+2p-q-v)}.$$

Pour δD petit devant D (lui-même grand devant 1), $\frac{P_l(D+\delta D)}{P_l(D)} \approx 1 + \frac{\delta D}{2D^2}$: on augmente le trafic d'un

facteur du deuxième ordre $\frac{\delta D}{2D^2}$ en augmentant la taille des domaines des gestionnaires locaux de δD .

4.2 Mise à jour par le gestionnaire global

La probabilité de mise à jour par le gestionnaire global s'évalue par $Q_{D-1,R} + Q_{-(D-1),L} = 2Q_{D-1,R}$ soit $P_0(D) = \frac{2p}{(2D-1)(1+2p-q-v)}$, assez logiquement décroissant quand D croît. Pour δD petit devant D , $\frac{P_0(D+\delta D)}{P_0(D)} \approx 1 - \frac{\delta D}{D}$: on gagne un facteur du premier ordre en $\frac{\delta D}{D}$: en comparant avec le paragraphe 4.1 il apparaît profitable d'augmenter la taille des domaines. Il faut garder à l'esprit qu'une telle augmentation va aussi dans le sens d'un plus grand centralisme, ce que notre architecture vise à limiter.

Supposons que nous avons n mobiles se déplaçant aléatoirement et sans interaction, il y en a $n \sum_{x \in \{S,R,L\}} Q_{d,x} = \frac{n}{2D-1}$ dans la cellule numérotée d – on n'est pas surpris de trouver ce résultat dans la mesure où les expressions de $Q_{d,x}$ nous disent que les états sont équidistribués selon d .

Par *handover*, le trafic engendré vaut :

- 2 messages locaux pour les deux BTS concernés (d et son voisin) ;
- $2\left(\frac{n}{2D-1}\right) - 1$ envois locaux des nouvelles clefs aux membres du groupe dans la cellule d ;
- 1 message local ou global de mise à jour de la clef du membre en cours de *handover*.

On dénombre au total $2nP_l(D) + nP_l(D)\left(2\left(\frac{n}{2D-1}\right) - 1\right) + n(P_0(D) + P_l(D))$ messages sur l'interface air, $2nP_0(D)$ messages échangés avec le gestionnaire global.

4.3 Perspectives

Nous disposons donc d'un modèle paramétrable qui rend compte du coût de signalisation engendré par notre architecture de distribution de clefs de contenu. Pour que cette évaluation de performance prenne tout son intérêt, il serait nécessaire d'appliquer ce modèle markovien à d'autres architectures de distribution du contenu et comparer les résultats obtenus.

5 Traitement par renormalisation d'un modèle de percolation dirigée

5.1 Modèle de percolation

Le XX^e siècle a vu une avancée considérable dans la connaissance des polymères, ces macromolécules aux propriétés inhabituelles.

On a commencé par les traiter avec un formalisme de type gaz parfait (molécules de volume nul et sans interaction), avant de bâtir un modèle géométrique qui pouvait s'appliquer à bien d'autres phénomènes physiques, le modèle de percolation.

À l'image de l'eau percolant à travers les grains de café, on considère que les liaisons chimiques s'établissent dans le réseau de monomères de manière probabiliste, jusqu'à former des amas de polymères. Les outils même du traitement mathématique de la percolation ont grandement évolué, de l'approche classique de type *champ moyen* où l'on moyenne l'action de l'ensemble des entités sur chaque entité cible, vers celle du groupe de renormalisation (début des années 70) qui prend en compte plus justement l'effet de changements microscopiques sur les propriétés macroscopiques et décrit ainsi de manière plus satisfaisante les phénomènes de transition critique.

Dans le cadre de l'évaluation des performances de notre modèle, on peut considérer que le mobile percole à travers le réseau fixe. Aux faibles probabilités de traversée d'un lien, les changements de cellule ne seront qu'occasionnels et n'engendreront pas un trafic de mise à jour de la fonction de décryptage important. Aux fortes probabilités, la mise à jour de la fonction de décryptage atteint la même fréquence que celle de la mise à jour de la clef de contenu. Nous chercherons ici à déterminer un équivalent du coût de mise à jour au niveau de la **transition** entre ces deux régimes.

i. Percolation par liens

On considère un réseau régulier de N sites de nombre de coordination z . Le réseau comporte donc $\frac{Nz}{2}$ liens, il admet $2^{\frac{Nz}{2}}$ configurations notées C . Cherchons le poids de C :

$w(C) = p^{n_{\text{liens occupés}}} (1-p)^{\frac{Nz}{2} - n_{\text{liens occupés}}} = p^{n_0} (1-p)^{n-n_0}$. On peut alors calculer la probabilité que i et j soient reliés : $P_{ij}(p) = \sum_{C: i \rightarrow j} w(C)$.

