ÉCOLE NORMALE SUPÉRIEURE DE CACHAN
Laboratoire Spécification et Vérification
CNRS UMR 8643

Numéro d'ordre : ENSC-2006n° 4

THÈSE

pour obtenir le grade de

DOCTEUR DE L'ÉCOLE NORMALE SUPÉRIEURE DE CACHAN
Discipline : Informatique

École Doctorale de Sciences Pratiques de l'ENS Cachan
E.D. numéro 285

présentée et soutenue publiquement
par

Pascal Lafourcade

le lundi 25 septembre 2006

Vérification de protocoles cryptographiques en présence de théories équationnelles.

devant le jury composé de :

Président	Claude Kirchner	Directeur de recherche au LORIA (Nancy)
Rapporteurs	Yassine Lakhnech	Professeur à l'Université Joseph Fourier (Grenoble)
	Luca Viganò	Chercheur à l'ETH (Zürich, Suisse)
Directeurs	Denis Lugiez	Professeur à l'Université Aix-Marseille I
	Ralf Treinen	Maître de Conférences à l'ENS Cachan
Examinateurs	Yannick Chevalier	Maître de conférence à l'Université Toulouse III

Résumé : L'essor d'Internet et des nouvelles technologies fait que l'informatique est au centre des moyens de communication actuels. Ces avancées technologiques impliquent un changement considérable de nos modes de consommation et de communication. Tous ces échanges de messages sont gérés par des protocoles de communications complexes que l'utilisateur ne contrôle pas totalement. Les usagers souhaitent que leurs communications soient "sécurisées". Les concepteurs de ces protocoles de communication sécurisent grâce à des méthodes cryptographiques les échanges de messages dans un environnement "hostile". Un tel environnement est constitué d'un participant malhonnête appelé intrus ou attaquant. Nous supposons que l'intrus contrôle le réseau sur lequel les messages sont échangés.

La vérification des protocoles cryptographiques assure qu'il n'existe pas d'attaque possible lors d'une exécution du protocole face à un certain intrus ou permet de trouver une attaque. Une des hypothèses importantes faites en vérification de protocoles cryptographiques est "l'hypothèse de chiffrement parfait" : le seul moyen d'obtenir le contenu d'un message chiffré est de connaître la clef de déchiffrement. Si un protocole est prouvé sûr sous cette hypothèse de chiffrement parfait, cela est insuffisant pour assurer qu'une information confidentielle échangée sur le réseau grâce à un protocole cryptographique entre deux participants reste secrète. Il se peut que certaines propriétés algébriques utilisées dans le protocole permettent à l'intrus d'obtenir de l'information.

Un des moyens pour affaiblir cette hypothèse de chiffrement parfait est de prendre en compte certaines propriétés algébriques dans le modèle de vérification afin d'analyser de manière plus réaliste les protocoles. Nous développons une approche formelle pour la vérification de la propriété de secret d'information pour les protocoles cryptographiques en présence de théories équationnelles et de l'homomorphisme.

Mots-Clefs : Vérification formelle, protocoles cryptographiques, propriétés algébriques, théories équationnelles.

Abstract : The rise of the internet of new technologies has reinforced the key role of computer science in communication technology. The recent progress in these technologies has brought a dramatic change in the ways how we communicate and consume. All these communication activities are subject to complex communication protocols that a user does not control completely. Users of communication protocols require that their communications are "secure". The developers of these communication protocols aim to secure communications in a hostile environment by cryptographic means. Such an environment consists of a dishonest communication participant, called an "intruder" or "attacker".. We suppose that the intruder controls the network on which the messages are exchanged.

The verification of a cryptographic protocol either ensures that no attack is possible against the execution of the protocol in presence of a certain intruder, or otherwise exhibits an attack. One important assumption in the verification of cryptographic protocols is the so-called "perfect cryptography assumption", which states that the only way to obtain knowledge about an encrypted message is to know its decryption key. This hypothesis is not sufficient to guarantee security in reality. It is possible that certain properties used in the protocol allow the intruder to obtain some information.

One way to weaken this perfect cryptography assumption is to take into account in the model certain algebraic properties. We develop a formal approach for verifying the so-called secrecy property of cryptographic protocols in the presence of equational theories and of homomorphism.

Keywords : Formal verification, cryptographic protocols, algebraic properties, equational theories.

Remerciements

« Dans la vie, les hommes sont tributaires les uns des autres.
Il y a donc toujours quelqu'un à maudire ou à remercier.»
Madeleine Ferron, Extrait de « Le chemin des dames ».

Je tiens tout d'abord à remercier tous les membres du jury, plus particulièrement Yassine Lakhnech et Luca Viganò les deux rapporteurs pour avoir accepté cette lourde tâche de relecture minutieuse de mon manuscrit, tâche d'autant plus difficile que cette thèse est rédigée en français. Je remercie aussi Yannick Chevalier d'avoir participé à mon jury et Claude Kirchner d'avoir bien voulu en être le président ; de même que mes deux directeurs de thèse : Denis Lugiez et Ralf Treinen pour leur soutien moral, scientifique et administratif qu'ils m'ont apporté durant ces trois années de thèse. Toujours disponibles, ils ont pris le temps de m'écouter, de répondre à mes nombreuses questions. Il m'ont aussi conseillé, orienté dans mon travail de recherche. Je les remercie également pour tous les moments agréables que nous avons partagés.

Tout cela n'aurait pas été sans l'aide de mes trois tutrices de DEA de l'UPS (Toulouse III) : Marie-Christine Lagasquié-Schiex, Hélène Fargier et Claudette Cayrol, qui m'ont donné goût à la recherche lors de mon stage à l'IRIT, qu'elles reçoivent toute ma considération.

Je remercie aussi Danièle Beauquier pour son encadrement exemplaire lors de mon année de monitorat à l'université Paris XII et de m'avoir permis de commencer à enseigner dans les meilleures conditions. Je suis également reconnaissant envers l'ensemble des membres de l'IUT de Fontainebleau avec lesquels j'ai pu collaborer activement durant deux ans.

Je tiens également à exprimer ma gratitude à l'ensemble des membres du Laboratoire Spécification et Vérification de l'E.N.S. de Cachan pour la bonne ambiance et la convivialité qui règnent au sein de ce laboratoire dynamique. Je remercie aussi l'ensemble des membres du Laboratoire d'Informatique Fondamentale de Marseille pour leur accueil chaleureux lors de mon année passée en leur compagnie. Mes voisins et voisines qui m'ont cotoyé dans les différents locaux que j'ai occupés ne sont pas oubliés (par ordre d'apparition Stéphanie, Vincent, Sébastien, Luccia, Nicolas, Nathalie et Mathieu), de même que les joueurs de Wanted, les footeux, les petits baigneurs et les squatcheurs du LSV qui tous ensemble contribuent à la bonne ambiance de cet agréable laboratoire.

Mes remerciements ne seraient pas complets si je ne mentionne pas les ex 2^{nd}6, particulièrement Aurélie, Marion, Stéphanie, Sébastien et Stéphane, les basketeur(se)s : Esther, Mélanie, Caro, Sandrine, Franck, Gros, Mam, Seb et les autres, les marseillaises Mao, Cam et Anne-françoise, les danseurs Elisabeth, Pierrot, Koko avec lesquels j'ai passé de bons moments, ainsi que les membres du groupe dit du samedi, du DU et plus particulièrement Nathalie pour son aide graphique. Je remercie aussi tous les membres de ma famille qui m'ont soutenu et accompagné durant de nombreuses années, ainsi que toutes les personnes qui m'ont apporté un soutien moral de loin ou de près.

« Whatever you do will be insignificant,
but it is very important that you do it. »
Mahatma Gandhi.

Table des matières

Liste des protocoles.

Table des figures

Table des notations

Nous utilisons les notations suivantes tout au long de cette thèse :

$\mathbb{N}$	entiers naturels
$\mathbb{Z}$	entiers relatifs
$\mathbb{Z}/2\mathbb{Z}$	le corps à deux éléments
$\mathbb{Z}/2\mathbb{Z}[h]$	anneau des polynômes en h sur $\mathbb{Z}/2\mathbb{Z}$
$\oplus$	opérateur du « ou exclusif »
$+$	opérateur additif de groupe abélien
$-$	opérateur inverse de groupe abélien
dom	dommaine d'une application
img	image d'une application
$\overline{u}$	vecteur u
$t\downarrow$	forme normal du terme t
$t\downarrow_{R/AC}$	forme normal modulo AC du terme t
$\{u\}_v$	chiffré du message u par la clef v
$\langle u, v\rangle$	paire du message u et v
$S(t)$	ensemble des sous-termes syntaxiques du terme t
$St(t)$	ensemble des sous-termes du terme t pour un chiffrement distributif
$S_c(t)$	ensemble des sous-termes du terme t pour un chiffrement distributif et commutatif
$NSt(t)$	ensemble des sous-termes du terme t
$Fact(t)$	ensemble des facteurs du terme t
mgu	unifieur le plus général
$T \vdash s$	le terme s est déductible à partir de l'ensemble de termes T
$\Vdash$	contrainte
$\Vdash_1$	contrainte une étape
$\Vdash_{\mathsf{M_E}}$	contrainte $\mathsf{M_E}$
.	produit entre polynômes de $\mathbb{Z}/2\mathbb{Z}[h]$
ACh	**A**ssociativité **C**ommutativité et **h**omomorphisme
ACUNh	**A**ssociativité **C**ommutativité **U**nité **N**ilpotence et **h**omomorphisme
AGh	**G**roupe **A**bélien et **h**omomorphisme
ACUN$\{.\}_.$	**A**ssociativité **C**ommutativité **U**nité **N**ilpotence et **C**hiffrement distributif
AG$\{.\}_.$	**G**roupe **A**bélien et **C**hiffrement distributif

Chapitre 1

Introduction.

« Computer Science is no more about computers than astronomy is about telescopes. »
E. W. Dijkstra.

Sommaire

Aujourd'hui dans la société moderne, l'informatique est omniprésente. Elle s'est introduite dans notre vie quotidienne et l'ordinateur, entraîné par l'essor d'internet, s'est démocratisé : de nombreux foyers possèdent de nos jours une boîte aux lettres électronique. L'apparition de l'informatique dans notre quotidien change nos modes de consommation et de communication. Il est désormais possible, à tous moments, de faire des achats dans une boutique virtuelle, de jouer ou de discuter via internet avec une personne située partout dans le monde. De plus en plus, les moyens de communication ne nécessitant pas un opérateur humain se développent : téléphone mobile, courrier électronique, forums de discussion, achats en ligne ... Tous ces échanges sont gérés par des protocoles de communications complexes que l'utilisateur ne contrôle pas totalement. Certains protocoles de communication envoient sur le réseau des informations sensibles, tel un numéro de carte bancaire, qui ne doivent pas tomber entre de mauvaises mains. Pour garantir un certain niveau de sécurité, les protocoles de communication chiffrent, grâce à des codes secrets, leurs échanges de messages. Malheureusement, l'usage de méthodes cryptographiques ne suffit pas pour garantir le secret d'une information confidentielle. Tous les jours, de nouvelles failles sont découvertes sur des protocoles cryptographiques. La découverte de telles attaques entraîne une perte économique considérable qui oblige alors à retirer momentanément le système, afin de le corriger ou le remplacer. Il est donc primordial de vérifier automatiquement la sécurité des protocoles cryptographiques avant leurs mises en service, afin de s'assurer qu'ils ne possèdent pas de failles. La sécurité de ces protocoles n'est pas uniquement garantie par l'usage d'une méthode de chiffrement cryptographique mais aussi par une vérification automatique et formelle du protocole lui-même.

Nous précisons maintenant la notion de protocole de communication, de participant honnête et d'intrus. Ensuite un bref aperçu historique des différentes techniques cryptographiques développées au fil des siècles nous amènera à présenter les principales techniques modernes de chiffrement. Enfin, nous montrons, sur un des plus célèbres protocoles cryptographiques, que malgré l'utilisation d'une méthode de chiffrement inviolable, il est possible d'obtenir une information secrète.

1.1 Protocoles de communication.

De part les siècles, de nombreuses techniques furent élaborées pour protéger les échanges d'informations. Les premières se basaient uniquement sur des méthodes issues de la cryptographie et développées par les militaires afin de protéger des informations secrètes. Ces méthodes utilisent une clef privée connue uniquement des deux participants. La transmission de cette clef qui s'effectuait, il y a encore quelques années, par un échange physique entre personnes de confiance, constitue une étape primordiale dans le protocole de communication. La perte de cette clef a des conséquences catastrophiques pour la confidentialité de la communication. Grâce aux avancées récentes des cryptographes, qui mirent au point des méthodes de chiffrement à clef publique, l'échange de cette clef privée entre deux participants est réalisé à l'aide d'un premier protocole de communication cryptographique utilisant un chiffrement asymétrique. Ainsi, aujourd'hui, les communications sécurisées s'effectuent entre deux individus « lambda » distants de plusieurs milliers de kilomètres, rapidement et à moindre coût, grâce aux avancées technologiques des moyens de communication et aux progrès des techniques de chiffrement.

1.1.1 Échange de messages.

La communication est l'échange d'information entre deux ou plusieurs personnes ; chaque personne étant appelée *participant* ou *agent*. Ces participants communiquent grâce à des envois de messages (concrètement, chaque message envoyé n'est qu'une suite de bits). Prenons un exemple : deux agents, Alice et Bob, souhaitent communiquer via un *canal de communication* qui peut être *public* ou *privé*. Les messages envoyés au moyen d'un canal de communication public sont visibles par tous les participants, alors que, sur un canal de communication privé, seuls les agents y ayant accès peuvent recevoir les messages émis sur ce canal. Ainsi, du temps de la guerre froide le « téléphone rouge » constituait un canal de communication chiffré entre Moscou et Washington. Si nous supposons ce canal de communication sûr, il devient alors un canal de communication privé entre ces deux puissances. D'autre part, les médias, tels la presse ou la radio, formaient un canal de communication public entre les deux blocs. Un protocole de communication permet l'échange de messages entre différents participants via un canal de communication. La forme, le contenu des messages, les différents participants à la communication et l'ordre dans lequel les messages sont échangés via un canal de communication spécifient un *protocole de communication*. Dès lors qu'un agent commence un protocole, nous dirons qu'une *session* du protocole débute, elle s'achèvera une fois le protocole terminé.

1.1.2 Les participants.

Chaque participant peut effectuer, simultanément, avec différents participants, plusieurs sessions du protocole. Les participants sont de deux types : honnêtes ou intrus.

1.1.2.1 Participants honnêtes.

Dans la description d'un protocole, les agents sont supposés être honnêtes, *i.e.* ils ont un comportement qui suit celui énoncé par une exécution normale du protocole. Les envois de messages sont précis et effectués dans l'ordre défini par le protocole. De plus, ces agents honnêtes ne collaborent pas avec les attaquants.

1.1.2.2 Intrus.

Un *intrus* ou *attaquant* est un participant qui ne suit pas exactement le déroulement du protocole. Il espionne les communications qui circulent sur les canaux publics, joue plusieurs sessions de protocoles avec des participants, en se faisant passer pour un agent honnête et ainsi effectue des actions non prévues par la spécification du protocole, afin de découvrir des informations supposées rester secrètes.

1.2 Un peu d'histoire.

Dès l'antiquité des moyens de transmission d'informations « sûrs » furent développés. Au Vème siècle avant J-C, Histaïaeus écrivit à Aristagoras en tatouant sur le crâne rasé d'un esclave son message. Il attendit la repousse des cheveux avant d'envoyer le messager avertir Aristagoras du danger qui se préparait. Ceci est un exemple d'utilisation de *stéganographie*. La stéganographie (du grec « steganos » couvert et « graphein » écrire) est l'art de cacher un message, de sorte que l'existence même du secret en soit dissimulée. Avec la stéganographie, un message est caché car il ne peut être détecté. De nombreuses techniques furent inventées comme l'encre invisible au temps de Pline (Ier siècle avant J-C) ou encore récemment le watermarking qui permet de dissimuler le copyright d'une image sans qu'il apparaisse sur l'image. D'autres techniques pour sécuriser les communications furent développées, elles sont pour la plupart basées sur une approche cryptographique (du grec « kruptos » cacher et de « graphein » écrire). Les techniques cryptographiques, contrairement à la stéganographie utilisée par Histaïaeus, modifient les messages originaux pour les rendre « incompréhensibles » afin d'assurer un certain niveau de sécurité face aux attaquants.

Une des premières techniques cryptographiques est le chiffrement par transposition. Pour chiffrer un message, l'ordre des lettres du message original est permuté. Pour le déchiffrer, il suffit d'appliquer la méthode inverse. Un des premiers exemples connus d'un tel chiffrement est la *scytale spartiate*, utilisée au Vème siècle avant J-C par les grecs. consiste en un bâton, autour duquel est enroulée une lanière de cuir. L'expéditeur écrit son message sur la lanière, puis une fois terminé la déroule et l'envoie. Le récepteur enroule à son tour la lanière reçue sur un bâton de même diamètre, ce qui lui permet ainsi de retrouver le texte original.

Une autre technique, appelée chiffrement par substitution, consiste à changer l'alphabet pour chiffrer un message. Elle était déjà utilisée du temps des romains sous le nom de « chiffrement de César ». Pour chiffrer un message, il faut décaler de trois lettres dans l'alphabet chaque lettre du message à transmettre. Pour décoder un message chiffré, il suffit de décaler chacune des lettres de 3 positions dans le sens inverse de l'alphabet. Nous présentons un exemple d'un message chiffré par cette méthode : « YHQL YLGL YLFL ». En appliquant la méthode exposée ci-dessus pour déchiffrer ce message nous retrouvons la célèbre phrase que prononça Jules César après sa victoire sur Pharnace roi du Borsphore à Zéla : « VENI VIDI VICI ».