En utilisant la fonction caractéristique σ telle que $\sigma_l = 1$ si le lien l est occupé, 0 sinon, $C = \left\{ \sigma_1, \dots, \sigma_{\frac{N_z}{2}} \right\}$ et $w(C) = \prod_{i=1}^{\frac{N_z}{2}} ((1-p)(1-\sigma_i) + p\sigma_i)$ avec la propriété $\sum_{\sigma_1=0} \dots \sum_{\sigma_i=0} \dots \sum_{\sigma_{\frac{N_z}{2}}=0} w(C) = 1$.

ii. Grandeurs caractéristiques

On peut montrer l'existence d'une transition critique dans ce modèle géométrique de percolation [XXXIII], c'est à dire l'existence d'une probabilité p_c à partir de laquelle un ensemble de couples $\{i, j\}$ *infiniment* distants non reliés pour des valeurs plus petites de p deviennent brusquement reliés :

- si $p < p_c$, on étudie la limite de $P_{i,j}(p)$ quand $|i - j| \rightarrow \infty$, elle vaut 0 ;
- si $p > p_c$, $P_{i,j}(p) \rightarrow P_\infty^2(p)$ où $P_\infty(p)$ est la probabilité pour un nœud d'être relié à l'amas infini *i.e.* l'ensemble des nœuds du réseau que le mobile peut atteindre (la puissance 2 vient de ce que l'on va très loin à partir de i et de j).

La valeur du seuil de percolation p_c n'est pas universelle, elle dépend essentiellement de la dimension de l'espace dans lequel est plongé le réseau ainsi que de la géométrie de ce dernier (en deux dimensions : carré, triangulaire, hexagonal...).

Ce seuil peut dans le cadre de notre évaluation de performances être interprété comme la frontière entre deux régimes pour le schéma de mobilité du MS :

- si $p < p_c$, le mobile reste sensiblement sous la juridiction du même gestionnaire d'adhésion ;
- si $p > p_c$, le mobile effectue de *grands* déplacements, il induit beaucoup de trafic.

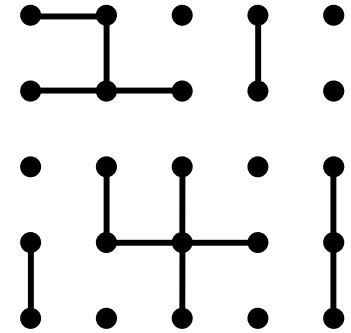
C'est donc au voisinage de p_c que le comportement du mobile nous intéresse.

On définit alors la longueur de corrélation ξ par

$$\begin{cases} P_{i,j} \underset{|i-j| \rightarrow \infty}{\sim} \exp\left(-\frac{|i-j|}{\xi(p)}\right) & \text{si } p < p_c \\ P_{i,j} - P_\infty^2 \underset{|i-j| \rightarrow \infty}{\sim} \exp\left(-\frac{|i-j|}{\xi(p)}\right) & \text{si } p > p_c \end{cases}$$

Intuitivement, c'est la longueur sur laquelle se fait sentir un changement de décision (passer par un lien ou non).

Pour N très grand, la densité d'amas de s sites vaut $\frac{n_s}{N}$ où n_s est le nombre d'amas de s sites. Pour l'exemple donné ci-contre, on compte ainsi $n_1 = \frac{7}{25}, n_2 = \frac{2}{25}, n_3 = \frac{1}{25}, n_4 = 0, n_5 = \frac{1}{25}, n_6 = \frac{1}{25}$ et $n_{s>6} = 0$. Partant de la densité d'amas de sites, on remarque que



$$\begin{aligned} \lim_{n \rightarrow \infty} \left(\sum_{s=0}^n s n_s \right) &= 1 & \text{si } p < p_c \\ &= 1 - P_\infty(p) & \text{si } p > p_c \end{aligned}$$

On utilise le deuxième moment de n_s pour définir ensuite la taille moyenne des amas par $S(p) = \sum_s s^2 n_s$.

iii. Exposants critiques

Dans le modèle de percolation, les grandeurs caractéristiques présentées en ii. présentent une discontinuité pour p au voisinage de p_c . On cherche alors les exposants critiques de ces grandeurs, c'est à dire les exposants qui donnent les équivalents des grandeurs caractéristiques au voisinage du seuil de percolation après la transition critique.

La théorie prévoit que ces exposants sont universels, c'est à dire qu'ils ne dépendent pas de la géométrie du réseau sous-jacent mais seulement de la dimension de l'espace dans lequel le réseau est plongé.

Du fait que l'on ne connaisse pas *a priori* la géométrie du réseau des BTS, il serait hasardeux de vouloir prévoir la valeur de p_c (de toute évidence située entre 0,34, cas du réseau triangulaire, et 0,65, pour l'hexagonal). L'universalité des exposants critiques permet par contre de prévoir le comportement du mobile au-delà de la transition, dès qu'elle est franchie.

Les exposants critiques β , γ et ν sont introduits par

$$- P_{\infty}(p) \underset{p \rightarrow p_c^+}{\propto} (p - p_c)^{\beta} ;$$

$$- \xi(p) \underset{p \rightarrow p_c^{\pm}}{\propto} |p - p_c|^{-\nu} ;$$

$$- S(p) \underset{p \rightarrow p_c^{\pm}}{\propto} |p - p_c|^{-\gamma} .$$

Une fois ces exposants déterminés, on aura donc une évaluation d'une grandeur caractéristique de l'amas infini (ξ), de la taille moyenne des amas (S) et de la probabilité pour un BTS d'y appartenir (P_{∞}).

iv. Percolation dirigée

On améliore le modèle de percolation isotrope en donnant un sens à chaque lien : il peut autoriser le passage du mobile dans un sens mais pas dans l'autre, ou encore dans les deux sens. Chacune de ses possibilités se faisant avec une probabilité (p ou q) définie par avance [XXXIV]. Certains exposants critiques sont alors dédoublés, un suivant chaque direction. On obtient alors des amas de la forme représentée fig. 5.1.1.

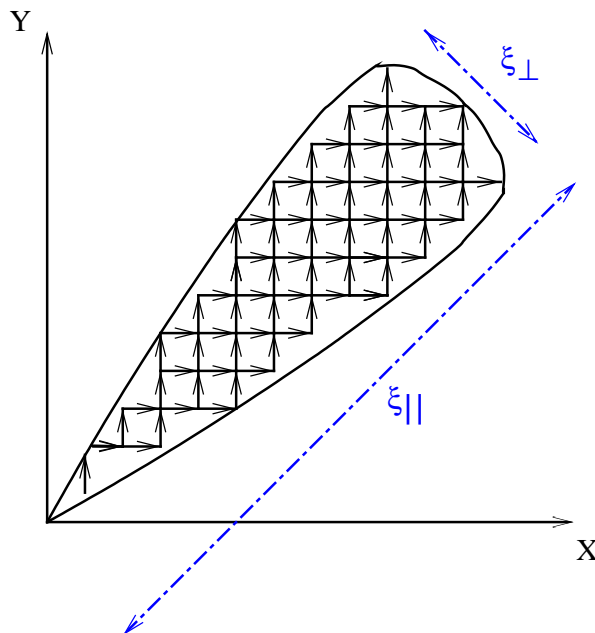


fig. 5.1.1 – Longueurs de corrélation parallèle et perpendiculaire

5.2 Traitement par renormalisation

Un progrès spectaculaire de la mécanique statistique après 1970 a été la construction d'une théorie des phénomènes critiques, à la suite des travaux de K. Wilson. La théorie de Wilson s'est développée à partir du concept d'invariance d'échelle introduit par L. Kadanoff : il n'y a pas d'échelle de longueur caractéristique pour les fluctuations critiques, qui ont le même aspect à toutes les longueurs d'onde (dès qu'on considère des longueurs grandes devant les dimensions atomiques). En termes mathématiques, l'invariance d'échelle est une invariance par un groupe appelé le groupe de renormalisation. À partir de ces concepts, on rend compte des phénomènes critiques, on explique les valeurs numériques des exposants critiques et leur universalité : on peut regrouper les systèmes étudiés en classes, et des systèmes très différents ont les mêmes exposants critiques dès lors qu'ils appartiennent à la même classe. Les théories modernes des changements de phase et des particules élémentaires ont beaucoup de points communs, ce qui a rapproché deux domaines de la physique qu'on aurait pu croire très éloignés l'un de l'autre.