Le chiffrement de Vigenère (XVIème siècle) est un autre chiffrement par substitution. Il s'agit d'une forme plus évoluée du chiffrement de César : 26 chiffrements de César sont appliqués dans un certain ordre. Cet ordre correspond à un mot ou une phrase connue de l'expéditeur et du récepteur du message. Cette information partagée constitue une *clef* qui permet d'effectuer dans le bon ordre les différents chiffrements de César. Ainsi, la même clef permet à la fois de chiffrer et déchiffrer un message. Notons que, dans une langue donnée, une étude de fréquence d'apparition des lettres de l'alphabet dans un texte fournit une aide précieuse pour « casser » les chiffrements par substitutions. Possédant un texte d'une longueur suffisante, il est alors possible de deviner les lettres les plus usitées, et ainsi de commencer à déchiffrer le message.

Pour terminer, citons un dernier exemple historique. Lors de la seconde guerre mondiale, les allemands mirent au point la machine « ENIGMA ». Celle-ci permettait de chiffrer un message grâce à un dispositif électro-mécanique qui, en fonction d'une clef donnée, réalisait une certaine combinaison de substitutions polyalphabétiques et de transpositions. Ainsi les allemands pensaient communiquer des informations en toute sécurité à leurs troupes. Mais les alliés, sous la direction d'Alan Turing, mirent au point « COLOSSUS 1 », un des premiers ordinateurs, ceci a permis de déchiffrer les messages générés, par ENIGMA. Pour plus de précisions sur l'histoire d'ENIGMA et des codes secrets en général, nous invitons le lecteur à consulter le célèbre livre de S. Singh [Sin99].

1.3 Un peu de cryptographie.

La cryptographie est un ensemble de techniques qui protègent un message en le transformant en un autre message : cette transformation modifie l'information contenue dans le message original pour rendre l'information transmise non compréhensible. Les cryptographes inventent des méthodes de chiffrement de plus en plus complexes, composées d'une fonction de chiffrement et d'une fonction de déchiffrement. La fonction de chiffrement permet de chiffrer un message donné m par une *clef* k, un paramètre de la fonction de chiffrement. Nous notons $\{m\}_k$ le message m chiffré par la clef k.

$$m \rightarrow \boxed{\text{fonction de chiffrement + clef}} \rightarrow \{m\}_k$$

La fonction de déchiffrement permet à partir d'un message chiffré $\{m\}_k$ de retrouver le message original m connaissant la clef de déchiffrement.

$$\{m\}_k \rightarrow \boxed{\text{fonction de déchiffrement + clef}} \rightarrow m$$

Ces fonctions cryptographiques sont en général basées sur des problèmes mathématiques dits « difficiles » à résoudre. Ainsi sans connaître la clef de déchiffrement il est difficile de déchiffrer un message, et ce, tant qu'il n'existera pas de moyen de résoudre le problème dit « difficile ». Nous présentons ici quelques unes des méthodes usuelles de chiffrement existantes. Pour cela nous distinguons deux catégories de chiffrement, les chiffrements symétriques et les chiffrements asymétriques. Pour plus de détails, le livre référence de B. Schneier [Sch96] et de nombreux autres ouvrages [H$^+$00, Kob00, Buc01, Mol01, DK02] proposent un aperçu assez complet des différentes méthodes de chiffrements existantes.

1.3.1 Chiffrements symétriques.

Le chiffrement symétrique utilise la même clef pour chiffrer et déchiffrer un message. La connaissance de cette clef est cruciale pour la confidentialité des informations échangées. Les algorithmes de chiffrement symétrique sont souvent basés sur des techniques de substitutions et de transpositions. Cela offre un moyen rapide et efficace pour chiffrer un message. Nous présentons trois méthodes de chiffrement : Le chiffrement de Vernam, le DES et l'AES.

1.3.1.1 Le chiffrement parfait (chiffrement de Vernam).

Il s'agit du « chiffrement à masque jetable » (One-time Pad) inventé par le Major J. Mauborgne et G. Vernam en 1917. Un masque jetable est une suite de bits aléatoires aussi longue que le message à chiffrer. Cette suite est un secret connu uniquement des deux participants et ne peut être utilisée qu'une seule fois. Le message original est codé sous forme de bits, pour le chiffrer, nous comparons chaque bit du masque et du message. S'ils sont égaux, nous écrivons 1 sinon 0 dans le message chiffré, ceci revient à effectuer une addition bit à bit modulo 2. Connaissant le masque, il est alors facile de reconstituer le message original.

Illustrons cette technique en additionnant modulo 26 les lettres de l'alphabet au lieu d'utiliser l'addition bit à bit modulo 2. Imaginons que nous souhaitons chiffrer une autre fameuse citation de Jules César, prononcée lorsqu'il franchit le Rubicon : « ALEA JACTA EST ». Nous utilisons comme masque jetable, une phrase connue à l'avance par les deux participants. Prenons la phrase de Jules César déchiffrée précédemment « VENI VIDI VICI ». La figure 1.1 montre comment cette phrase est chiffrée en « WQSJ FJGCW NVC ».

$$
\begin{array}{c|cccccccccccc}
 & A & L & E & A & J & A & C & T & A & E & S & T \\
 & 1 & 12 & 5 & 1 & 10 & 1 & 3 & 20 & 1 & 5 & 19 & 20 \\
+ & V & E & N & I & V & I & D & I & V & I & C & I \\
 & 22 & 5 & 14 & 9 & 22 & 9 & 4 & 9 & 22 & 9 & 3 & 9 \\
\hline
= & W & Q & S & J & F & J & G & C & W & N & V & C \\
 & 23 & 17 & 19 & 10 & 6 & 10 & 7 & 3 & 23 & 14 & 22 & 3
\end{array}
$$

FIG. 1.1 – Exemple de chiffrement à masque jetable.

Malheureusement, le chiffrement à masque jetable présente quelques inconvénients lors de sa mise en pratique car le masque doit être :

- aussi long que le message à chiffrer.
- utilisé une seule fois.
- généré de manière aléatoire pour éviter qu'il ne soit deviné.
- échangé de manière sûre entre les participants.

Sans connaître le masque il est prouvé sous ces conditions qu'il est impossible de retrouver le message original. Le chiffrement à masque jetable est un chiffrement parfait *i.e.* sans la clef le message est indéchiffrable.

Comme l'a fort bien souligné Steve Bellovin : « As a practical person, I've observed that one-time pads are theoretically unbreakable, but practically very weak. By contrast, conventional ciphers are theoretically breakable, but practically strong. » : le chiffrement à masque jetable est inviolable en théorie mais inutilisable en pratique. Il faut donc se tourner vers d'autres méthodes de chiffrement.

1.3.1.2 DES (Data Encryption Standard).

Le 23 Novembre 1976, le NIST (National Institute of Standards and Technology) adopte le standard de chiffrement DES [DH77, Nat80, Nat77, Nat88, SB88]. Ce chiffrement conçu par IBM sous le nom de LUCIFER a été choisi par la NIST après quelques petites modifications. Ce chiffrement symétrique permet de chiffrer des messages de 64 bits avec une clef k de 56 bits. Pour chiffrer un texte, il faut d'abord le découper en blocs de 64 bits puis appliquer le chiffrement sur chacun des blocs. Ce procédé est appelé *mode de chiffrement par blocs*. Nous donnons ci-dessous une idée simple et intuitive du fonctionnement du DES (pour plus de détails consulter le livre de B. Schneier « Applied Cryptography » [Sch96]). Ce chiffrement est constitué de 16 enchaînements successifs d'opérations de transposition, de substitution et de chiffrement de Vernam. Les avancées matérielles en informatique permettent aujourd'hui, en un temps raisonnable de « casser » un message chiffré avec DES par « force brute », *i.e.* en testant toutes les clefs possibles grâce à une énumération exhaustive. En 1998, la NIST lança un appel d'offres pour choisir, l' « Advanced Encryption Standard » (AES), le successeur du DES, devenu trop sensible aux attaques par recherches exhaustives.

1.3.1.3 AES (Advanced Encryption Standard).

Le standard de chiffrement AES fut adopté en 2000 par le NIST en remplacement du DES. Le NIST nomma AES cet algorithme symétrique de Rijndael, conçu par Vincent Rijmen et Joan Daemen [DR01, DR02]. Ce chiffrement est constitué de substitutions, de décalages, de « ou exclusif » et de multiplications dans un corps fini de polynômes fixés ; ces opérations sont élémentaires, simples et rapides à calculer. Il permet de crypter des blocs de 128, 192 ou 256 bits en utilisant des clefs symétriques de 128, 192 ou 256 bits. Le choix de la taille de la clef et de la taille des blocs sont indépendants, il y a donc au total 9 combinaisons possibles. Ceci laisse une plus grande flexibilité à l'utilisateur d'AES en fonction du niveau de sécurité et de la vitesse de calcul désirés.

1.3.2 Chiffrements asymétriques.

Un chiffrement asymétrique utilise une clef de chiffrement différente de celle de déchiffrement. La clef de chiffrement est souvent connue de tous les agents, elle est appelée *clef publique* et permet de construire un message chiffré. Cependant seuls les participants connaissant la clef de déchiffrement, appelée *clef privée*, peuvent déchiffrer les messages. Les chiffrements asymétriques sont la plupart du temps basés sur l'existence de fonctions mathématiques dites à *sens unique* ou *sens unique avec trappe*. Une fonction à sens unique est une fonction mathématique facilement calculable mais dont la réciproque est, en pratique, impossible à calculer car trop coûteuse. Par exemple, le produit de deux grands nombres premiers est une opération mathématique simple, mais trouver à partir de ce produit les deux nombres premiers est un problème connu difficile. Ce problème est appelé problème de factorisation. Les fonctions à sens unique avec trappe sont, elles aussi, facilement calculables mais leur réciproque est difficile à effectuer sauf pour les personnes qui connaissent une information secrète : la trappe. Ainsi, la factorisation en produit de deux nombres premiers est une fonction à sens unique avec trappe : si nous connaissons un des deux nombres premiers, il devient alors aisé par une simple division de retrouver le second nombre premier. Nous illustrons ci-dessous ces méthodes de chiffrement. D'abord nous décrivons le chiffrement RSA, certainement le plus connu et le plus utilisé des chiffrements asymétriques, ensuite nous présentons le chiffrement asymétrique probabiliste d'ElGamal.

1.3.2.1 Chiffrement RSA (1978).

Le chiffrement RSA, inventé par Rivest, Shamir et Adleman en 1978, est basé sur le problème difficile de la factorisation d'un entier en produit de deux grands nombres premiers. La clef publique est constituée de (n, e) et la clef privée est (d, n) tel que $n = pq$ où p et q sont deux nombres premiers, $e \in \{2, \ldots, (p-1)(q-1)\}$ et d est calculé tel que $ed = 1 \bmod (p-1)(q-1)$. Pour chiffrer par l'algorithme RSA un message m de taille inférieure à n avec la clef publique (n, e) nous calculons :

$$\{m\}_{(n,e)} = m^e \bmod n$$

Pour retrouver le message original m à partir d'un message chiffré $m' = \{m\}_{(n,e)}$ nous calculons :

$$m = (m')^d \bmod n$$

1.3.2.2 Chiffrement d'ElGamal (1985).

En 1985, ElGamal invente une technique de chiffrement asymétrique probabiliste *i.e.* un même message m peut avoir plusieurs chiffrés différents. Cette fois le problème difficile utilisé est le logarithme discret, provenant de la théorie des nombres, *i.e.* à partir de $g^a \bmod p$ il est difficile de retrouver a, où $g^a \bmod p$ dénote g à la puissance a modulo p. Pour utiliser le chiffrement d'ElGamal, il faut d'abord choisir un grand nombre premier p et un générateur g du groupe multiplicatif dans $\mathbb{Z}/p\mathbb{Z}$. La clef privée est un nombre entier aléatoire χ tel que $1 \leq \chi \leq p-2$. La clef publique est constituée de p, g et $h = g^\chi \bmod p$. Par ce chiffrement, un message m chiffré est représenté par un couple de nombres (a, b). Pour coder un message, choisissons aléatoirement un nombre r tel que $1 \leq r \leq p-2$ et calculons le couple de nombres (a, b) tel que :

$$\{m\}_{(p,g,h)} = (g^r \bmod p, mh^r \bmod p) = (a, b)$$

Nous retrouvons le message m à partir du chiffré $(a, b) = \{m\}_{(p,g,h)}$, connaissant χ, en calculant :

$$m = \frac{b}{a^\chi} \bmod p$$

.

1.3.2.3 D'autres chiffrements asymétriques.

Il existe de nombreuses méthodes de chiffrements asymétriques. Elles sont souvent rattachées à un problème mathématique difficile, comme le chiffrement de Rabin, inventé en 1979, qui est basé sur le problème difficile des racines carrées dans un corps fini. Citons une dernière méthode de chiffrement qui repose sur un problème difficile beaucoup plus complexe. Il s'agit de l'usage des courbes elliptiques en cryptographie proposé en 1985 par Miller [Mil85] (pour une approche générale des courbes elliptiques, c.f. par exemple [Kob85]).

1.3.3 Avantages et inconvénients des chiffrements symétriques et asymétriques.

Nous comparons ces deux types de chiffrements qui sont complémentaires et omniprésents dans les protocoles cryptographiques modernes.

Le chiffrement symétrique est plus rapide à calculer et utilise des clefs de tailles plus petites qu'un chiffrement asymétrique. De plus, il est basé des fonctions mathématiques simples et donc facilement implementables au niveau matériel. Par contre, pour chaque communication avec chaque participant, une clef différente est nécessaire. Il faut donc gérer la distribution d'un grand nombre de clefs, sachant que la divulgation de la clef serait catastrophique pour la sécurité de la communication.

Un chiffrement asymétrique permet de signer facilement un message, alors qu'un chiffrement symétrique ne peut le faire. La distribution des clefs publiques est très simple à gérer avec ce genre de chiffrement. De plus, cette méthode de chiffrement utilise des clefs de grandes tailles et nécessite un temps de calcul plus long et plus de ressources que lors d'un chiffrement symétrique, ceci à cause de la complexité des opérations à effectuer.

1.4 Exemple : Le protocole de Needham-Schroeder.

Pour attaquer un protocole cryptographique, il existe deux approches possibles : La première consiste à essayer de déchiffrer les messages chiffrés échangés. Ce type d'attaques développées par les cryptographes porte sur l'algorithme cryptographique employé. La seconde approche suppose que la méthode de chiffrement est inviolable, grâce à un raisonnement logique, cherche à obtenir de l'information. Ce genre d'attaque est appelée attaque logique. Nous présentons maintenant la plus célèbre de ces attaques logiques.

En 1978, R. Needham et M. Schroeder [NS78b] firent prendre conscience qu'un protocole utilisant un canal de communication public n'est pas nécessairement un échange de messages sûr, même si toutes les informations transmises sont cryptées par un chiffrement inviolable. De part la conception même du protocole, des informations confidentielles peuvent être découvertes par un intrus.

1.4.1 Le protocole.

Ce protocole décrit l'échange de messages entre deux participants, par exemple Alice et Bob, que nous dénoterons respectivement par A et B. Les deux participants veulent échanger en toute sécurité une clef privée N_b. Le protocole utilise un algorithme de chiffrement asymétrique. Nous dénotons par $\{m\}_{pub(A)}$ un message m chiffré par la clef publique d'Alice $pub(A)$ et considérons que tous les participants connaissent toutes les clefs publiques. Le protocole est décrit comme suit : :

1. Alice envoie à Bob un message chiffré par la clef publique de Bob, $pub(B)$. Il contient son identité A et un nombre aléatoire N_a « fraîchement » choisi par Alice, aussi appelé *nonce*.
2. Une fois ce message reçu, Bob choisit un nouveau nonce N_b et répond à Alice. Il envoie le nonce N_a précédemment reçu, ainsi que le nouveau nonce N_b dans un message chiffré par la clef publique d'Alice, $pub(A)$. N_b constitue la future clef échangée.
3. Alice confirme à Bob qu'elle a bien reçu le message, en lui renvoyant le nonce N_b chiffré par $pub(B)$, la clef publique de Bob.

$$\begin{array}{llcll} 1 & A & \rightarrow & B: & \{A, N_a\}_{\mathsf{pub}(B)} \\ 2 & B & \rightarrow & A: & \{N_a, N_b\}_{\mathsf{pub}(A)} \\ 3 & A & \rightarrow & B: & \{N_b\}_{\mathsf{pub}(B)} \end{array}$$

Protocole 1.4.1: Protocole de Needham-Schroeder.

Une fois le protocole terminé, Alice est convaincue d'avoir effectué une session du protocole avec Bob, car il lui a renvoyé son nonce. De même, Bob pense avoir communiqué avec Alice. Ils partagent alors une information commune N_b qui sera la clef publique utilisée dans leurs communications futures. Ils pensent également qu'ils sont les seuls à connaître N_b.

1.4.2 L'attaque.

Une *attaque* est l'exécution d'une ou plusieurs sessions du protocole qui permet à l'intrus d'apprendre une information supposée secrète. G. Lowe trouve, 17 ans après la publication du protocole, une attaque en utilisant un outil automatique de vérification de protocoles cryptographiques [Low96]. Cette attaque est plus connue sous le nom de « man in the middle ». Elle permet à un intrus Charlie, identifié par C, d'obtenir le nonce N_b échangé par Alice et Bob. Pour ce faire, Charlie joue deux sessions en parallèle du protocole comme indiqué ci-dessous :

Alice commence une session du protocole avec Charlie grâce à l'envoi 1.1. L'intrus déchiffre ce message et récupère l'identité d'Alice et son nonce N_a. Il peut ensuite commencer une autre session avec Bob et se faire passer pour Alice auprès de Bob en lui envoyant le message 2.1. D'après le message qu'il a reçu, Bob pense communiquer avec Alice. Il répond donc à Alice comme l'indique le protocole. Il participe par cet envoi à la seconde étape de la première session 1.2 entre Alice et Charlie et à la seconde étape de la seconde session 2.2 entre Charlie et Bob. À ce stade du protocole, Alice communique avec Charlie. Par suite, en 1.3, après avoir déchiffré le message reçu, Alice renvoie à Charlie les nonces N_a et N_b cryptés par la clef publique $pub(C)$, comme cela est spécifié dans le protocole. Charlie conclut alors la seconde session avec Bob par le message 2.3. Charlie connaît ainsi le nonce N_b et Bob pense parler en toute sécurité à Alice alors qu'il communique en fait avec Charlie.

$$
\begin{array}{rcclc}
1.1 & A & \rightarrow & C : & \{A, N_a\}_{\mathsf{pub}(C)} \\
2.1 & C & \rightarrow & B : & \{A, N_a\}_{\mathsf{pub}(B)} \\
2.2 \text{ et } 1.2 & B & \rightarrow & A : & \{N_a, N_b\}_{\mathsf{pub}(A)} \\
1.3 & A & \rightarrow & C : & \{N_b\}_{\mathsf{pub}(C)} \\
2.3 & C & \rightarrow & B : & \{N_b\}_{\mathsf{pub}(B)}
\end{array}
$$

Ce protocole sert à échanger une clef symétrique N_b entre Alice et Bob. L'intrus connaît maintenant cette clef, il peut alors continuer sans aucun problème à écouter les messages envoyés par Bob à Alice, les déchiffrer et même, se faire passer pour Alice. En effet, il peut répondre à Bob en chiffrant ses réponses grâce à la clef symétrique N_b.