C'est pour justifier théoriquement la remarque de Widom que Kadanoff introduisit en 1966 l'idée fondamentale du groupe de renormalisation et de l'invariance d'échelle, idée directrice développée depuis par de nombreux physiciens, en particulier Kenneth Wilson. Le fait physique fondamental manifesté par une transition de phases est celui de l'existence de fluctuations géantes qui, au point critique, deviennent macroscopiques. Au point critique, la longueur caractéristique du système qu'est la longueur de corrélation diverge et le système ne possède donc plus de longueur caractéristique. C'est dire qu'il est invariant par changement d'échelle. D'où l'idée de considérer le point critique comme un point fixe pour des transformations d'échelles.

La classe de renormalisation qu'on considère ici est celle de percolation. On se propose donc d'utiliser l'invariance par changement d'échelle au seuil de renormalisation pour calculer les coefficients critiques définis en Chapitre I -1.1 iii. .

i. Début d'approche analytique

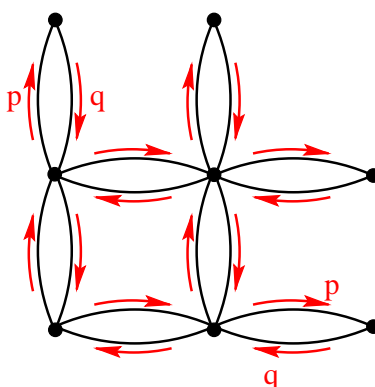


fig. 5.2.1 – Renormalisation d'un facteur 2

fig. 5.2.1 représente une cellule de dimensions 2×2 qu'on se propose de renormaliser en une cellule de dimensions 1×1 . En calculant les points fixes de cette transformation, on

sera alors à même de déterminer les caractéristiques du système au voisinage du seuil de percolation.

On arrive à réduire tout réseau à une combinaison de diodes placées en parallèle, en série ou suivant un pont de Wheatstone qui se réduit comme en fig. 5.2.2. Ces calculs aboutissent à la relation de récurrence suivante :

Si on se place dans le cas $q=0$, en notant $p'=R(p)$ puis sachant que $\xi(p) \underset{p \rightarrow p_c}{\propto} |p - p_c|^{-\nu}$ (Chapitre I -1.1 ii.), on en déduit

$$\begin{aligned} \frac{\xi}{2} &= |R(p) - P_c| \\ &= \left| R(p_c) + (p - p_c) \left(\frac{\partial R}{\partial p} \right) (p_c) - p_c \right| \text{ en développant autour de } p_c \\ &= (p - p_c)^{-\nu} \left(\frac{\partial R}{\partial p} (p_c) - 1 \right)^{-\nu} \end{aligned}$$

et donc $\nu = \frac{\log 2}{\log \left| \frac{\partial R}{\partial p} \right|_{p_c}}$. Ce calcul approché donne $\nu \approx 1,792$, très proche du résultat

numérique avéré approché par 1,733.

Même une renormalisation d'un facteur 2 grossier permet donc d'avoir des résultats intéressants quand aux exposants critiques.