1.5 Plan de la thèse.

Dans la première partie, nous détaillons les différentes composantes qui interviennent dans la vérification de protocoles cryptographiques. Nous précisons les deux types d'intrus que nous étudions par la suite : l'intrus passif et l'intrus actif. La notion de l'hypothèse de chiffrement parfait sera également introduite. Elle stipule que le seul moyen de déchiffrer un message est de connaître sa clef de déchiffrement. Puis, nous avons rassemblé les principaux résultats existants en vérification de protocoles cryptographiques : d'abord sous l'hypothèse du chiffrement parfait, ensuite les différents résultats qui tentent d'affaiblir cette hypothèse en illustrant dans chacun des cas la théorie par un exemple de protocole. Enfin, nous dégageons différents protocoles utilisant la propriété d'homomorphisme soit par leur conception même, soit par le choix de la méthode de chiffrement utilisée. La suite de cette thèse, fera l'objet de l'étude cette propriété d'homomorphisme, peu étudiée jusqu'à présent.

Nous proposons d'abord dans la deuxième partie de la thèse une extension du modèle de Dolev-Yao qui permet de prendre en compte l'affaiblissement de l'hypothèse du chiffrement parfait en intégrant à ce modèle une théorie équationnelle vérifiant certains critères. Le reste de la seconde partie de ce document concerne l'intrus passif. Nous établissons une approche qui traite de manière homogène les différentes théories équationnelles relatives à la propriété d'homomorphisme. Nous redémontrons d'abord le résultat pour le problème de déduction dans le modèle usuel de Dolev-Yao, puis, nous le prouvons dans le cadre d'un seul symbole homomorphique sur un opérateur muni des axiomes du « ou exclusif » et des groupes abéliens. Ensuite nous considérons le cas où ce symbole homomorphique est une fonction de chiffrement, et une fonction de chiffrement commutative pour les théories équationnelles du « ou exclusif » et des groupes abéliens et démontrons que le problème de déduction de l'intrus est décidable. Finalement nous montrons à travers des exemples que le problème d'unification et le problème de déduction de l'intrus modulo une théorie équationnelle sont deux problèmes indépendants.

Dans la dernière partie, nous nous intéressons à l'intrus actif pour la théorie équationnelle du « ou exclusif » avec un symbole homomorphique. Nous obtenons une procédure de décision dans ce cadre, en caractérisant d'abord les protocoles dits « déterministes » par un système de contraintes particulier. Ensuite, nous étudions en détail l'unification modulo ACUNh, puis nous résolvons des systèmes d'équations quadratiques diophantiennes particuliers. Finalement, tous ces éléments nous permettent de mettre en place un algorithme de décision qui résoud le problème de sécurité (intrus actif) en présence de la théorie équationnelle ACUNh pour les protocoles déterministes.

Première partie

Modèles pour la vérification de protocoles cryptographiques.

Chapitre 2

Vérification de protocoles cryptographiques.

« There are two kinds of cryptography in this world :
cryptography that will stop your kid sister from reading your files,
and cryptography that will stop major government from reading your files. »
Bruce Schneier, Applied Cryptography.

Sommaire

Maintenant, de plus en plus de protocoles cryptographiques sont vérifiés avant leurs commercialisations. L'analyse de ces protocoles est un problème de vérification en présence de canaux de communications non sécurisés. Étant donné un modèle de la spécification d'un protocole cryptographique, un modèle de l'intrus et une propriété à vérifier, nous voulons décider si le modèle satisfait bien la propriété considérée en présence de cet intrus. Nous énonçons donc les principales propriétés concernant les protocoles cryptographiques.

2.1 Propriétés à vérifier.

Étant donné un protocole, nous nous intéressons à savoir s'il vérifie une certaine propriété. Par exemple au paragraphe 1.4 page 8, nous souhaitons savoir si le nonce N_b était bien une donnée confidentielle entre Alice et Bob, ou encore être sûr que Bob communiquait bien avec Alice. Il existe de nombreuses propriétés de sécurité, nous présentons en quelques mots les plus usuelles. Dans le reste de la thèse, nous nous intéresserons uniquement à la propriété de secret, car cette propriété est la plus simple et la plus naturelle des propriétés qu'un protocole doit vérifier. De plus les autres propriétés décrites ci-après sont parfois définies à partir de la notion de données secrètes, ainsi notre travail peut s'appliquer à la vérification de ces propriétés.

2.1.1 Secret.

La plupart des auteurs utilisent la notion de secret suivante : un *s*ecret s est une donnée confidentielle ne devant pas être découverte par une tierce personne qui n'est pas censée la connaître. Un protocole vérifie la propriété de secret pour un secret s, si une personne malhonnête (l'intrus) en fonction de ces capacités ne peut jamais obtenir s échangé entre plusieurs participants honnêtes. Exhiber une attaque où un intrus obtient le secret s prouve que le protocole ne vérifie pas la propriété de secret pour la donnée s, cela correspond à savoir si le message s est accessible pour l'intrus. Il existe donc deux approches complémentaires pour vérifier la propriété de secret : soit nous cherchons une preuve que le secret est préservé, soit nous recherchons une attaque.

2.1.2 Autres propriétés.

Il existe une multitude de propriétés en fonction de chaque protocole à vérifier. Malheureusement pour chaque propriété, il y a de nombreuses définitions plus ou moins équivalentes suivant les auteurs. Dans le reste de cette section, nous indiquons de manière informelle l'idée usuelle et intuitive des propriétés les plus courantes.

2.1.2.1 Authentification.

La propriété d'authentification garantit à un agent qu'il communique bien avec le « bon » participant. Ceci est fort utile pour sécuriser les transactions bancaires sur internet. Quelles conséquences si une banque croit être en relation avec un client alors qu'il s'agit en fait d'une personne malhonnête. Il faut alors définir formellement comment garantir qu'un agent communique avec le « bon » participant. De nombreuses définitions différentes sont proposées dans la littérature, pour les consulter voir par exemple le livre de S. Schneider [Sch98, Low97b, Gol99, Gol00].

2.1.2.2 Anonymat.

Lors d'un vote à bulletin secret certaines règles doivent être respectées. Les règles électorales doivent garantir l'anonymat des votants : seul le votant connaît son vote. De nombreux protocoles de votes sont créés et l'anonymat est une des propriétés fondamentales qu'ils vérifient. Une propriété semblable d'anonymat a été développée dans d'autres situations. Par exemple pour la téléphonie mobile, l'intrus ne doit pas être capable de localiser un individu qui utilise son téléphone portable alors que la police elle aurait la possibilité de le faire. Le protocole doit donc garantir un certain niveau d'anonymat par rapport aux différents agents. Un autre exemple, lors de recherches sur internet, garantir l'anonymat des internautes empêche un site web de déterminer quelle est l'identité de ces visiteurs. Grâce au système TOR [DMS04], il est actuellement possible de naviguer sur internet en étant anonyme. De nombreuses définitions formelles de la notion d'anonymat existent dans chacun de ces contextes [SS96, HS04, DCJW04, KR05b, VS06], une liste de ces définitions est présentée par J.Y. Halpern et K.R. O'Neill [HO04]

2.1.2.3 Équité.

Cette propriété est utilisée par exemple dans la signature de contrat électronique. Si un client souhaite signer un contrat avec une banque et que celle-ci signe le contrat en premier, le client peut alors s'adresser à une banque concurrente et faire valoir cette signature pour négocier un meilleur contrat. Il y a donc un avantage certain à signer un contrat en second, le but de la propriété d'équité est de garantir qu'aucun participant n'est désavantagé lors de la signature du contrat. De nombreuses définitions ont été proposées [MGK03, KR02, CMSS03, KKW06] et sont utilisées dans de nombreux domaines.

2.2 Objectifs de la vérification.

La vérification de la sécurité de protocoles cryptographiques s'articule autour de deux axes complémentaires : rechercher une attaque ou prouver un protocole sûr.

2.2.1 Trouver une attaque.

Le premier axe de la vérification consiste à chercher une attaque sur un protocole. Observons que la majeure partie des attaques sur internet sont découvertes par des personnes « malicieuses » et que retrouver manuellement, sans la connaître, la célèbre attaque de « man in the middle » sur protocole de Needham-Schroeder de la section 1.4 page 8 n'est pas chose facile. La recherche d'attaques n'est donc pas une tâche aisée même pour un spécialiste, car comme l'a souligné G. Lowe, en développant son outil de vérification, l'esprit humain est faillible. Ainsi, de nombreux outils de vérification automatique de protocoles ont vu le jour et constituent une aide précieuse à la découverte d'attaques. Nous donnons une liste non exhaustive et nous décrivons quelques uns de ces outils dans la section 2.4.2.1 page 23 : AVISPA [ABB+02, ABB+05], CASPER/FDR [Low97a],COPROVE [CES06], Hermes [BLP03], Murphi [MMS97b], NRL [Mea96], Proverif [Bla04], Scyther [Cre].

2.2.2 La sûreté d'un protocole.

Un autre point important de la vérification est de prouver qu'il n'existe pas d'attaque sur un protocole donné en présence d'un intrus possédant certaines capacités, ce qui manuellement peut être fastidieux et complexe. Certains outils prouvent la propriété de sûreté d'un protocole pour un nombre borné de sessions, ce qui réduit considérablement l'espace de recherche. Il peut exister une attaque pour un nombre de sessions plus grand que celui considéré lors de la vérification, mais en faisant cette restriction, de nombreuses attaques furent découvertes grâce à des outils de vérification automatique basés sur des modèles formels. Malheureusement, vérifier formellement un protocole en général n'est pas possible. Plusieurs résultats d'indécidabilité pour les protocoles cryptographiques sont prouvés dans différentes modélisations [EG83, CC01, DLMS99, AC02]. Heureusement, en considérant certaines abstractions et restrictions sur la modélisation des protocoles cryptographiques, il est possible de vérifier formellement si un protocole est sûr contre les attaques menées par un intrus possédant certaines capacités. Si cette vérification formelle échoue, elle nous fournit dans le cas d'une abstraction correcte une preuve de la non-sûreté du protocole, cette preuve correspond effectivement à une véritable attaque si l'abstraction est complète. Une modélisation ou une abstraction formelle de protocole cryptographique est *correcte* *i.e.* , s'il existe une attaque dans le modèle concret ; alors il existe une attaque dans le modèle abstrait. Réciproquement, La modélisation est *complète* s'il existe une attaque dans le modèle abstrait alors il existe une attaque dans le modèle concret. Par exemple, les modélisations proposées par Y. Lakhnech *et alii* [BLP06], B. Blanchet [BP03], R. Ramanujam [RS03b], D. Monniaux [Mon99], J. Goubault-Larrecq [GL00] ou encore T. Genet *et alii* [GK00] sont des sur-approximations vérifiant la propriété de secret. Ces abstractions sont correctes mais pas complètes, certaines attaques trouvées ne sont pas de « vraies » attaques.

La vérification formelle de protocoles cryptographiques permet donc soit de montrer que la classe de protocoles modélisée est indécidable, soit de prouver la sûreté d'un protocole, soit de montrer l'insécurité du protocole en exhibant une attaque sur le protocole face à un certain intrus.

Depuis la faille découverte par G. Lowe sur le protocole de Needham-Schroeder, la vérification automatique de protocoles a connu un véritable essor. D'autant que M. Burrows *et alii* [BAN89] ont prouvé en 1989 que ce protocole était sûr dans un modèle différent. Il est donc important de bien comprendre les différentes hypothèses faites dans les différents modèles. Nous présentons maintenant les principales hypothèses utilisées dans la littérature ensuite nous proposons un récapitulatif non exhaustif des résultats existants. Nous présentons dans ce document les travaux prenant en comptes les plus pertinentes pour des théories équationnelles relaivement aux travaux réalisés. Dans notre article [CDL06], nous donnons une présentation plus complète en décrivant les propriétés d'associativité, des courbes elliptiques, et en proposant d'autres exemples de protocoles utilisant le « ou exclusif », le groupes abéliens et l'exponentielle modulaire. Remarquons que de nombreux protocoles sont décrits dans l'article de Clark et Jacob [CJ97], la base protocole SPORE [Jac] et la librairie de protocole du projet AVISPA [ABB$^+$05] disponibles en ligne.

Il existe plusieurs niveaux de modélisation possibles, nous en distinguerons deux : l'approche calculatoire (« computational ») et l'approche symbolique. La première considère les protocoles comme des envois de bits entre différents participants, et cherche à assurer qu'aucune information secrète n'est divulguée. Cette modélisation aborde le problème plus concrètement, en étant plus proche de l'implantation du protocole, ce modèle est souvent appelé *modèle concret*. Une deuxième approche plus formelle, que nous adoptons dans ce document, consiste à abstraire les échanges de bits entre participants par des termes et à raisonner avec ces termes pour vérifier le protocole.

Depuis quelques années, de nombreux travaux [AR00, BPW03, MW04, Lau04, BP04, AC04a, Lau05, BP05, AC05a, BCK05] voient le jour pour faire le lien entre l'approche calculatoire et l'approche symbolique. Ces travaux justifient l'approche symbolique en montrant que les attaques découvertes formellement restent vraies dans le modèle concret et que s'il n'y a pas d'attaques dans le modèle concret alors il n'y a pas d'attaques dans le modèle symbolique sous certaines hypothèses. Dans cette thèse, nous adoptons l'approche symbolique pour vérifier les protocoles cryptographiques.

2.3 Principales hypothèses de la modélisation.

Les protocoles cryptographiques sont constitués des échanges de messages entre des participants sur des canaux de communications. Chaque participant joue un rôle bien défini. Le protocole spécifie pour chaque agent quel message doit être renvoyé en fonction d'un motif de message reçu. Ces messages sont représentés par des termes construits à partir de symboles de fonctions utilisées dans le protocole tels la paire et le chiffrement de messages. D'autres symboles de fonction peuvent s'ajouter à cet ensemble de symboles usuels, selon la théorie équationnelle considérée. La modélisation par des termes dans l'approche symbolique ne permet pas d'effectuer des calculs sur les termes, si la théorie sous-jacente n'est pas explicite : Par exemple, une fonction permettant d'additionner deux termes est ajoutée au modèle pour que la théorie équationnelle de l'addition soit prise en compte dans la vérification du protocole. Pour capturer tout le pouvoir de ces nouveaux symboles de fonction nous considérons dans les parties II et III la vérification de protocoles cryptographiques en présence de théories équationnelles. Nous présentons maintenant les différentes principales hypothèses faites dans l'approche formelle.

2.3.1 Canaux de communications.

Il est possible de distinguer deux types de canaux de communications : les canaux de communications publics et les canaux de communications privés.

2.3.1.1 Publics.

Un canal de communication est dit *public* si tous les messages émis sur ce canal peuvent être vus par tous les participants, qu'ils soient honnêtes ou pas. Les messages échangés sur ces canaux sont donc connus de tous, en particulier de l'intrus qui peut s'en servir pour monter une attaque.

2.3.1.2 Privés.

Les canaux de communications dits *privés* sont des canaux de communications définis entre certains participants honnêtes, et seuls ces participants peuvent recevoir et envoyer des messages sur ces canaux. Par conséquent, un intrus ne peut donc pas écouter les messages qui circulent sur ce genre de canaux.

La plupart des modélisations considèrent uniquement des canaux publics ce qui confère un pouvoir plus grand à l'intrus et qui est une hypothèse de travail plus réaliste. De plus, assurer en pratique que les messages circulent sur un canal de communication privé n'est pas une tâche aisée, car il est difficile de garantir qu'un message ne soit pas écouté par un intrus. Les récentes découvertes en cryptographie quantique laissent entrevoir l'espoir, que dans les années à venir, il existera un moyen physique garantissant un canal de communication privé entre deux personnes.

2.3.2 Nombre de sessions.

Dans la modélisation des protocoles cryptographiques, le nombre de sessions, jouées par les participants, sont a priori infinis. Malheureusement considérer un nombre non borné de sessions est un des points importants de l'indécidabilité du problème de sécurité pour la vérification de protocoles cryptographiques, comme nous allons le voir par la suite. D'autre part, considérer une seule session n'est pas réaliste, car comme nous l'avons vu en introduction, l'attaque sur le protocole de Needham-Schroeder nécessite deux sessions en parallèle. Pour éviter l'indécidabilité due à un nombre infini de sessions et pour essayer de trouver certaines failles sur le protocole nous considérons, comme de nombreux autres travaux dans le domaine, un nombre borné de sessions, ce qui est formellement équivalent à considérer une seule session comme l'ont montré M. Rusinowitch et M. Turuani [RT01].

2.3.3 Deux modèles de l'intrus.

Pour l'étude de la propriété de secret, nous distinguons deux types d'intrus : l'intrus passif et l'intrus actif. Dans les deux cas, l'intrus possède des capacités lui assurant un certain pouvoir pour monter des attaques. Pour donner le maximum de pouvoir à l'intrus, il est souvent assimilé au réseau, ainsi il a connaissance de tous les messages qui sont émis sur les canaux de communications publics.