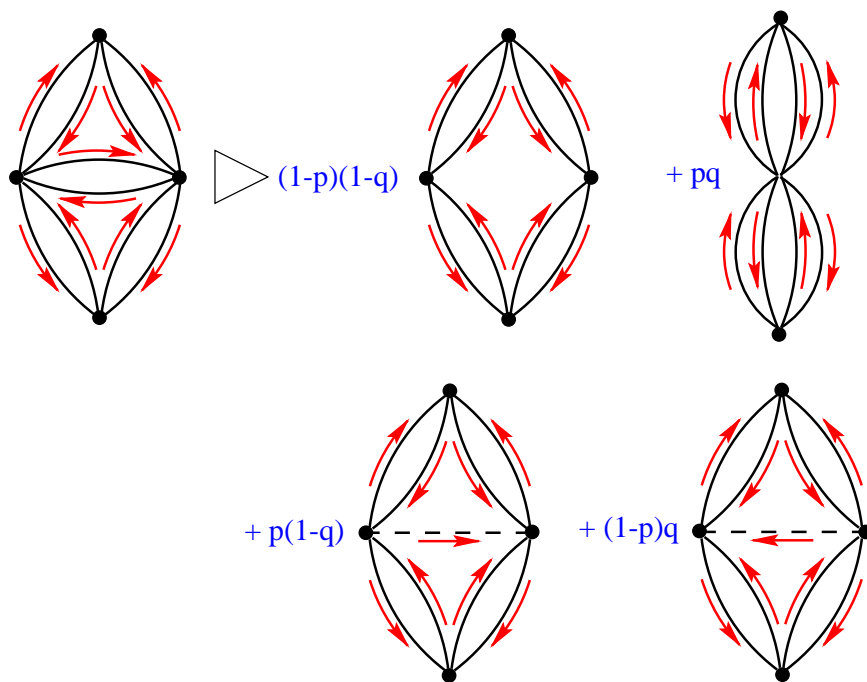


fig. 5.2.2 – Résolution du pont de Wheatstone

ii. Comparaison et perspectives

Nous avons esquissé rapidement un traitement par renormalisation d'un modèle de percolation. Le principal apport d'une telle approche par rapport à celle fondée sur une approche classique markovienne est son insensibilité par rapport à la structure géométrique du réseau sous-jacent. De plus, notre traitement s'intéresse spécifiquement au régime de transition du schéma de mobilité de l'utilisateur, et non pas à tous les régimes. Le point délicat en terme de trafic est en effet non pas quand le mobile ne bouge pas (cas statique), ni quand il change de manière incessante de cellule (la mise à jour de la clef de contenu se synchronise sur celle de la clef groupe), mais bien à la transition entre ces deux régimes.

L'approche par modèle de percolation promet donc non pas de prévoir *quand* il y aura percolation (la valeur de p_c n'est pas universelle), mais *comment* la percolation se fera, en donnant des équivalents des éléments caractéristiques du schéma de mobilité de l'utilisateur au voisinage de la transition critique.

Chapitre VI - Conclusion

L'état de l'art a mis en évidence deux approches différentes à la problématique des services de protection du contenu. La première, essentiellement académique, vise à protéger les droits de l'utilisateur et propose des services inédits par une modulation fine de la notion de propriété intellectuelle : je vends une partie de mes droits sur l'image et la transaction que j'effectue avec l'acheteur vise à lui garantir les droits qu'il vient d'acquérir. La deuxième approche est plus traditionnelle, je vends une donnée multimédia mais pas les droits de reproduction, et je me protège en utilisant du chiffrement ou du tatouage.

Dans le cadre d'une diffusion multicast de données multimédia en push, aucune des solutions proposées n'est satisfaisante essentiellement à cause d'un mauvais passage à l'échelle. À partir d'une modélisation originale de la gestion des différentes clefs afférentes à la distribution d'un contenu, nous avons conçu une architecture opérant par-dessus un réseau mobile de type GSM ayant des capacités multicast.

La gestion en plans de notre modèle le rend particulièrement adapté à la distribution de contenu. Chaque entité est cantonnée à son rôle et ne délègue pas plus d'autorité que nécessaire.

L'architecture qui en découle offre une gestion distribuée de l'appartenance au groupe, l'abonnement et le désabonnement n'ont un impact que local sur l'ensemble du réseau tout en offrant des garanties de limitation d'accès au données au membre du groupe à un instant donné.

Nous avons enfin construit un modèle markovien visant à rendre compte de la mobilité de l'utilisateur, afin de pouvoir évaluer l'impact des différents paramètres du système (dimension des différentes zone, schéma de mobilité) sur le trafic induit. Nous avons aussi esquissé un modèle géométrique de percolation dirigée pour rendre compte de l'impact de la mobilité du membre du groupe sur le trafic occasionné par la gestion des clefs.