2.3.3.1 L'intrus passif.

L'intrus passif peut uniquement écouter les messages échangés sur le réseau, il n'a pas la possibilité d'émettre. En fonction de ses capacités déductives, l'intrus peut alors déduire de l'information grâce aux messages échangés.

Exemple 1 *Considérons une instance particulière du protocole de Shamir que nous détaillons dans la section 2.4.2.2 page 24, avec le chiffrement de Vernam. Le protocole entre Alice et Bob est le suivant :*

$$\begin{array}{lllll} 1 & A & \rightarrow & B: & s \oplus K_a \\ 2 & B & \rightarrow & A: & s \oplus K_a \oplus K_b \\ 3 & A & \rightarrow & B: & s \oplus K_b \end{array}$$

Il suffit alors à l'attaquant passif de faire le « ou exclusif » des trois messages échangés $s \oplus K_a$, $s \oplus K_a \oplus K_b$ et $s \oplus K_b$ pour retrouver immédiatement le message s. Si l'intrus ne possède pas cette capacité, il ne lui est pas possible d'apprendre cette information. Nous voyons bien l'importance que joue la modélisation des capacités de l'intrus dans la vérification des protocoles cryptographiques.

Soit T l'ensemble des messages que l'intrus a observé sur le réseau, le problème de déduction de l'intrus est de savoir si un intrus passif peut à partir de l'ensemble de messages T déduire un secret s.

2.3.3.2 L'intrus actif.

L'intrus actif peut comme l'intrus passif écouter et déduire de l'information à partir des messages échangés sur le réseau en fonction de ses capacités. Mais il peut aussi forger de nouveaux messages et les envoyer sur le réseau, par ce biais il peut se faire passer pour un agent honnête.

Exemple 2 *Nous rappelons le protocole 1.4.1 page 8 de Needham-Schroeder :*

$$\begin{array}{lllll} 1 & A & \rightarrow & B: & \{A, N_a\}_{\mathsf{pub}(B)} \\ 2 & B & \rightarrow & A: & \{N_a, N_b\}_{\mathsf{pub}(A)} \\ 3 & A & \rightarrow & B: & \{N_b\}_{\mathsf{pub}(B)} \end{array}$$

La célèbre attaque « man in the middle », décrite dans l'introduction de cette thèse, est une attaque qui ne peut pas être réalisée en présence d'un intrus passif. Il est nécessaire de considérer un intrus actif qui se fait passer pour un agent honnête et ainsi apprend N_b supposé être secret.

Pour un protocole donné, si un intrus actif ne peut pas déduire un secret s, nous dirons que le protocole est sûr en fonction des capacités de l'intrus prises en compte. S'il est décidable qu'un protocole préserve un secret s en présence d'un intrus actif possédant certaines capacités, nous dirons que le problème de sécurité est décidable pour cette modélisation de l'intrus. Comme nous le verrons plus tard il est possible qu'un protocole soit sûr contre un intrus avec certaines capacités de déduction et qu'en augmentant le pouvoir de l'intrus par une théorie équationnelle, le protocole possède alors une faille et ne soit plus sûr.

Notons que la vérification de protocoles cryptographiques dans le cas d'un intrus passif *i.e.* pour le problème de déduction de l'intrus, constitue une première étape nécessaire afin de vérifier le problème de sécurité des protocoles cryptographiques.

2.3.4 Hypothèse de chiffrement parfait.

Pour commencer à analyser et vérifier les protocoles cryptographiques, les chercheurs ont modélisé les fonctions mathématiques de chiffrement par des « boîtes noires » de telle sorte que le seul moyen de déchiffrer un message chiffré est de connaître la clef de déchiffrement. Cette hypothèse consistant à considérer les méthodes de chiffrement comme des « boîtes noires » est appelée hypothèse de chiffrement parfait. Cette hypothèse a permis de trouver de nombreuses failles dites « logiques » sur de nombreux protocoles.

2.3.5 Nonces.

Les protocoles cryptographiques utilisent des nombres aléatoires « frais » dans leur spécification. Cela signifie que le générateur de nombres aléatoires garantit une probabilité faible que deux nombres aléatoires soient identiques. Ces nombres aléatoires sont appelés *nonces*. Nous supposons dans la suite que tous les nonces sont des nombres distincts et imprévisibles, et que n'importe quel participant honnête ou pas puisse générer un nonce.

Le nombre de nonces est potentiellement illimité, ce qui rend la vérification des protocoles cryptographiques indécidable. Une restriction souvent considérée est de borner le nombre de nonces ; cela ne suffit pas forcément à rendre le problème décidable mais lève un des points qui est à l'origine de l'indécidabilité. Notons que le fait de considérer un nombre borné de sessions, nous permet de considérer qu'il n'y a qu'un nombre fini de nonces utilisés et donc de considérer les nonces comme des constantes deux à deux distinctes, ce qui lève deux points essentiels de l'indécidabilité de la vérification des protocoles cryptographiques. Une autre possibilité est de considérer les nonces comme des fonctions dépendantes des messages reçus précédemment dans la session, comme le font B. Blanchet [Bla01] ou C. Weidenbach [Wei99] en modélisant les protocoles cryptographiques grâce à des clauses logiques. Cette approche implique que deux nonces seront différents si et seulement si les messages échangés précédemment sont différents. Dans ce cas il se peut que les mêmes nonces soient utilisés dans deux sessions qui commencent par les mêmes premiers échanges de messages et donc conduisent à de fausses attaques. Plus précisément, en représentant les protocoles par des clauses de Horn, B. Blanchet autorise que chaque message soit utilisé plusieurs fois, ainsi il considère un nombre non borné de sessions. Le processus de résolution utilisé ne termine donc pas forcément en général, mais il montre que pour des protocoles « marqués » (« tagged ») la terminaison de sa procédure est assurée. À cause de la modélisation des nonces, cette procédure peut fournir de « fausses » attaques, attaques qui ne peuvent pas avoir lieu en pratique.

2.4 Résultats existants.

Nous présentons ici les différents résultats de décidabilité et d'indécidabilité existants pour le problème de sécurité et le problème de déduction de l'intrus. Dans un premier temps nous focalisons notre attention sur les modèles qui considèrent l'hypothèse du chiffrement parfait. Ensuite nous montrons les résultats existants qui tentent d'affaiblir cette hypothèse.

L'hypothèse de chiffrement parfait a permis d'obtenir de nombreux résultats sur les protocoles cryptographiques. Grâce à cette modélisation des méthodes de chiffrement, les chercheurs ont pu s'affranchir de la complexité mathématique sous-jacente aux fonctions cryptographiques et ainsi vérifier les protocoles cryptographiques. Nous présentons les principaux travaux de ces dernières années pour la vérification de protocoles sous l'hypothèse de chiffrement parfait. Ces résultats sont récapitulés dans la Figure 2.1 page ci-contre. Ensuite, de nombreux travaux furent développés dans l'optique d'affaiblir cette hypothèse du chiffrement parfait pour être plus proche de primitives cryptographiques, et ne plus les considérer comme des boîtes noires. La majeure partie de ces travaux est basée sur l'ajout de propriétés algébriques dans la modélisation des protocoles cryptographiques. Cela permet de capturer les spécificités des fonctions cryptographiques utilisées et également certaines propriétés algébriques que le protocole utilise de part sa conception même. Nous présentons quelques-uns des principaux travaux existants qui affaiblissent l'hypothèse de chiffrement parfait. Une vue synthétique de ces résultats est donnée dans la figure 2.2 page 25, pour une vue plus complète cf [CDL06].

Tous ces résultats furent obtenus dans différents modèles comme le spi-calcul [AG99], la réécriture de multi-ensembles, les clauses de Horn [Bla01, NNS02, Gou05], ou encore les « strand spaces » [FHG98]. Nous énonçons les résultats sans mentionner le modèle utilisé, car il est d'usage d'accepter que ces résultats restent valident dans les autres modèles. Il existe d'ailleurs certains travaux qui font le lien entre les différents modèles proposés, comme le papier de S. Bistarelli *et alii* [BCLM03] qui permet de faire le lien entre les algèbres de processus et la réécriture de multi-ensembles ou comme le papier de Cervesato *et alii* [CDM$^+$00] qui relie les strands et la réécriture mutli-ensembles.

2.4.1 Avec hypothèse du chiffrement parfait.

Dans cette section, nous présentons les principaux résultats de vérification de protocoles cryptographiques établis sous l'hypothèse du chiffrement parfait. Nous rappelons que sous cette hypothèse le seul moyen de retrouver le message chiffré par un algorithme de chiffrement est de connaître la clef de déchiffrement. Les primitives cryptographiques sont alors vues comme des boîtes noires. Les propriétés algébriques des fonctions de chiffrement ne sont pas alors prises en compte dans la vérification des protocoles. Le tableau 2.1 page ci-contre résume les principaux résultats de décidabilité et d'indécidabilité existants pour la vérification de protocoles cryptographiques sous cette hypothèse.

2.4.1.1 Résultats d'indécidabilité.

Les protocoles cryptographiques sont souvent décrits sous forme de petits programmes simples. Souvent les protocoles cryptographiques sont constitués d'une séquence de messages envoyés et reçus, ces opérations sont exécutées en fonction du déroulement du protocole. Cette description concise ne rend pas pour autant la vérification aisée. Vérifier un protocole cryptographique en toute généralité est un problème indécidable. L'indécidabilité provient de plusieurs aspects de la modélisation des protocoles, comme le nombre non borné de sessions, la longueur des messages, la

<table>
<tr><th rowspan="2">Nombre borné de sessions</th><th colspan="2">Nombre non borné de sessions</th></tr>
<tr><th>Sans nonce</th><th>Avec nonces</th></tr>
<tr><td rowspan="4">co-NP-complète
[RT01]</td><td>Cas général :
Indécidable
[EG83, CC01]</td><td>Cas général :
Indécidable
[EG83, CC01, DLMS99, AC02]</td></tr>
<tr><td>Longueur de message borné :
DEXP-TIME-complet
[DLMS99, CKR$^+$03b]</td><td>Longueur de message borné :
Indécidable [DLMS99, AC02]
avec des sous-termes non unifiables :
Décidable [RS03a]</td></tr>
<tr><td>Protocoles marqués :
EXP-TIME
[BP03]</td><td>Longueur de message borné,
Protocoles fortement typés
Identités inférables :
Décidable [Low98]
—
Protocoles marqués :
Décidable [RS03b]</td></tr>
<tr><td>Une copie :
DEXP-TIME-complet
[CLC03, SV05]</td><td>Protocoles ping-pong :
P-TIME [DY83]</td></tr>
</table>

FIG. 2.1 – Principaux résultats existants sous l'hypothèse de chiffrement parfait.

génération de nonces. Nous décrivons succinctement quelques-uns des principaux résultats d'indécidabilité existants pour le problème de sécurité sous l'hypothèse de chiffrement parfait.

Si nous considérons un nombre borné de sessions, l'indécidabilité provient alors de la génération des nonces. Plusieurs codages du problème de correspondance de Post [DW83] sont proposés dans ce cas. S. Even et O. Goldreich montrent en utilisant ce problème indécidable que le problème de sécurité pour un nombre borné de nonces reste encore indécidable [EG83]. De même H. Comon-Lundh et V. Cortier [CC01] prouvent que le problème est encore indécidable même sans utiliser de clefs composées. Dans ces deux approches, l'indécidabilité provient du fait que la taille des messages n'est pas bornée.

En 1999, N. Durgin *et al.* [DLMS99] montrent l'indécidabilité du problème de sécurité même en bornant la taille de messages. Ils prouvent leur résultat en codant la génération de noncesen clauses existentielles de Horn grâce aux primitives de chiffrement et de construction de paire de messages. Par une analyse plus fine, R. Amadio et W. Charatonik [AC02] montrent, en utilisant uniquement la primitive de chiffrement, que le problème de sécurité est indécidable en codant une machine de Turing à deux compteurs.

2.4.1.2 Résultats de décidabilité.

Après avoir identifié les principales raisons de l'indécidabilité du problème de sécurité pour les protocoles cryptographiques sous l'hypothèse du chiffrement parfait, comme la longueur des messages, le nombre de sessions ou encore le nombre non borné de nonces, nous présentons les résultats de décidabilité sous cette hypothèse du chiffrement parfait. Ces résultats nécessitent de fortes restrictions sur les protocoles pour éviter l'indécidabilité sous-jacente à ce problème.

Même si le nombre de sessions est supposé borné, et si le nombre de nonces utilisés est lui aussi borné, il n'est pas facile de montrer que le problème de sécurité est décidable, car l'intrus peut engendrer un nombre infini de messages. M. Rusinowitch et M. Turuani [RT01] ont étendu l'étude menée par R. Amadio *et al.* [ALV02] et proposent une procédure de décision co-NP-complète pour le problème de sécurité avec un nombre borné de sessions sous l'hypothèse de chiffrement parfait, dans le modèle d'attaquant de Dolev-Yao. Nous expliquons en détail ce modèle dans la section 3.1 page 46.

Un des premiers résultats dans ce domaine est le résultat de D. Dolev *et al.* pour les protocoles ping-pong entre deux participants [DY83], où chaque participant ne fait qu'appliquer une suite d'opérateurs sur le message qu'il a reçu et renvoie ensuite le message ainsi généré. L'étude de ces protocoles ping-pong permet d'obtenir une procédure en temps polynomial (« P-TIME ») pour résoudre le problème de sécurité de ces protocoles et ce, même pour un nombre non-borné de sessions.

En 1989, G. Lowe [Low98] a obtenu un résultat de décidabilité pour le problème de sécurité avec un nombre non borné de nonces pour une sous-classe particulière de protocoles. Dans sa sous-classe de protocoles, il a supposé la taille des messages secrets bornée, et que chaque participant puisse analyser les messages reçus et que les identités soient inférables à partir de messages.

R. Ramanujam et S. Suresh étudient le cas où le nombre de nonces n'est pas borné, en considérant bien sûr deux restrictions. Premièrement, ils prouvent que le problème de sécurité est décidable en supposant que la taille des messages soit bornée et qu'aucun sous-termes d'un message ne puisse être unifié avec un sous-terme d'un autre message [RS03a]. Dans un deuxième temps [RS03b], ils forcent la condition de non unification des sous-termes chiffrés en imposant que tous sous-termes chiffrés soient marqués (« tagged ») par un nombre frais. Sous cette nouvelle condition et pour un nombre non borné de nonces et une taille de message non bornée, ils démontrent que le problème de sécurité est décidable. Malheureusement tous les protocoles ne vérifient pas cette restriction, par exemple le protocole de Yahalom [BAN89, Pau01, CJ97] où certains messages chiffrés sont renvoyés à un autre participant sans pouvoir être déchiffrés.

Il existe de nombreux travaux [CLC03, CKR$^+$03b, DLMS99] dans lesquels les auteurs considèrent le problème de sécurité avec un nombre borné de nonces et d'autres restrictions. Citons H. Comon-Lundh et V. Cortier [CLC03] qui considèrent une classe de protocoles, dans laquelle à chaque envoi de message un agent ne peut copier qu'une seule composante du message reçu. Ils montrent que le problème de sécurité est alors décidable en 3-EXP-TIME. H. Seidl et K. Verma [SV05] étudient en détail la complexité de cette classe de protocoles et montrent que le problème de sécurité est DEXP-TIME-complet. Les travaux de Y. Chevalier *et alii* [CKR$^+$03b] et N. Durgin *et alii* [DLMS99] supposent que la taille des messages est bornée et obtiennent également que le problème de sécurité est DEXP-TIME-complet. Dans son approche, B. Blanchet [BP03] considère un nombre borné de nonces et utilise un procédé de marquage des protocoles (« tagging ») pour obtenir la décidabilité dans le cas d'un nombre non borné de sessions pour le problème de sécurité. D'autres résultats similaires de décidabilité [Bor01, MS01] existent dans d'autres modèles que ceux considérés précédemment. Citons le résultat obtenu par H. Hüttel [Hüt02] dans lequel il montre que la propriété de *secret fort* dans le contexte de l'algèbre de processus est décidable. Un secret s vérifie la propriété de secret fort si une session contenant le secret s est indistinguable de toutes sessions qui contiendraient le secret s' à la place de s

2.4.2 Vers un affaiblissement de l'hypothèse du chiffrement parfait.

Les primitives de chiffrement employées dans les protocoles cryptographiques sont, la plupart du temps, basées sur des fonctions mathématiques à sens unique. Considérer ces primitives comme des « boîtes noires » a permis de vérifier certains protocoles cryptographiques et découvrir des failles de sécurité importantes. Comme nous le détaillerons par des exemples de protocoles, il existe de nombreuses attaques qui utilisent des propriétés algébriques du protocole lui-même ou de la méthode de chiffrement utilisée. Considérer les méthodes de chiffrement et les protocoles sans prendre en compte ces propriétés algébriques ne permet pas de trouver toutes les attaques.

Nous présentons maintenant les différents travaux qui prennent en compte l'affaiblissement du chiffrement parfait en considérant la plupart du temps un nombre borné de sessions. L'ajout de théories équationnelles permet d'affaiblir l'hypothèse du chiffrement parfait et d'analyser de manière plus réaliste les protocoles cryptographiques. Nous présentons d'abord certains outils capables de prendre en compte certaines propriétés algébriques. Ensuite nous décrivons les principaux résultats de décidabilité existants qui tentent d'affaiblir cette hypothèse pour le problème de déduction en considérant un intrus passif et le problème de sécurité en présence d'un intrus actif. Cette présentation n'est pas exhaustive, nous avons réalisé une étude plus complète [CDL06]. Dans la suite, nous porterons notre attention uniquement sur certaines théories équationnelles. Le tableau 2.2 page 25 récapitule l'ensemble des résultats décrits ci-après.

2.4.2.1 Outils.

Il existe peu d'outils automatiques qui vérifient les protocoles cryptographiques en prenant en compte les propriétés algébriques, nous en présentons brièvement certains.

L'outil ProVerif développé par B. Blanchet [Bla04] permet à l'utilisateur de spécifier une théorie équationnelle grâce à des clauses de Horn. Les équations ainsi modélisées par des clauses de Horn sont analysées par l'outil. Signalons que pour des théories équationnelles trop complexes l'outil ne termine pas.

L'outil Hermes [BLP03] permet de vérifier si une donnée est secrète dans un protocole. Il génère l'ensemble des données qu'un intrus ne doit pas apprendre et l'ensemble de données non secrètes. Un des objectifs de la nouvelle version de cet outil serait de prendre en compte la notion de temps pour la vérification des protocoles cryptographiques.

La plate forme AVISPA [ABB+05] est capable d'analyser les protocoles en prenant certaines propriétés algébriques. Cette plate-forme est composée des outils suivants : On-the-Fly-Model-Checker (OFMC) [BMV03, BMV05b], The constraint-Logic-based AttackSearcher (CL-AtSe) [Tur06], SAT-based Model-Checker (SATMC) [AC05b, AC04b] et Tree Automata based on Automatic Approximation for the Analysis of Security Protocols (TA4SP) [BKV06, BHK05]. Récemment, la propriété du « ou exclusif » a été ajoutée à l'outil CL-ATSE. Cet outil essaye dans sa dernière version [Tur06] d'adopter une approche modulaire afin de prendre en compte différentes théories équationnelles.

Une récente étude de M. Hussain et D. Seret [HS06] compare l'outil AVISPA et l'outil Hermes. Ces deux outils considèrent deux langages de spécification différents en entrée et ne traitent pas des mêmes propriétés algébriques. Bien que ces deux outils ne soient pas comparables à mon avis, selon M. Hussain et D. Seretles Hermes semble plus facile à utiliser qu'AVISPA et réciproquement AVISPA semble considérer plus de protocoles de par l'expressivité plus grande de son langage de spécification.

Un autre outil est l'analyseur de protocole NRL [Mea96] de C. Meadows. Cet outil est basé sur les techniques de sur-réduction (« narrowing ») et permet de prendre en compte certaines théories équationnelles. Dans la première version de l'analyseur, certaines règles classiques de déduction étaient prises en compte pour modéliser une propriété du chiffrement symétrique. Cette propriété consiste à remarquer que chiffrer deux fois un message m nous redonne m. Par la suite, l'outil a été amélioré [Mea00], afin d'analyser les protocoles utilisant l'exponentielle à la Diffie-Hellman, comme le protocole IKA.1 présenté dans la section 2.4.2.5 page 32.

L'outil Casper [Low97a], développé par G. Lowe et A.W. Roscoe, modélise partiellement le chiffrement de Vernam (qui utilise le « ou exclusif »). Grâce à cet outil G. Lowe [LR97] trouva certaines attaques sur le protocole TMN [TMN89], protocole décrit en détail dans la section 2.5.2.4 page 39. De plus cet outil permet d'analyser les protocoles avec timestamps sous certaines hypothèses. Nous présentons dans la section 2.4.2.6 page 34 un protocole utilisant ces notions.

Nous présentons maintenant certaines propriétés algébriques utilisées dans les protocoles cryptographiques. Nous donnons pour chaque propriété un exemple de protocole utilisant la propriété explicitement dans sa spécification ou possédant une attaque basée sur cette propriété. A chaque protocole décrit, nous énonçons les principaux résultats théoriques qui sont associés.

2.4.2.2 Chiffrement commutatif.

Un chiffrement est commutatif s'il vérifie la propriété suivante :

$$\{\{m\}_{k_1}\}_{k_2} = \{\{m\}_{k_2}\}_{k_1}$$

où m est un message, k_1 et k_2 sont deux clefs distinctes. Nous donnons d'abord deux exemples de protocoles utilisant cette propriété et ensuite nous rappelons les résultats théoriques la prenant en compte.

Le protocole de Shamir (Shamir's Three-Pass Protocol).

Ce protocole inventé en 1996 par A. Shamir est décrit dans l'article de J. Clark et J. Jacob [CJ97] recensant de nombreux protocoles et attaques. Ce protocole est également présenté dans le livre « Applied Cryptography » de B. Schneier [Sch96].

	Problème de déduction de l'intrus	**Problème de sécurité**	
		Nombre de sessions borné	**Nombre de sessions non borné**
Commutativité du chiffrement	*P-TIME [CKRT04]*	*co-NP-complet [CKRT04]*	**Ping-pong protocoles** *co-NP-complet [Tur03]*
Préfixe	*P-TIME [CKRT03]*	*co-NP-complet [CKRT03]*	*Décidable [CRZ05]*
« ou exclusif »	*PTIME [CKRT03]*	**Cas général** *Décidable [CLS03]* **Protocoles restreints** *co-NP-complet [CKRT03]*	**Une copie, sans nonce** *Décidable [CLC03]* **Two-way automata, sans nonce** *Décidable [Ver03]*
Groupe abélien	*P-TIME [Che03]*	*Décidable [Shm04, MS05]*	**Two-way automata, sans nonce** *Décidable [Ver03]*
Groupe abélien et exponentielle modulaire	*P-TIME [CKR+ 03a]*	**Cas général** *Décidable [Shm04]* **Protocoles restreints** *co-NP-complet [CKR+ 03a]*	**Propriété AC de l'exponentielle modulaire sans nonce** *Semi-Décision Procédure [GLRV04]*
Timestamps	*P-TIME* (1)	*Décidable [BEL04]*	*Semi-Décision Procédure [DG04]*

(1) Pour le problème de déduction de l'intrus, la notion de temps n'a pas de sens. Ce résultat se déduit donc du résultat sans théorie équationnelle en considérant les valeurs temporelles comme des constantes connues de l'intrus.

FIG. 2.2 – Récapitulatif des résultats de décidabilité en présence de théories équationnelles.

Description du protocole : Le « Shamir's Three-Pass Protocol » nécessite l'emploi d'un chiffrement commutatif. Il est spécifié de la manière suivante pour deux agents : Alice dénoté par A et Bob dénoté par B.

$$\begin{array}{llcll} 1 & A & \rightarrow & B: & \{m\}_{K_a} \\ 2 & B & \rightarrow & A: & \{\{m\}_{K_a}\}_{K_b} \\ 3 & A & \rightarrow & B: & \{m\}_{K_b} \end{array}$$

Protocole 2.4.1: Protocole de Shamir (« Shamir's Three-Pass Protocol»).

Alice envoie à Bob un message chiffré par sa clef K_a. Bob chiffre à son tour le message reçu avec sa clef K_b et le retourne à Alice. Comme le chiffrement est commutatif, Alice peut déchiffrer le message et envoyer à Bob $\{m\}_{K_b}$, message que Bob est alors capable de déchiffrer.

Remarquons, comme nous l'avons signalé dans l'exemple 1 page 18, que ce protocole ne doit pas être utilisé avec le chiffrement de Vernam qui possède bien la propriété suivante : $\{\{m\}_{K_a}\}_{K_b} = m \oplus K_a \oplus K_b = \{\{m\}_{K_b}\}_{K_a}$. Il suffit alors à l'attaquant passif de faire le « ou exclusif » des trois messages échangés $m \oplus K_a$, $m \oplus K_a \oplus K_b$ et $m \oplus K_b$ pour retrouver le message m, sans connaître pour autant les deux clefs utilisées par Alice et Bob.

Attaque : Les participants ne vérifient pas l'authenticité de l'identité de leurs correspondants. La fameuse attaque « man in the middle », décrite sur le protocole de Needham-Schroeder au paragraphe 1.4 page 8, peut être jouée ici encore plus simplement. Supposons qu'un chiffrement asymétrique et commutatif soit utilisé.

$$\begin{array}{rrcll} i.1 & A & \rightarrow & I(B): & \{m\}_{K_a} \\ ii.1 & I(A) & \rightarrow & B: & \{m\}_{K_a} \\ ii.2 & B & \rightarrow & I(B): & \{\{m\}_{K_a}\}_{K_b} \\ i.2 & I(B) & \rightarrow & A: & \{\{m\}_{K_a}\}_{K_I} \\ i.3 & A & \rightarrow & I(B): & \{m\}_{K_I} \\ ii.3 & A & \rightarrow & I(B): & \{m\}_{K_b} \end{array}$$

Ainsi Alice pense parler avec Bob et lui transmettre le message confidentiel m en toute sécurité. Malheureusement elle communique avec l'intrus qui prend donc connaissance du message confidentiel m.

Protocole d'échange de clef de Diffie-Hellman.

En 1978, W. Diffie et M. Hellman proposent un protocole d'échange de clef [DH76]. Ce protocole basé sur la commutativité de la fonction exponentielle porte le nom de ces auteurs : le protocole d'échange de clef de Diffie-Hellman.

Description du protocole : Alice choisit un nombre premier P et un générateur G du groupe $\mathbb{Z}/P\mathbb{Z}$. Alice envoie à Bob ces données. Bob choisit un nonce Nb et envoie G à la puissance Nb modulo P à Alice, message que nous notons par $exp(G, Nb)$ mod P. Finalement Alice choisit à son tour un nonce Na et envoie G à la puissance Na modulo P à Bob. Les deux participants peuvent donc calculer la nouvelle clef, qu'ils partagent, en élevant à la puissance de leurs nonces respectifs le message reçu modulo P. Ceci est dû à la propriété suivante de commutativité de l'exponentielle :

$exp(exp(G, x), y) \bmod P = exp(exp(G, y), x) \bmod P$

En considérant le chiffrement comme une boîte noire B. Blanchet prouva que la nouvelle clef échangée $exp(exp(G, Nb), Na)$ mod P est bien une donnée secrète [Bla01].

$$\begin{array}{llcll} 1 & A & \rightarrow & B: & P, G \\ 2 & B & \rightarrow & A: & exp(G, Nb) \bmod P \\ 3 & A & \rightarrow & B: & exp(G, Na) \bmod P \end{array}$$

Protocole 2.4.2: Protocole d'échange de clef de Diffie-Hellman.

Attaque : Le protocole n'authentifie pas les participants. Il est donc possible à un intrus de prendre l'identité d'un des participants, cette attaque connue est présentée dans la base de protocoles SPORE [Jac].

Résultats théoriques existants.

Y. Chevalier *et alii* [CKRT04] présentent une procédure de décision NP pour le problème de sécurité des protocoles en présence d'un chiffrement commutatif *i.e.* $\{\{x\}_y\}_z = \{\{x\}_z\}_y$. Par exemple le chiffrement RSA, un des chiffrements les plus connus de nos jours, vérifie cette propriété à condition d'utiliser le même modulo.

2.4.2.3 Propriété de préfixe.

Cette propriété concerne le mode de chiffrement. Un algorithme de chiffrement vérifie la propriété préfixe si à partir d'un message chiffré par la clef k de la forme suivante : $\{x, y\}_k$, il est possible de déduire le message $\{x\}_k$. Par exemple, la méthode ECB (Electronic CodeBook) pour chiffrer un message avec un algorithme de chiffrement possède cette propriété, nous rappelons le fonctionnement de cette méthode ECB pour un algorithme de chiffrement donné, chiffrant un message de longueur n. Elle consiste pour chiffrer un message de longueur plus grand que n à découper le message à chiffrer en morceaux de taille n et de les chiffrer un par un par l'algorithme de chiffrement pour obtenir le chiffré du message m. Notons que cette technique ECB est rarement utilisée en pratique.

Une méthode plus sophistiquée est la méthode CBC (cipher-block chaining). Dans cette méthode, le texte m à chiffrer est découpé en n blocs de même longueur $m = P_1P_2\cdots P_n$ (où des bits aléatoires peuvent être ajoutés pour compléter la dernière composante de m pour qu'ils aient tous la même longueur). Le message m chiffré avec la clef K est $\{m\}_K = C_0C_1C_2\cdots C_n$ où $C_0 = I$ (un bloc d'initialisation) et les $C_i = \{C_{i-1} \oplus P_i\}_K$ pour $i = 1, \ldots, n$. La méthode CBC possède la propriété préfixe car : si $C_0C_1C_2\cdots C_iC_{i+1}\cdots C_n = \{P_1P_2\cdots P_iP_{i+1}\cdots P_n\}_K$ alors $C_0C_1C_2\cdots C_i = \{P_1P_2\cdots P_i\}_K$, ce qui implique que si nous connaissons $\{x, y\}_z$ nous pouvons en déduire $\{x\}_z$, si bien entendu la longueur de x est un multiple de la longueur de bloc utilisée. Cette propriété semble anodine, mais permet de monter des attaques sur des protocoles comme les deux protocoles décrits ci-après. Ensuite nous présentons les quelques travaux étudiant cette propriété.

Récemment, pour un nombre non borné de sessions V. Cortier *et alii* [CRZ05] prouve par résolution de clauses de Horn que le problème de sécurité pour CBC est décidable.

Exemples de protocoles.

La propriété de préfixe permet à partir d'un message chiffré de déduire chacun des préfixes du message chiffré. Cette propriété dépend fortement de l'algorithme de chiffrement utilisé, comme la méthode ECB ou CBC détaillée précédemment.

Nous présentons deux protocoles : le protocole de Denning-Sacco à clef symétrique et le protocole de Needham-Schroeder à clef symétrique. Si nous utilisons une méthode de chiffrement qui a la propriété de préfixe, ces deux protocoles présentent une attaque. Notons qu'en 1996 S. Bellovin [Bel96] s'est également intéressé à cette propriété d'un point de vue plus cryptographique.

Protocole de Denning-Sacco à clef symétrique avec CBC.

En 1981, D. E. Denning et G. M. Sacco [DS81] proposent une version modifiée du protocole de Needham-Schroeder à clef symétrique avec timestamps.

Description du protocole : Ce protocole permet d'échanger la clef symétrique Kab entre deux participants A et B en utilisant un serveur de confiance S, des timestamps et des clefs symétriques entre les participants et le serveur. Le premier participant A informe le serveur qu'il souhaite communiquer avec B. Le serveur lui renvoie chiffrés avec la clef symétrique partagée entre S et A, l'identité de B, la nouvelle clef Kab entre A et B, un marqueur temporel T et un message chiffré avec la clef symétrique entre S et B. Ce message alors transmis à B par A contient l'identité de A, la nouvelle clef entre A et B et le même marqueur temporel, permettant à B de s'assurer de la fraîcheur de la clef ainsi reçue.

$$\begin{array}{llcll} 1 & A & \rightarrow & S: & A, B \\ 2 & S & \rightarrow & A: & \{B, Kab, T, \{A, Kab, T\}_{Kbs}\}_{Kas} \\ 3 & A & \rightarrow & B: & \{A, Kab, T\}_{Kbs}, A, B \end{array}$$

Protocole 2.4.3: Protocole de Denning-Sacco à clef symétrique.

Attaque : Y. Chevalier et L. Vigneron [CV02] montent une attaque sur ce protocole en remarquant que les messages 2 et 3 contiennent le même terme $\{A, Kab, T\}_{Kbs}$. Il est nécessaire que les noms des agents, les nonces et les timestamps soient un multiple de la longueur des blocs utilisés par la méthode de chiffrement.

Lors de cette attaque l'intrus prend l'identité de l'agent B dans une première session avec le serveur S. En utilisant la réponse du serveur $\{A, Kab, T, \{B, Kab, T\}_{Kas}\}_{Kbs}$ et la propriété de préfixe, l'intrus est capable de déduire le message $\{A, Kab, T\}_{Kbs}$. Enfin l'intrus envoie ce message à B dans une seconde session ii du protocole. En recevant ce message envoyé par l'intrus, B accepte une nouvelle valeur pour la clef symétrique qu'il pense partager avec A, alors que A ne participe pas au protocole. Par ce processus l'intrus en jouant cette seconde session du protocole avec B prend la place de A aux yeux de B, ainsi l'intrus et B partagent la même clef supposée secrète.

$$\begin{array}{rrcll} i.1 & I(B) & \rightarrow & S: & B, A \\ i.2 & S & \rightarrow & I(B): & \{A, Kab, T, \{B, Kab, T\}_{Kas}\}_{Kbs} \\ ii.3 & I(A) & \rightarrow & B: & \{A, Kab, T\}_{Kbs} \end{array}$$

Protocole Needham-Schroeder à clef symétrique avec CBC.

En 1978, R. Needham et M. Schroeder [NS78a] proposent un protocole qui tente d'établir une clef symétrique Kab entre deux participants A et B à l'aide d'un serveur S lors d'une session.

Description du protocole : Les trois premiers messages assurent la distributions d'une clef symétrique Kab entre deux agents A et B grâce au serveur. Les deux derniers messages permettent aux participants de s'authentifier en utilisant la fonction successeur, dénotée par *succ*.

Attaque : Nous présentons ici l'attaque découverte par O. Pereira et J.-J. Quisquater [PQ00]. Cette attaque est basée sur la propriété de préfixe du mode de chiffrement CBC. Supposons que le message $\{Na, B, Kab, \{Kab, A\}_{Kbs}\}_{Kas}$ échangé lors de la seconde communication du protocole

$$\begin{array}{llcll}
1 & A & \rightarrow & S: & A, B, Na \\
2 & S & \rightarrow & A: & \{Na, B, Kab, \{Kab, A\}_{Kbs}\}_{Kas} \\
3 & A & \rightarrow & B: & \{Kab, A\}_{Kbs} \\
4 & B & \rightarrow & A: & \{Nb\}_{Kab} \\
5 & A & \rightarrow & B: & \{succ(Nb)\}_{Kab}
\end{array}$$

Protocole 2.4.4: Protocole de Needham-Schroeder à clef symétrique.

soit tel que tous ses composants soient de la longueur d'un bloc utilisé pour CBC. Par conséquent un intrus peut extraire le message suivant $\{Na, B\}_{Kas}$ et se faire passer pour B dans une nouvelle session avec A, ainsi il peut intercepter la clef secrète Na que A croit secrète entre A et B.

$$\begin{array}{rrcll}
i.1 & A & \rightarrow & S: & A, B, Na \\
i.2 & S & \rightarrow & A: & \{Na, B, Kab, \{Kab, A\}_{Kbs}\}_{Kas} \\
ii.3 & I(B) & \rightarrow & A: & \{Na, B\}_{Kas} \\
ii.4 & A & \rightarrow & I(B): & \{Na'\}_{Na} \\
ii.5 & I(B) & \rightarrow & A: & \{succ(Na')\}_{Na}
\end{array}$$

Résultats théoriques existants.