Annexe A Calculs d'Évaluation de Performances

■ Initialisation de Mapple

```
[ > restart;
```

■ Évaluation des performances

Les premières équations ont été résolues à la main, on a ainsi abouti à une équation récurrente linéaire à coefficients constants pour $Q(d,R)=QR(d)$, on a ensuite rapidement exprimé $Q(d,L)=QL(d)$ et $Q(d,S)=QS(d)$ en fonction des paramètres d'intégration de $Q(d,R)$.

■ Définition des fonctions $Q(y,x)$

```
[ > QR:=d->alpha*d+beta;
                                QR := d → α d + β
> QL:=d->alpha*d+(1+q-v)*(alpha-beta)+2*beta)/(
1-q+v)-alpha;
                                QL := d → α d +  $\frac{(1+q-v)(\alpha-\beta)+2\beta}{1-q+v}$  - α
> QS:=d->(1-q-v)/(2*p)*(QR(d-1)+QL(d+1));
                                QS := d →  $\frac{1}{2} \frac{(1-q-v)(QR(d-1)+QL(d+1))}{p}$ 
```

■ Vérification des équation de Chapman-Kolmogoroff

```
[ > simplify{QR(d)-QL(d)-(q-v)*(QR(d-1)-QL(d+1))};
                                0
> simplify{QL(d+1)-2/(1-q+v)*QR(d)+(1+q-v)/(1-q+v)*QR(d-1)};
                                0
> simplify{2*p*QS(d)-(1-q-v)*(QR(d-1)+QL(d+1))};
                                0
> simplify{QR(d)-p*QS(d)-q*QR(d-1)-v*QL(d+1)};
                                0
> simplify{QL(d)-p*QS(d)-v*QR(d-1)-q*QL(d+1)};
                                0
```

■ Expression puis résolution de l'équation de Chapman-Kolmogoroff en 0 pour QL

```
[ > eqn := 2*p*QS(0)-2*(1-q-v)*QL(1)=0;
> solve(eqn, {alpha});
{alpha=0}
On affecte à alpha la valeur trouvée par Maple :
> assign(%);
```

■ Expression puis résolutoin de l'équation de Chapman-Kolmogoroff en D-1 (conditions aux limites)

On introduit de nouvelles variables pour $Q(D-1,x)=QxDm1$ car l'expression générale trouvée précédemment ne s'applique pas.

```
[ > eqns := {2*p*QSDm1-(1-q-v)*{QR(D-2)+QRDm1},
QRDm1=p*QSDm1+q*QR(D-2)+v*QRDm1,
QLDm1=p*QSDm1+v*QR(D-2)+q*QRDm1};
eqns := {2 p QSDm1 = (1 - q - v) (beta + QRDm1),
QRDm1 = p QSDm1 + q beta + v QRDm1,
QLDm1 = p QSDm1 + v beta + q QRDm1}
> solve(eqns, {QSDm1, QRDm1, QLDm1});
{QRDm1 = beta, QSDm1 = -frac(beta(q-1+v), p), QLDm1 = beta}
Affectation des résultats...
> assign(%);
```

■ Expression puis résolution de l'équation de normalisation

On se sert des propriétés de symétrie de QS, QR et QL.

```
[ > somme := 2*sum('QS(d)+QR(d)+QL(d)',
'd'=1..(D-2)) +
2*QRDm1+QSDm1+QS(0)+QR(0)+QL(0);
> solve(somme=1, {beta});
{beta = -frac(1, 2 D q - D + D v - 2 D p - q + 1 - v + 2 p)}
> assign(%);
```

■ Affichage des résultats

On essaie d'aider Maple au maximum pour faire afficher un résultat présentable

```
[ > collect(simplify(QS(d)), D);
frac(1, 2 (q - 1 + v - 2 p) D - q + 1 - v + 2 p)
> collect(QR(d), D);
```

```

-1/2 (q-1+v-2p)D-q+1-v+2p
> collect(simplify(QL(d)), D);
-1/2 (q-1+v-2p)D-q+1-v+2p

```

■ Estimation des couts de mise à jour

```

> cout_local=simplify(2*sum('QR(d)+QL(d)',
'd'=1..(D-2))+QR(0)+QL(0)+2*QLDm1);
cout_local=-2 p / (q-1+v-2p)
> cout_global=collect(2*QRDm1, 'D');
cout_global=-p / ((q-1+v-2p)D-q+1-v+2p)

```


Index

- 3**
3GPP 24
- A**
AKE 21
ATA 28
- C**
CPSA
4C-entity 18
CCI 18, 21
CMI 18
CPPM 18
CPRM 18, 19
CPTWG 18
CSS 18
DTCP 18
DTLA 23
DVI 23
HDCP 19, 23
SDMI 18, 19
SRM 22
CST 41
- D**
DD 37, 38
Diffie-Hellman 22
- E**
El Gamal 9, 31
- G**
GSM 34
BSC 34
BTS 34
Handover 30, 35, 58
HLR 34
Interface A 34
Interface A bis 34
MS 30, 34
MSC 34
VLR 34
- I**
IETF 24
RFC 24
- K**
KD 37, 38, 39
KEK 39
- M**
MM 37, 38, 40
Mobile Music Forum 30
MPEG 24
IPI 25
IPMP 24
IPMP-Ds 25
IPMP-ES 25
MPEG-21 25
MPEG-4 24
MPEG-7 25
- O**
OPIMA 23
IPMP 24
OVM 24
- P**
percolation
amas infini 60
exposants critiques 61
longueur de corrélation 61
modèle de percolation 59
renormalisation 63
taille moyenne des amas 61
privacy 7
- R**
RSA 9
- S**
SDMI
LCM 19
PD 19
secrecy
join secrecy 38
leave secrecy 38
SIM 7
- T**
TEK 38
TPH 7

TTP 7, 12

Bibliographie

-
- [I] “*Communication Security*” Refik Molva, cours du DEA RSD, 2001
- [II] “*Handbook of Applied Cryptography*” Alfred MENEZES, Paul VAN OORSCHOT, Scott VANSTONE.. <http://cacr.math.uwaterloo.ca/hac/>
- [III] “*Watermarking schemes evaluation*” Fabien A.P. PETITCOLAS. IEEE Signal Processing Magazine, 17(5) : 58 64, septembre 2000.
- [IV] “*Watermarking digital image and video data*” Gerhard C. LANGELAAR, Iwan SETYAWAN, et Reginald L. LAGENDIJK. IEEE Signal Processing Magazine, 17(5) :20, septembre 2000.
- [V] “*Digital watermarking for dvd video copy protection*” Maurice MAES, Ton KALKER, Jean-Paul M.G. LINNARTZ, Joop TALSTRA, Geert F.G. DEPOVERE, and Jaap HAITSMA. IEEE Signal Processing Magazine, 17(5) :47 57, septembre 2000.
- [VI] “*Watermarking schemes and protocols for protecting rightful ownership and customers’ rights*” Lintian QIAO, Klara NAHRSTEDT.. Journal of Visual Communication and Image Representation, 9(3) :194 210, Septembre 1998.
- [VII] “*Tatouage d’image :Gain en robustesse et intégrité des images*” Christian REY. PhD thesis, Université d’Avignon et des Pays de Vaucluse, 2001.
- [VIII] “*Content protection system architecture, a comprehensive framework for content protection*” Intel Corporation, International Business Machines Corporation, Ltd. Matsushita Electric Industrial Co. et Toshiba Corporation. White paper, 4C entity, <http://www.4Centity.com>, février 2000. Revision 0.81.
- [IX] “*4c 12 bit watermark specification*” Intel Corporation, International Business Machines Corporation, Ltd. Matsushita Electric Industrial Co., and Toshiba Corporation. Specification, 4C entity, <http://www.4Centity.com>, octobre 1999.
- [X] “*SDMI. Portable device specification*”. Specification, SDMI, Los Angeles, juillet 1999.
- [XI] “*Content protection for recordable media specification: Introduction and common cryptographic elements*”. Intel Corporation, International Business Machines Corporation, Ltd. Matsushita Electric Industrial Co., and Toshiba Corporation. Specification, 4C entity, June 2000. Revision 0.93.
- [XII] “*Cryptanalysis of contents scrambling system*”, Franck A. STEVENSON.. <http://www.cs.cmu.edu/~dst/DeCSS/FrankStevenson/analysis.html>, November 1999.
- [XIII] “*Digital transmission content protection specification*”. Hitachi Ltd., Intel Corporation, International Business Machines Corporation, Ltd. Matsushita Electric Industrial Co., and Toshiba Corporation., 5C, http://www.dtcp.com/data/DTCP_spec11_informational.pdf, juillet 2000. Volume 1 (Informational Version).