Y. Chevalier *et alii* [CKRT03] ont montré en utilisant leur technique que la propriété de préfixe considérée sous la méthode de chiffrement CBC vérifie bien leurs conditions d'oracle. Ils en déduisent que le problème de sécurité en présence de la propriété préfixe est co-NP-complet.

Récemment, S. Kremer et M. Ryan [KR05a] utilisent la méthode de chiffrement CBC et l'outil de B. Blanchet Proverif [Bla01] pour trouver une attaque sur le protocole de Needham-Schroeder-Lowe à clef publique [Low95].

2.4.2.4 « Ou exclusif ».

De nombreux protocoles utilisent le « ou exclusif », cet opérateur dénoté par le symbole de fonction n-aire $\oplus$ a les propriétés ($E1$) d'associativité, ($E2$) de commutativité, ($E3$) l'existence d'un élément neutre 0 et la propriété ($E4$) de nilpotence.

$$\begin{array}{rcll}
x \oplus (y \oplus z) & = & (x \oplus y) \oplus z & \text{(E1)} \\
x \oplus y & = & y \oplus x & \text{(E2)} \\
x \oplus 0 & = & x & \text{(E3)} \\
x \oplus x & = & 0 & \text{(E4)}
\end{array}$$

Ces axiomes sont généralement dénotés par ACUN, pour Associativité, Commutativité, Unité et Nilpotence. Nous présentons deux protocoles utilisant le « ou exclusif » dans leur spécification et sur lesquels il est possible de monter une attaque qui utilise elle aussi les propriétés du « ou exclusif ». Il existe bien entendu de nombreux autres protocoles qui se servent du « ou exclusif » dans la littérature, comme par exemple le « Gong's Mutual authentication Protocol » [Gon89]. Nous présentons deux protocoles se servant du « ou exclusif » et nous évoquerons ensuite les travaux existants qui prennent en compte cet opérateur.

Exemples de protocoles.

Nous décrivons d'abord le protocole d'authentification de Bull et ensuite le protocole WEP. Ces deux exemples de protocoles utilisent dans leur spécification explicitement le « ou exclusif ».

Protocole d'authentification de Bull.
Ce protocole inventé en 1997 par J. Bull et D. J. Otway [BO97] permet l'échange d'une nouvelle clef entre plusieurs participants et un serveur S. Ce protocole utilise également dans sa spécification le « ou exclusif ».

Description du protocole : Ce protocole fonctionne avec un nombre arbitraire n d'agents et un serveur S. Une fois le protocole fini, chaque participant x et y partage une nouvelle clef de session Kxy connue uniquement des deux participants x et y. Pour plus de simplicité, nous présentons le protocole pour trois participants A, B et C. L'agent A commence le protocole et communique avec le serveur en passant par le participant B puis par le participant C. Ainsi la nouvelle clef Kab est une clef uniquement partagée entre A et B et la nouvelle clef Kcb est une clef uniquement partagée entre B et C.

$$
\begin{array}{llcll}
1 & A & \rightarrow & B: & Xa = h((A,B,Na),Kas),A,B,Na \\
2 & B & \rightarrow & C: & Xb = h((B,C,Nb,Xa),Kbs),B,C,Nb,Xa \\
3 & C & \rightarrow & S: & Xc = h((C,S,Nc,Xb),Kcs),C,S,Nc,Xb \\
4 & S & \rightarrow & C: & A,B,Kab \oplus h(Na,Kas),\{A,B,Na\}_{Kab}, \\
 & & & & B,A,Kab \oplus h(Nb,Kbs),\{B,A,Nb\}_{Kab}, \\
 & & & & C,B,Kbc \oplus h(Nc,Kcs),\{C,B,Nc\}_{Kbc} \\
5 & C & \rightarrow & B: & A,B,Kab \oplus h(Na,Kas),\{A,B,Na\}_{Kab}, \\
 & & & & B,A,Kab \oplus h(Nb,Kbs),\{B,A,Nb\}_{Kab}, \\
 & & & & B,C,Kbc \oplus h(Nb,Kbs),\{B,C,Nb\}_{Kbc} \\
6 & B & \rightarrow & A: & A,B,Kab \oplus h(Na,Kas),\{A,B,Na\}_{Kab}
\end{array}
$$

Protocole 2.4.5: Protocole d'authentification de Bull.

Attaque : En 1998, Ryan et Schneider ont découvert une attaque sur ce protocole [RS98], connue sous le nom de « domino attack ». Cette attaque permet à C un intrus de connaître la nouvelle clef Kab partagée uniquement par A et B. Tout d'abord, l'agent C connaît les messages suivants $Kab \oplus h(Nb,Kbs)$ et $Kbc \oplus h(Nb,Kbs)$ qui lui sont renvoyés par le serveur S à l'étape 4. Puisque C connaît la clef Kbc il peut alors faire le « ou exclusif » entre ces deux messages et cette clef, pour obtenir la clef Kab. Cette attaque en « domino » se généralise bien entendu à n participants.

Protocole « Wired Equivalent Privacy ».
Ce protocole de communication pour les réseaux sans fil est plus connu sous le nom de protocole WEP [80299]. Ce protocole est très simple, il permet de protéger les échanges de données lors de communication wifi.

Description du protocole : L'agent A chiffre le message M en calculant le « ou exclusif » entre $RC4(v,Kab)$ et la paire $\langle M, C(M)\rangle$ où la fonction C est une fonction d'intégrité de somme. La fonction $RC4(v,k)$ est initialisée par le vecteur v et la clef k, et produit une longue séquence de bits pseudo-aléatoires. Le protocole est constitué d'un seul envoi de message où l'agent A envoie alors le vecteur initial v et le chiffré de M.

Attaque : N. Borisov *et alii* présentent plusieurs attaques sur ce protocole [BGW01]. Elles nécessitent les propriétés du « ou exclusif » et la propriété de la fonction C utilisée. Nous présentons deux de ces attaques.

$$1 \quad A \quad \rightarrow \quad B : \quad v, \langle M, C(M) \rangle \oplus RC4(v, Kab)$$

Protocole 2.4.6: Protocole WEP (Wired Equivalent Privacy).

La première attaque se sert uniquement des propriétés du « ou exclusif ». Nous considérons les chiffrés $C1$ et $C2$ des messages $P1$ et $P2$ avec le même vecteur v et la même clef k. Nous calculons le « ou exclusif » de $C1$ et $C2$:

$$\begin{array}{rcll} C1 \oplus C2 & = & \langle P1, C(P1) \rangle \oplus RC4(v,k)) \oplus (\langle P2, C(P2) \rangle \oplus RC4(v,k)) & \\ & = & \langle P1, C(P1) \rangle \oplus \langle P2, C(P2) \rangle) \oplus (RC4(v,k) \oplus RC4(v,k)) & (E1)(E2) \\ & = & \langle P1, C(P1) \rangle \oplus \langle P2, C(P2) \rangle & (E3)(E4) \end{array}$$

En utilisant les axiomes $(E1)$, $(E2)$, $(E3)$ et $(E4)$ un agent, s'il connaît le message $P1$ chiffré par $C1$, peut déchiffrer $C2$ et donc retrouver $P2$.

La seconde attaque permet à un intrus qui connaît un message D, de modifier un message chiffré envoyé, sans modifier la valeur de la fonction d'intégrité de somme. Elle utilise une propriété d'homomorphisme de la fonction C implémentée par « CRC-32 ». Nous détaillerons cette attaque dans la section 2.5.2.1 page 38 où nous présentons la propriété d'homomorphisme en détail.

Résultats théoriques existants.

Le « ou exclusif » est présent dans de nombreux protocoles et a suscité beaucoup d'intérêt ces dernières années. H. Comon-Lundh et V. Shmatikov [CLS03] proposent une procédure de décision basée sur la résolution de contraintes pour résoudre le problème de sécurité en présence du « ou exclusif ». La même année, Y. Chevalier *et al.* [CKRT03] obtiennent eux aussi une procédure de décision pour ce problème pour une classe de protocoles cryptographiques plus restreinte.

En 2003, H. Comon-Lundh et V. Cortier [CLC03] prouvent un résultat de décidabilité pour une extension de la classe de Skolem de la logique du premier ordre pour la théorie équationnelle du « ou exclusif ». Comme application directe de ce résultat, les auteurs obtiennent que l'analyse formelle des protocoles cryptographiques en présence du « ou exclusif » soit décidable pour un nombre non borné de sessions. Pour cela ils supposent un nombre fini de nonces et que chaque message ne contienne au plus qu'une copie d'une partie du message reçu. Notons également que K. Verma [Ver03] a prouvé la décidabilité pour une sous-classe de la logique du premier ordre en considérant les automates d'arbres bi-directionnel (« two-way tree automata ») en présence du « ou exclusif ».

L'outil Casper [Low97a] prend en compte le « ou exclusif » dans le cas du chiffrement de Vernam. Rappelons que le chiffré de Vernam du message m par la clef k est simplement le message $m \oplus k$. Pour modéliser cela, G. Lowe ajoute une nouvelle règle de déduction à l'intrus pour lui permettre de calculer $m \oplus m'$ à partir de m et m', et déduire m à partir de k et $m \oplus k$. Grâce à Casper, G. Lowe et A.W. Roscoe [LR97] ont retrouvé une faille sur le protocole TMN [TMN89], décrit plus précisément dans la section 2.5.2.4 page 39. Ils proposent alors une modification pour ce protocole qu'ils prouvent sûre grâce à leur outil.

2.4.2.5 Groupe abélien et exponentielle modulaire.

Présentons d'abord les axiomes des groupes abéliens : le symbole $\times$ correspond à un opérateur binaire multiplicatif des groupes abéliens. Un groupe abélien possède les propriétés suivantes :

$$\begin{aligned}
x \times (y \times z) &= (x \times y) \times z && \text{(associativité) (E1)}\\
x \times y &= y \times x && \text{(commutativité) (E2)}\\
x \times 1 &= x && \text{(élément neutre) (E3)}\\
x \times x^{-1} &= 1 && \text{(inverse) (E4)}
\end{aligned}$$

Nous notons par $exp(x, a)$, x à la puissance a. Cette théorie équationnelle possède les quatre axiomes énoncés précédemment des groupes abéliens, plus les deux axiomes suivants :

$$\begin{aligned}
exp(exp(x, y), z) &= exp(x, y \times z) \text{ (E5)}\\
exp(x, 1) &= x \text{ (E6)}
\end{aligned}$$

Exemples de protocoles.

Nous ne donnons pas d'exemple particulier de protocoles qui utilisent uniquement les propriétés des groupes abéliens car l'usage de ces propriétés est courant dans les protocoles cryptographiques, mais presque toujours associée avec d'autres propriétés, comme l'exponentielle modulaire. Nous mentionnons une liste non exhaustive des protocoles utilisant ces propriétés : le protocole dit de « Schnorr's » [CDS94], le protocole MAKEP [JP02], le protocole SRP [Wu98], le protocole AMP-3 [Kwo03]. Mentionnons aussi l'algorithme d'échange de clef de Fortezza [Nat98] et le protocole PAK-Z [Kwo03, MPS00] qui sont deux protocoles qui eux utilisent le « ou exclusif » et l'exponentielle modulaire. Nous présentons en particulier le protocole IKA.1, connu aussi sous le nom de GDH.2.

IKA.1.

Le Initial Key Agreement protocole, pour IKA.1, a été introduit par Steiner *et alii* en 1996 [STW96].

Description du protocole : Ce protocole permet d'établir une clef entre un nombre fixé n de participants $A_1, \ldots, A_n$. Chaque agent connaît son nonce secret N_i et connaît la base g de l'exponentielle. La clef partagée par les participants du groupe est g élevé à la puissance du produit de tous les nonces. Nous distinguons trois types d'agents : A_1 qui débute le protocole, le dernier agent A_n qui termine le protocole et distribue à chaque agent la clef partielle et les autres agents qui constituent une chaîne de A_1 à A_n. Le protocole fonctionne comme suit : l'agent A_1 commence par envoyer à l'agent A_2 le message : exp(g,1),exp(g,N$_1$). Puis l'agent A_i pour $i = 2, \ldots, n-1$:

- reçoit un message $m = m_1, \ldots, m_i$.
- enregistre la dernière composante de m dans la variable $x = m_i$.
- élève à la puissance N_i tous les m_j du message m, pour obtenir $m'_j = {m_j}^{N_i}$, $1 \leq j \leq i$.
- crée un nouveau message m' tel que $m' = m'_1, \ldots, m'_{i-1}, x, m'_i$.
- envoie ce nouveau message m' à l'agent A_{i+1}.

Le dernier agent A_n reçoit un message composé de n parties. Il les élève toutes à la puissance N_n. La dernière partie de ce message est la clef de groupe et le reste constitue la clef partielle de groupe. L'agent A_n diffuse alors à tous les agents cette clef partielle. Ainsi chaque agent obtient, en élevant à la puissance N_i la clef partielle de groupe qu'il a reçue, la clef de groupe à la i position. Pour plus de lisibilité nous décrivons le protocole uniquement pour trois agents A, B et C.

Nous détaillons le calcul du message envoyé par B à la seconde étape :

$$\begin{aligned}
exp(exp(g, 1), Nb) &= exp(g, (1 \times Nb)) \quad (E5)\\
&= exp(g, Nb) \qquad (E1)(E2)(E3)
\end{aligned}$$

et

$$(exp(g, Na), Nb)) = exp(g, (Na \times Nb)) \quad (E5)$$

$$
\begin{array}{llll}
1 & A & \rightarrow & B: \quad g, g^{N_a} \\
2 & B & \rightarrow & C: \quad g^{N_b}, g^{N_a}, (g^{N_a})^{N_b} \\
3 & C & \rightarrow & A, B: \quad (g^{N_b})^{N_c}, (g^{N_a})^{N_c}
\end{array}
$$

Protocole 2.4.7: Protocole IKA.1 (Initial Key Agreement).

Nous présentons maintenant le détail du calcul effectué par l'agent A pour retrouver la clef de groupe $exp(g, Na \times Nb \times Nc)$:

$$
\begin{array}{rcll}
exp(exp(exp(g, Nb), Nc), Na) & = & exp(exp(g, Nb \times Nc), Na) & (E5) \\
& = & exp(g, Na \times Nb \times Nc) & (E1)(E2)(E5)
\end{array}
$$

Attaque : Il existe une attaque sur ce protocole [MD02, GLRV04], où l'intrus prend l'identité du dernier agent. Puis il renvoie au premier agent A le message suivant g, g^{N_a} intercepté entre A et B lors du premier message. Or comme $exp(exp(g, 1), Na) = exp(g, Na)$ l'agent A pense que la clef de groupe est $exp(g, Na)$ aussi connue de l'intrus. Cette attaque est possible car le protocole ne vérifie pas l'identité des agents. Pour éviter ce genre d'attaque les auteurs proposent alors une amélioration dans laquelle les agents s'autentifient [AST00].

Résultats théoriques existants.

Nous présentons d'abord les nombreux résultats sur les groupes abéliens seuls et ensuite ceux qui considèrent les groupes abéliens et l'exponentielle modulaire.

Groupes abéliens.

Le problème de déduction de l'intrus est décidable dans le cas des groupes abéliens comme l'ont montré H. Comon-Lundh et V. Shmatikov [CLS03].

Le problème de sécurité pour les groupes abéliens fut abordé par J. Millen et V. Shmatikov [MS03, Shm04, MS05]. Dans un premier temps [MS03], ils réduisent le problème de sécurité en un problème de satisfiabilité de contraintes. Ce problème de résolution de contraintes est ensuite réduit à la résolution d'équations quadratiques diophantiennes, problème en général indécidable. Par la suite, ils ramènent la résolution du système de contrainte en un système d'équation quadratique particulier [Shm04, MS05]. Ils proposent alors une méthode de résolution fort compliquée de ces systèmes particuliers d'équations quadratiques et obtiennent alors une procédure de décision pour le problème de sécurité en présence de la théorie équationnelle des groupes abéliens pour un nombre borné de sessions.

Également, pour un nombre non borné de sessions et avec un nombre de nonces borné, K. Verma [Ver03] propose un résultat de décidabilité pour une classe particulière de protocoles en présence de la théorie équationnelle des groupes abéliens. Cette classe correspond aux protocoles qui peuvent être modélisés par un automate d'arbre bi-directionnel. Il s'est aussi intéressé à la décidabilité de deux autres théories équationnelles assez proches. Premièrement, il considère les trois premiers axiomes *i.e.* ACN : Associativité, Commutativité et élément Neutre. Deuxièmement, il prend en compte les axiomes ACN plus les deux équations suivantes $-(x + y) = (-x) + (-y)$ et $-(-x) = x$.

Groupe abélien et exponentielle modulaire.

Cette théorie équationnelle prend en compte les propriétés de la multiplication et de l'exponentielle modulaire. Ainsi il est possible de modéliser l'exponentielle modulaire dite de Diffie-Hellman et les propriétés du chiffrement asymétrique RSA décrites dans la section 1.3.2.1 page 7 de l'introduction. Cette théorie équationnelle est présente dans de nombreux protocoles, soit dans la méthode de chiffrement employée, soit dans la conception même du protocole. Notons que tous les résultats présentés ci-après supposent que l'opérateur multiplicatif n'apparaît que dans l'exponentielle, ce qui évite de multiplier les exponentielles entre elles. Cette restriction est une condition nécessaire pour avoir un résultat de décidabilité. Tout d'abord D. Kapur *et alii* [KNW03] ont montré l'indécidabilité de l'unification pour la théorie de l'exponentielle modulaire avec la distributivité sur la multiplication, ce qui entraîne immédiatement l'indécidabilité du problème de sécurité, car comme nous le verrons dans la section 8.1 page 125, si l'unification est indécidable alors le problème de sécurité est aussi indécidable.