-
- [XIV] “*5c digital transmission content protection white paper*”, Hitachi Ltd., Intel Corporation, International Business Machines Corporation, Ltd. Matsushita Electric Industrial Co., et Toshiba Corporation. White paper, 5C, http://www.dtcp.com/data/wp_spec.pdf, juillet 1998.
- [XV] “*High-bandwidth digital content protection Silicon Image*”. White paper, Silicon Image - Intel, février 2000.
- [XVI] “*Interoperable content protection for digital tv*”, B.J. VAN RIJNSOEVER and J.P. LINNARTZ. ICME 2000., 3, juillet août 2000. <http://www.iec.ch/opima>.
- [XVII] “*IEC. OPIMA specification. Specification*”, IEC, <http://www.iec.ch/opima/spec10.pdf>, 1999. Version 1.0.
- [XVIII] “*Request for Comments (RFC) Editor Homepage*”, <http://www.rfc-editor.org/>
- [XIX] 3GPP-Third Generation Partnership Project, <http://www.3gpp.org/>
- [XX] The MPEG Home Page, <http://www.cseit.it/mpeg/>
- [XXI] ISO/IEC JTC1/SC29/WG11 N3939, MPEG Requirements Group, MPEG-21 Proposed Draft Technical Report, Janvier 2001, Pisa MPEG meeting (http://www.cseit.it/mpeg/public/mpeg-21_pdtr.zip)
- [XXII] “*Improving the WWW: caching or multicast?*”, Pablo RODRIGUEZ, Keith W. ROSS, Ernst W. BIERSACK, Institut Eurécom. 3W3 Cache Workshop, Computer Networks and ISDN Systems 30 (1998) 2223-2243.
- [XXIII] “*Scalable Multicast Security in Dynamic Groups*”, Refik MOLVA, Alain PANNETRAT, Institut Eurécom., Singapore, novembre 1999. 6th ACM conference on Computer and Communications Security.
- [XXIV] “*IP Multicast Channels: EXPRESS Support for Large-scale Single-source Applications*”. Hugh W. HOLBROOK, David R. CHERITON, SIGCOMM 1999: 65-78~
- [XXV] “*Iolus: A Framework for Scalable Secure Multicasting*”, S. MITTRA, ACM Computer Communication Review, vol. 27, pp. 277-288, Oct. 1997. ACM SIGCOMM'97, Sept. 1997.
- [XXVI] “*Secure Group Communications Using Key Graphs*”, Chung KEI WONG, Mohamed GOUDA, and Simon S. LAM, Proceedings of ACM SIGCOMM, Vancouver, British Columbia, September 1998.
- [XXVII] “*IP Multicast Security: Issues and Directions*”, T. HARDJONO and G. TSUDIK, Annales de Telecom, to appear in 2000.
- [XXVIII] “*Multicast Security: A Taxonomy and Efficient Construction*”, CANETTI, J. GARAY, G. ITKIS, D. MICCIANCIO, M. NAOR, and B. PINKAS. In Proc. IEEE Infocom, March 1999.
- [XXIX] “*Random Oracles are Practical: A Paradigm for Designing Efficient Protocols*”, Mihir BELLARE, Phillip ROGAWAY. First ACM Conference on Computer and Communications Security

-
- [XXX] “*Scalable Multicast Security with Dynamic Recipient Groups*”. Refik MOLVA and Alain PANNETRAT, *ACM Transactions on Information and System Security*, 3, August 2000.
- [XXXI] “*Crypto-Gram*”, Bruce SCHNEIER, <http://www.counterpane.com/crypto-gram-0102.html#1>, February 2001.
- [XXXII] “*Mobile users : To update or not to update?*”, Amotz BAR-NOY, Ilan KESSLER and Moshe SIDI., *Wireless Networks* 1 (1995) 175-185.
- [XXXIII] “*Cours d’Échelles d’espace et de temps*”, Michel LAGÜES, Physique-Chimie Paris, troisième année, option physique.
- [XXXIV] “*Percolation and conduction in random resistor-diode networks*”, in *Percolation Structures and Processes*, vol. 5 of *Annals of the Israel Physical Society*. S. REDNER. Bristol, 1983.