En 2003, J. Millen et V. Shmatikov [MS03] mentionnent que le problème de sécurité est encore une question ouverte même dans le cas d'un générateur de groupe fixé. Depuis, des avancées furent réalisées à ce sujet. Tout d'abord M. Boreale et M.G. Buscemi [BB03] proposent une procédure de décision qui nécessite une borne *a priori* sur le nombre de facteurs dans chaque produit. Ensuite, Y. Chevalier *et al.* [CKR$^+$03a] prouvent que le problème de sécurité est co-NP-complet en présence de la théorie équationnelle des groupes abéliens et de l'exponentielle modulaire pour un générateur arbitraire mais avec des restrictions sur la manière dont les agents honnêtes et l'intrus peuvent déduire de l'information à partir des produits des exposants. Enfin, en 2004 V. Shmatikov [Shm04, MS05] réduit le problème de sécurité de l'intrus pour cette théorie équationnelle à un système particulier d'équations quadratiques diophantiennes dont il propose une méthode de résolution, pour prouver la décidabilité du problème de sécurité.

Remarquons aussi que J. Goubault-Larrecq *et alii* [GLRV04] ont développé une implantation pour vérifier les protocoles cryptographiques qui utilisent l'exponentielle modulaire avec un générateur g fixé. Dans ce cadre, cet opérateur est modélisé par un symbole exp et un opérateur associatif et commutatif $\times$. Par conséquent le terme $exp(M_1 \times M_2)$ représente g à la puissance M_1 le tout à la puissance M_2 soit : $(g^{M_1})^{M_2}$. Une règle de déduction est ajoutée pour permettre à un agent ou à l'attaquant d'élever à une puissance connue M_2 par exemple, un terme connu $exp(M_1)$. Le résultat obtenu est donc $exp(M_1 \times M_2)$. Grâce à cette modélisation, les auteurs ont pu vérifier les protocoles d'échange de clefs IKA.1. Avec cette formalisation de la théorie équationnelle ils n'ont pas pris en compte tout le pouvoir de l'exponentielle modulaire et par conséquent n'ont pas retrouvé toutes les failles existantes sur ce protocole. Plus particulièrement, en ne modélisant pas les inverses dans leur abstraction ils ne sont pas capables de retrouver l'attaque proposée par O. Pereira et J.-J Quisquater [PQ01].

Enfin, notons que C. Meadows a enrichi son outil NRL [Mea96] afin de vérifier les protocoles utilisant cette théorie équationnelle, en particulier elle a analysé le protocole AGDH-2 [Mea00] qui est très similaire au protocole IKA.1.

2.4.2.6 Horodateurs (« Timestamps »).

Inclure la notion de temps dans les protocoles est une idée intuitive. Cette idée est en plein essor actuellement, elle se traduit par l'utilisation de marqueurs temporels appelés en anglais « timestamps », nous emploierons indifféremment horodateur, marqueur temporel et timestamps.

Exemples de protocoles.

Nous présentons le protocole « Wide Mouthed Frog » comme un exemple employant des marqueurs temporels.

Wide Mouthed Frog Protocol

Le protocole présenté ici est l'un des plus simples utilisant le temps et les horodateurs. Il fut proposé par M. Burrows *et alii* [BAN89].

Description du protocole : Ce protocole utilise un serveur S pour assurer que la clef Kab, utilisée par les agents A et B, est récente.

$$\begin{array}{llcll} 1 & A & \rightarrow & S: & A, \{Ta, B, Kab\}_{Kas} \\ 2 & S & \rightarrow & B: & \{Ts, A, Kab\}_{Kbs} \end{array}$$

Protocole 2.4.8: Protocole « Wide Mouthed Frog ».

Attaque : Un attaque fut trouvée en 1995 [AN95]. L'idée de cette attaque est qu'un intrus peut garder une clef Kab fraîche en utilisant le serveur comme un oracle de la manière suivante :

$$\begin{array}{lrcll} 1.1 & A & \rightarrow & S: & A, \{Ta, B, Kab\}_{Kas} \\ 1.2 & S & \rightarrow & B: & \{Ts, A, Kab\}_{Kbs} \\ 2.1 & I(B) & \rightarrow & S: & B, \{Ts, B, Kab\}_{Kbs} \\ 2.2 & S & \rightarrow & A: & \{Ts', B, Kab\}_{Kas} \\ 3.1 & I(A) & \rightarrow & S: & A, \{Ts', B, Kab\}_{Kas} \\ 3.2 & S & \rightarrow & B: & \{Ts'', A, Kab\}_{Kbs} \end{array}$$

Cette attaque peut être évitée si un agent s'aperçoit de la redondance des messages échangés et ignore donc ces messages, en pensant à juste titre qu'il s'agit d'un attaquant.

Résultats théoriques existants.

Les marqueurs temporels sont des données temporelles (une date, une heure) qui sont ajoutées aux messages dans les protocoles cryptographiques pour éviter qu'un agent ne réutilise un message déjà employé auparavant. La plupart des méthodes existantes de vérification de protocoles cryptographiques remplacent ces marqueurs par des nonces, car prendre en compte des notions temporelles rendrait la vérification plus complexe. Toutefois il existe des attaques basées sur la falsification des ces timestamps, comme l'attaque décrite précédemment sur le Wide Mouthed Frog protocol [BAN89].

Certains des travaux qui prennent en compte les timestamps dans les protocoles cryptographiques utilisent des assistants de preuves [ES00],ce qui nécessite l'aide de l'utilisateur, ou des vérificateurs de modèles (« model-checkers ») à état fini comme l'outil Casper [Low97a] qui lui, nécessite une borne sur la taille de l'espace de recherche du temps. En 2004, L. Bozga *et alii* [BEL04] propose une procédure de décision symbolique pour un nombre borné de sessions pour des protocoles cryptographiques utilisant les timestamps. Récemment, des procédures automatiques ont vu le jour pour aborder cette notion de timestamps. Citons l'approche développée par G. Delzanno et P. Ganty [DG04] basée sur la structure des données pour représenter les ensembles de configurations à explorer pour un nombre arbitraire de sessions en parallèle. Comme la vérification du problème de sécurité est indécidable en général pour un nombre non borné de session, la terminaison de leur procédure n'est pas garantie.

Autres théories équationnelles. Nous présentons d'autres théories équationnelles étudiées dans la littérature. Tout d'abord la propriété d'associativité de la paire représentée par l'égalité suivante $\langle\langle x, y\rangle, z\rangle = \langle x, \langle y, z\rangle\rangle$ est prise en compte dans certains outils de manière partielle, comme par exemple Casrul [JRV00]. Ceci est dû au fait que l'unification modulo Associativité est décidable [Mak77] mais infinitaire [Abd87, AP90, Jaf90, Plo72, Sie75, LS75]. Les modélisations de cette propriété permettent de trouver des attaques dites de « type », comme sur le protocole de Needham-Schroeder avec chiffrement ECB que nous présenterons dans la section 2.5.2.2 page 38.

Certains autres résultats proposent un cadre générique pour la vérification de la sécurité des protocoles cryptographiques. Par exemple, M. Abadi et V. Cortier [AC04a, AC05a] montrent que le problème de déductibilité et le problème de l'équivalence statique sont décidables en temps polynomial pour des théories sous-termes convergentes, *i.e.* des théories décrites par des équations de la forme $M = N$ où N est un sous-terme de M. En même temps, S. Delaune et F. Jacquemard [DJ04] prouvent que le problème de sécurité pour un intrus actif est décidable (co-NP-complet) pour une classe un peu plus restreinte que celle introduite par M. Abadi et V. Cortier. Ce dernier résultat permet de considérer par exemple la théorie de Dolev-Yao standard [DY83] ou les destructeurs explicites. Le modèle avec destructeurs explicites donne à l'intrus le pouvoir d'appliquer la fonction de déchiffrement sur n'importe quel message, il faut alors considérer une théorie équationnelle qui permet de déchiffrer un message chiffré auquel la fonction de déchiffrement est appliquée. Remarquons qu'une généralisation de ce résultat a été proposée par M. Baudet [Bau05], une version plus détaillée des preuves de cet article figure dans sa thèse [Bau06]. De plus, deux travaux récents de J. Millen [Mil03] et de C. Lynch et C. Meadows [LM04] comparent l'approche avec destructeurs explicites et l'approche standard sans destructeur explicite présentée au chapitre 3 page 45.

Dernièrement, H. Comon-Lundh [CL04] tente de séparer l'analyse des capacités de l'intrus et les règles du protocoles. Dans ce travail, détaillé dans la thèse de V. Bernat [Ber06], ils présentent un système de déduction à la Dolev-Yao prenant le pouvoir de l'intrus comme paramètre. Dans cette approche, les règles du protocoles sont vues comme une capacité additionnelle pour l'intrus. Ils prennent ainsi en compte un nombre borné et non borné de sessions pour la vérification de la propriété de secret sans théorie équationnelle.

Récemment, D. Basin *et alii* [BMV05a] montrent un résultat de décidabilité uniforme et modulaire pour le problème de déduction de l'intrus en présence d'équations algébriques exprimant les propriétés des opérateurs cryptographiques. Les auteurs supposent des restrictions sur la taille des messages et sur l'analyse des messages que l'intrus peut effectuer. Ils partagent la théorie équationnelle en deux parties : une partie « théorie d'annulation » (C) (« cancellation theory ») représentée par un système de réécriture terminant et convergent, et une autre partie où la théorie équationnelle possède une classe d'équivalence fini (FEC) (« finite equivalence class »). Ils démontrent alors que si la théorie équationnelle est partageable en deux parties disjointes (C) et (FEC) alors le problème de déduction de l'intrus est décidable sous ces restrictions.

Enfin Y. Chevalier et M. Rusinovitch [CR05] ont montré un résultat de combinaison pour différentes théories équationnelles : Si le problème de sécurité est prouvé décidable pour deux théories équationnelles, qui ne partagent pas de symboles de fonction, alors le problème de sécurité est décidable pour l'union des deux théories. Ce résultat permet d'utiliser les différents résultats présents dans la Figure 2.2 page 25 pour obtenir de nouveaux résultats pour les combinaisons des différentes théories déjà analysées. Dans leur approche ils considèrent le modèle avec des destructeurs explicites pour modéliser le pouvoir de chiffrement et de déchiffrement de l'intrus.

2.5 Une propriété importante : l'homomorphisme.

Nous focalisons notre attention dans la suite de cette thèse sur la propriété d'homomorphisme. Nous donnons d'abord une définition de cette propriété et les quelques résultats de vérification existants qui la prennent en compte. Ensuite, nous présentons quelques protocoles l'utilisant. Enfin, nous annonçons le contenu des chapitres suivants.

2.5.1 Définition et travaux existants.

Un opérateur h possède la propriété d'homomorphisme par rapport aux opérateurs g et g' si :

$$h(g(x, y)) = g'(h(x), h(y))$$

Nous étudierons le cas particulier où $g = g'$, cette situation est plus complexe à analyser car nous pouvons rappliquer plusieurs fois la propriété, alors que quand $g \neq g'$, le changement de symbole bloque toute nouvelle application de la propriété.

Signalons d'abord qu'une des premières approches formelles pour affaiblir l'hypothèse du chiffrement parfait est le travail de S. Even *et alii* [EGS86]. Ils ont analysé le chiffrement RSA en essayant de prendre en compte l'opérateur multiplicatif $\times$ et la propriété suivante : $\{x\}_k \times \{y\}_k = \{x \times y\}_k$, si le même modulo est considéré. Ils considèrent uniquement la propriété entre les clefs et leurs inverses, modélisée par les deux équations suivantes : $\{\{x\}_k\}_{k^{-1}} = x$ et $\{\{x\}_{k^{-1}}\}_k = x$. Le résultat de leur article montre que si un protocole ping-pong est sûr dans leur modèle abstrait alors il est sûr en utilisant le chiffrement RSA.

Un autre résultat concernant cette théorie équationnelle, fut établi par H. Comon-Lundh et R. Treinen [CLT03] dans le cas passif. Ils montrent que le problème de déduction est décidable et illustrent leur résultat en montrant que le problème de déduction est décidable en temps polynomial en présence de la propriété d'homomorphisme instanciée de la manière suivante :

$$\{\langle x, y\rangle\}_k = \langle\{x\}_k, \{y\}_k\rangle$$

Cette propriété est vérifiée en particulier par la méthode de chiffrement ECB (Electronic Code Book), décrite dans la section 2.4.2.3 page 27. Nous verrons plus particulièrement dans la section 2.5.2.2 page suivante que cette propriété, entre le chiffrement et la paire, permet de trouver une attaque sur le protocole de Needham-Schroeder corrigé par G. Lowe. Nous présentons maintenant plusieurs protocoles qui utilisent la propriété d'homomorphisme.

2.5.2 Protocoles utilisant la propriété d'homomorphisme.

La propriété d'homomorphisme est présente sous différentes formes dans la littérature. Par exemple de nombreux protocoles de vote emploient une méthode de chiffrement qui utilise la propriété suivante :

$$\{x + y\}_k = \{x\}_k * \{y\}_k$$

Par exemple le chiffrement d'ElGamal [El 85], présenté en introduction, possède cette propriété.

D'autre part le chiffrement RSA possède la propriété d'homomorphisme suivante, si les égalités sont modulo n où la clef $k = (e, n)$:

$$\{x * y\}_k = \{x\}_k * \{y\}_k$$

car $\{x\}_k = x^e \bmod n$.

Nous présentons d'abord une autre attaque sur le protocole WEP qui utilise la propriété d'homomorphisme de la fonction d'intégrité C. Puis, nous montrerons comment la version corrigée du protocole de Needham-Schroeder possède une attaque si la méthode de chiffrement ECB est employée. Ensuite nous expliquerons comment les protocoles de vote se servent souvent de manière cruciale de la propriété d'homomorphisme. Puis nous présenterons le protocole TMN, ce protocole est notre protocole de référence pour l'analyse des protocoles en présence de l'axiome équationnel d'homomorphisme sur le « ou exclusif ».

2.5.2.1 Le protocole « Wired Equivalent Privacy » (WEP).

Nous reprenons le WEP protocole, décrit à la section 2.4.2.4 page 30, et présentons une deuxième attaque, décrite par N. Borisov *et alii* [BGW01], qui utilise le fait que la fonction d'intégrité C, implémentée par la fonction CRC-32, possède la propriété suivante d'homomorphisme :

$$C(x \oplus y) = C(x) \oplus C(y)$$

L'intrus modifie alors un message intercepté chiffré en faisant le « ou exclusif » du message de son choix et ainsi brouille ou change le message en préservant la somme d'intégrité, qui attestera de la validité du message. Pour cela, l'attaquant intercepte $\langle M, C(M)\rangle \oplus RC4(v, Kab)$ et connaît D. L'intrus peut alors obtenir le chiffré associé au message $M \oplus D$ en calculant :

$$\begin{array}{lll} & (\langle M, C(M)\rangle \oplus RC4(v, Kab)) \oplus \langle D, C(D)\rangle & \\ = & RC4(v, Kab) \oplus (\langle M, C(M)\rangle \oplus \langle D, C(D)\rangle) & (E1)(E2) \\ = & RC4(v, Kab) \oplus \langle M \oplus D, C(M) \oplus C(D)\rangle & (E6) \\ = & RC4(v, Kab) \oplus \langle M \oplus D, C(M \oplus D)\rangle & (E5) \end{array}$$

2.5.2.2 Le protocole Needham-Schroeder-Lowe avec ECB.

En 1996, G. Lowe [Low95] propose une correction au protocole de Needham-Schroeder, nous présentons ici cette correction. Ensuite nous montrons que si le chiffrement utilisé est la méthode Electronic Code Book (ECB) il existe une attaque. Rappelons juste la propriété ECB : $\{A, B, C\}_k = \{A\}_k, \{B\}_k, \{C\}_k$ où la taille des blocs A, B, C est un multiple de la taille de bloc utilisée par le chiffrement considéré.

La correction de G. Lowe au protocole de Needham-Shroeder consiste à rajouter l'identité de l'agent B dans sa réponse au premier participant A pour éviter que l'intrus ne réutilise ce message.

$$\begin{array}{lllll} 1 & A & \rightarrow & B: & \{Na, A\}_{pub(B)} \\ 2 & B & \rightarrow & A: & \{Na, Nb, B\}_{pub(A)} \\ 3 & A & \rightarrow & B: & \{Nb\}_{pub(B)} \end{array}$$

Protocole 2.5.1: Protocole de Needham-Schroeder-Lowe.

Attaque : Grâce à cette correction l'intrus ne peut plus réutiliser le message retourné par B. Mais, il peut jouer deux sessions du protocole et extraire le message $\{Na, Nb\}_{pub(A)}$ à partir de $\{Na, Nb, B\}_{pub(A)}$ grâce à la méthode de chiffrement ECB employée ici. L'intrus peut alors composer le message $\{Na, Nb, I\}_{pub(A)}$ et demander à A de lui déchiffrer Nb, dans la session qu'il joue avec l'agent A.

$$\begin{array}{llcll} 1.1 & A & \rightarrow & I: & \{A, N_a\}_{\mathsf{pub}(C)} \\ 2.1 & I(A) & \rightarrow & B: & \{A, N_a\}_{\mathsf{pub}(B)} \\ 2.2 & B & \rightarrow & I(A): & \{N_a, N_b, B\}_{\mathsf{pub}(A)} \\ 1.2 & B & \rightarrow & A: & \{N_a, N_b, I\}_{\mathsf{pub}(A)} \\ 1.3 & A & \rightarrow & I: & \{N_b\}_{\mathsf{pub}(I)} \\ 2.3 & I(A) & \rightarrow & B: & \{N_b\}_{\mathsf{pub}(B)} \end{array}$$

Cette attaque est possible uniquement si la méthode de chiffrement utilisée est ECB.

2.5.2.3 Protocole de décompte secret d'élection à plusieurs autorités.

Nous présentons une version simplifiée du décompte des voix du « Multi-Authority Secret Ballot Election Protocol » introduit par R. Cramer, M. Franklin, B. Schoenmakers et M. Yung [CGS97] en 1996. Cette étape utilise de manière cruciale la propriété suivante du chiffrement probabiliste utilisé :

$$\{m1, r1\}_K * \{m2, r2\}_K = \{m1 + m2, r\}_K$$

Citons parmi de nombreux chiffrements probabilistes existants qui vérifient cette propriété les chiffrements proposés par par T. ElGamal [El 85], P. Paillier [Pai99, FPS01], S. Goldwasser-S. Micali [GM84], J. Benaloh [Ben87, CF85], D. Naccache-J. Stern [NS97], T. Okamoto-S. Uchiyama [OU98].

Ce protocole permet d'organiser des élections avec un serveur S et A_i votants. Chaque votant choisit son vote V_i parmi deux valeurs possibles -1 ou $+1$. Lors d'un référendum la valeur -1 représente le vote pour le « non » et le $+1$ pour le « oui ». Ensuite chaque votant chiffre son vote grâce à la clef publique $pub(S)$ du serveur S. Le chiffrement utilisé est probabiliste ce qui évite la révélation d'information lors de la transmission du vote par une éventuelle « attaque par dictionnaire ». Dans le protocole, le vote contient également d'autres informations pour satisfaire certaines propriétés usuelles des protocoles de vote. Une fois tous les votes envoyés au serveur, le serveur calcule le produit de tous les messages reçus. Le serveur déchiffre alors, en un seule fois, ce produit pour obtenir le résultat des élections.

$$\begin{array}{rccll} i & A_i & \rightarrow & S: & Bi = \{Vi, ri\}_{pub(S)} \\ n+1 & S & \rightarrow & A_i: & S \text{ déchiffre } B1 * ... * Bn \text{ et publie le résultat du vote.} \end{array}$$

Protocole 2.5.2: Protocole de décompte secret d'élection à plusieurs autorités.

Grâce à la propriété d'homomorphisme, si le résultat est positif alors le « oui » emporte le référendum, s'il est négatif alors le « non » gagne, sinon il y a autant de votes pour que de vote contre. Avec cette technique, le serveur ne déchiffre jamais le vote d'un seul votant à la fois, il ne peut donc pas attribuer un vote à un votant. Cette propriété d'anonymat, introduite dans la section 2.1.2.2 page 14, est une des propriétés importantes qu'un protocole de vote doit vérifier.

2.5.2.4 Le protocole TMN.

En 1989 M. Tatebayashi, N. Matsuzaki et D.B. Newman proposent un protocole d'échange de clef [TMN89] entre deux participants utilisant un serveur S. Ce protocole utilise le « ou exclusif » dans son fonctionnement, si nous utilisons un chiffrement homomorphique cela entraîne une attaque.

Description du protocole : L'agent A génère un nonce Na. Il débute alors le protocole en envoyant au serveur ce nonce Na chiffré par la clef publique du serveur $pub(S)$. Le serveur prévient alors le second agent B que A souhaite communiquer avec lui. L'agent B renvoie au serveur son nonce Nb chiffré par $pub(S)$. Nb sera la clef que partageront les deux agents une fois le protocole fini. Le serveur peut alors déchiffrer les deux messages et renvoyer à l'agent A, le « ou exclusif » des deux nonces Na et Nb. Ainsi l'agent A peut retrouver Nb car il connaît Na.

$$\begin{array}{llcll} 1 & A & \rightarrow & S: & A, B, \{Na\}_{pub(S)} \\ 2 & S & \rightarrow & B: & S, A \\ 3 & B & \rightarrow & S: & B, A, \{Nb\}_{pub(S)} \\ 4 & S & \rightarrow & A: & S, B, Nb \oplus Na \end{array}$$

Protocole 2.5.3: Protocole d'échange de clef TMN.

Attaque : En 1994, G. J. Simmons présente une attaque sur ce protocole, si nous utilisons le chiffrement RSA [MMS97a]. Ce chiffrement, comme nous l'avons rappelé au début de cette section, possède la propriété d'homomorphisme suivante :

$$\{x * y\}_k = \{x\}_k * \{y\}_k$$

Cette attaque met en jeu un intrus I et son complice M, l'intrus connaît son nonce N_I et celui de son complice N_M. L'intrus intercepte dans une session normale le message $\{Nb\}_{pub(S)}$ et joue le protocole normalement avec son complice M. Au lieu d'envoyer $\{N_I\}_{pub(S)}$ au serveur, il envoie le produit de ce message et du message intercepté soit : $\{N_I\}_{pub(S)} * \{Na\}_{pub(S)}$ ce qui est égal à $\{N_I * Na\}_{pub(S)}$ car le chiffrement est homomorphique. Le serveur après avoir reçu $\{N_M\}_{pub(S)}$ de la part du complice de l'intrus, retourne $N_I * Na \oplus N_M$. L'intrus peut alors retrouver Na.

$$\begin{array}{llcll} 1 & I & \rightarrow & S: & I, M, \{N_I\}_{pub(S)} * \{Na\}_{pub(S)} = \{N_I * Na\}_{Pub(S)} \\ 2 & S & \rightarrow & M: & S, I \\ 3 & M & \rightarrow & S: & M, I, \{N_M\}_{pub(S)} \\ 4 & S & \rightarrow & I: & S, M, (N_I * Na) \oplus N_M \end{array}$$

En 1997, J. Mitchell *et alii* modélisent ce protocole pour l'analyser avec l'outil Murϕ [DDHY92]. Ils n'arrivent pas à retrouver [MMS97a] l'attaque proposée par G. J. Simmons [Sim94] . La même année, Lowe et Roscoe [LR97] utilisent l'outil FDR [Ros94, Ltd97] et CSP [Hoa85, Ros97] pour analyser ce protocole et trouvent de nombreuses attaques car les messages ne sont pas authentifiés. Ils découvrent une attaque semblable à celle de G. J. Simmons en rejouant le premier message d'une session correcte du protocole entre A et B. Cette attaque peut être supposée non-valide car nous pouvons imaginer qu'un serveur refuse de rejouer le protocole avec un même message pour le premier échange.

Pour prendre en compte toute la puissance de l'attaque de G. J. Simmons, nous devons enrichir les capacités de l'intrus par la propriété d'homomorphisme sur l'opérateur de groupe abéliens. Remarquons, si le chiffrement possède la propriété suivante d'homomorphisme : $\{a\}_k \oplus \{b\}_k = \{a \oplus b\}_k$, il est possible d'adapter cette attaque. Cette attaque nécessite un intrus actif, si nous considérons l'attaque pour le $\oplus$ et abstrayons le chiffrement par la clef publique du serveur par un symbole homomorphique h que seul le serveur peut détruire, alors nous pouvons appliquer la procédure que nous élaborons dans le chapitre III page 119 pour retrouver cette attaque (pour une présentation complète de cet exemple consulter [DLLT05]).

2.5.3 Nos contributions.

Après avoir répertorié et classé [CDL06] par théories équationnelles les protocoles cryptographiques, nous étudions plus particulièrement certaines théories équationnelles possédant un symbole homomorphique dénoté h sur un opérateur dénoté par le symbole $\oplus$. Les théories équationnelles étudiées possèdent certains des axiomes suivants :

1. $(x \oplus y) \oplus z = x \oplus (y \oplus z)$ (Associativité A)
2. $x \oplus y = y \oplus x$ (Commutativité C)
3. $0 \oplus x = x$ (Unité U)
4. $x \oplus x = 0$ (Nilpotence N)
5. $x \oplus I(x) = 0$ (Inverse I)
6. $h(x \oplus y) = h(x) \oplus h(y)$ (Homomorphisme h)

Nous nous intéressons dans cette thèse aux théories équationnelles en présence de l'axiome 6 d'homomorphisme pour le symbole de fonction h et les différentes théories équationnelles pour le symbole $\oplus$ que nous dénotons par :

- ACh la théorie équationnelle constituée des axiomes 1, 2, et 6.
- ACUNh la théorie équationnelle constituée des axiomes 1, 2, 3, 4 et 6, cette théorie correspond à un homomorphisme sur le « ou exclusif ».
- AGh la théorie équationnelle constituée des axiomes 1, 2, 3, 5 et 6, dans ce cas nous parlerons de l'homomorphisme sur les groupes abéliens.

Dans la seconde partie de ce document, nous présentons le modèle de Dolev-Yao et cherchons à l'étendre pour prendre en compte des théories équationnelles, théories qui peuvent être représentées par un système de réécriture confluent. Puis, nous donnons une approche pour le cas passif dans le modèle de Dolev-Yao étendu, dans laquelle nous montrons un premier résultat générique de localité simple en présence des théories équationnelles. Ensuite nous montrons comment ce résultat peut être appliqué au cas du modèle classique de Dolev-Yao pour retrouver le résultat de déduction pour le problème de l'intrus. Puis nous l'appliquons au cas des théories équationnelles suivantes : AGh, ACUNh et ACh. En s'inspirant de nos travaux initiaux [LLT05, LLT04] qui traitaient pour la première fois la propriété d'homomorphisme sur ces opérateurs associatifs et commutatifs dans le cas passif, S. Delaune proposa une procédure P-TIME [Del06a] pour résoudre le problème de déduction de l'intrus pour ces théories équationnelles. L'idée principale de ce travail est de remplacer différentes applications des règles de construction pour le symbole homomorphique h et le symbole $\oplus$ par une seule « macro » règle de déduction. Nous étudierons ensuite les théories ACUNh et AGh, lorsque le symbole homomorphique est le symbole de chiffrement commutatif ou pas. Nous proposons alors une procédure de décision pour le problème de déduction de l'intrus en présence de ces théories. Enfin nous montrons les relations existantes entre le problème d'unification et le problème de déduction de l'intrus.

Dans la dernière partie, nous focalisons notre attention sur le cas actif pour la théorie ACUNh et proposons une procédure de décision pour le problème de sécurité. Nous caractérisons d'abord une classe de protocoles, les protocoles déterministes. Ensuite, nous montrons le lien entre l'unification et le problème de sécurité. Puis élaborons une méthode de résolution pour une classe particulière d'équations diophantiennes. Enfin nous présentons notre procédure de décision pour un nombre borné de sessions en présence de la théorie équationnelle ACUNh. Cette procédure nous permet de retrouver [DLLT05] l'attaque découverte par G. J. Simmons sur le protocole de TMN, en modélisant le chiffrement asymétrique du serveur par la fonction homomorphique h..

Deuxième partie

Problème de déduction de l'intrus (intrus passif).

Chapitre 3

Extension du modèle de Dolev-Yao par des propriétés algébriques.

« Obvious" is the most dangerous word in mathematics. »
Eric Temple Bell.

Sommaire

Au début des années quatre-vingt, Dolev et Yao [DY81] ont modélisé les capacités de l'intrus par un modèle de déduction. Nous présentons d'abord le modèle originel de Dolev-Yao et ensuite l'extension usuelle de ce modèle par une théorie équationnelle. Cette extension augmente le pouvoir de déduction de l'intrus et permet ainsi une vérification des protocoles cryptographiques contre un attaquant plus puissant, ce qui assure un plus grand niveau de sécurité.

Ensuite, nous regardons plus précisément les théories équationnelles possédant un symbole binaire *associatif et commutatif* (AC). Nous séparons ce symbole de fonction n-aire, des autres symboles de la théorie, et nous le représentons par une nouvelle règle de déduction (GX) dans le modèle étendu de Dolev-Yao. Nous utiliserons dans la suite de la thèse ce modèle étendu par des théories équationnelles.

3.1 Le modèle de Dolev-Yao.

En 1981, Dolev et Yao [DY81, DY83] proposent un modèle de l'intrus pour analyser les protocoles cryptographiques de communications. Ce modèle est constitué d'un système de preuve dédié aux protocoles cryptographiques et représente les capacités que possèdent un attaquant pour apprendre de l'information, par exemple s'il connaît une clef et un terme il peut chiffrer ce terme avec cette clef. Nous donnons les hypothèses faites par le modèle de Dolev-Yao, avant de le présenter plus en détail.

3.1.1 Hypothèses du modèle.

Le modèle de Dolev-Yao est un des premiers modèles formels pour la vérification de protocoles cryptographiques. Cette formalisation des protocoles idéalise les protocoles en considérant les points suivants.

3.1.1.1 Un réseau idéalisé.

Le modèle de Dolev-Yao ne prend pas en compte la notion de temps, ni les pertes de messages. Nous supposons un réseau « idéal » *i.e.* aucun message échangé n'est perdu et tous les messages sont envoyés et reçus instantanément par les agents. Ainsi, avec le modèle introduit par Dolev et Yao, il n'est pas possible de prendre en compte le temps dans les capacités de l'attaquant. De plus, dans ce modèle, tous les canaux de communications sont supposés être publics, ainsi nous sommes dans un environnement le plus hostile possible pour effectuer la vérification.

3.1.1.2 Abstraction des messages échangés.

En réalité, les messages échangés par des agents lors d'une communication sont des chaînes de bits. Dans ce modèle, comme dans de nombreuses approches symboliques en vérification de protocoles, les messages échangés sont abstraits par des termes générés par une algèbre de termes. Dans le modèle initial de Dolev-Yao, cette algèbre de termes ne contient que les symboles des fonctions de chiffrement, de construction d'une paire et un ensemble de termes constants. De plus, les identités des participants honnêtes ainsi que les nonces sont représentés par des constantes de la signature.

3.1.1.3 Hypothèse du chiffrement parfait.

Le modèle, proposé en 1981 par Dolev et Yao, fait l'hypothèse du chiffrement parfait. Les messages chiffrés sont donc analysés comme des « boites noires » contenant un message qu'il n'est possible d'ouvrir uniquement si la clef de déchiffrement est connue.

3.1.1.4 Connaissance initiale de l'intrus.

Dans le modèle de Dolev-Yao, un intrus peut utiliser n'importe quel terme qu'il a observé sur le réseau précédemment et ce, autant de fois qu'il le désire. Nous disons habituellement que « l'intrus est le réseau » ainsi, les messages, qui circulent sur le réseau, sont connus par l'intrus et constituent l'ensemble de sa connaissance initiale.

3.1.1.5 Capacités de l'intrus.

Dans ce système, l'intrus peut appliquer une des règles données par le modèle. Il peut déchiffrer un message s'il connaît la clef de déchiffrement, il peut chiffrer un message avec n'importe quelle clef en sa possession et il est capable de construire et détruire des paires de termes. Ces capacités sont modélisées par un système de déduction que nous présentons dans la section suivante.

3.1.2 Le système de déduction de Dolev-Yao.

En considérant toutes ces hypothèses, le modèle de Dolev-Yao est constitué d'un système de déduction logique, où la connaissance initiale de l'intrus est représentée par un ensemble de termes T. Ces termes correspondent aux différents messages échangés lors du protocoles, à certains nonces, aux clefs publiques et aux identités des différents agents.

Nous rappelons quelle est la structure de ce système de preuve, avant de présenter, dans la figure 3.1, les différentes règles du modèle de Dolev-Yao représentant le pouvoir accordé à l'intrus.

Le modèle usuel de Dolev-Yao [DY81] définit les capacités de déduction de l'intrus en supposant l'hypothèse de chiffrement parfait. Nous notons par le séquent $T \vdash u$ le fait que l'intrus peut déduire le terme u à partir de sa connaissance initiale T en utilisant les règles de son système de déduction. Ce système de déduction décrit par la figure 3.1 est composé des règles suivantes : (A) l'intrus a observé sur le réseau un terme et il le connaît, (P) l'intrus peut construire une paire à partir de deux termes, (UL), (UR) il sait également extraire chaque composante d'une paire, (C) l'intrus peut alors chiffrer un message m par une clef k, (D) s'il connaît la clef k il peut alors déchiffrer un message chiffré par cette clef.

$$
\begin{array}{ll}
(A)\dfrac{u \in T}{T \vdash u} & (UL)\dfrac{T \vdash \langle u, v\rangle}{T \vdash u} \\
(P)\dfrac{T \vdash u \qquad T \vdash v}{T \vdash \langle u, v\rangle} & (UR)\dfrac{T \vdash \langle u, v\rangle}{T \vdash v} \\
(C)\dfrac{T \vdash u \qquad T \vdash v}{T \vdash \{u\}_v} & (D)\dfrac{T \vdash \{u\}_v \qquad T \vdash v}{T \vdash u}
\end{array}
$$

FIG. 3.1 – Le modèle de déduction de Dolev-Yao.

Nous présentons ici le modèle de Dolev-Yao pour un chiffrement symétrique, ceci afin d'être le plus clair possible dans la suite du document. Considérer le chiffrement asymétrique demande de distinguer les clefs publiques et les clefs privées, ce qui implique de modifier le système de déduction de Dolev-Yao. Les résultats obtenus dans cette thèse se généralisent aux chiffrement à clefs publiques en prenant en compte ces modifications dans le modèle de Dolev-Yao.

Définition 1 (Taille d'une preuve) *La taille d'une preuve P dans ce système de déduction, notée $|P|$, correspond au nombre de nœuds de la preuve P.*

Définition 2 (Preuve minimale) *Une preuve P de $T \vdash u$ est* minimale *s'il n'existe pas de preuve P' de $T \vdash u$ avec moins de nœuds que P i.e. telle que $|P'| < |P|$.*

Dans ce système de preuve, une règle est appliquée si les hypothèses de cette règles sont vraies *i.e.* il existe une condition de filtrage (« matching ») implicite sur les hypothèses de la règle, lors de son application. Une règle de déduction n'est appliquée que s'il existe des termes dans T qui filtrent (« match ») le motif de terme décrit par la règle.

Des extensions de ce modèle furent proposées par la suite pour prendre en compte d'autres symboles de fonctions utilisés par les protocoles et se rapprocher de la réalité, nous le ferons ici en prenant en compte des théories équationnelles. Une autre approche consiste à considérer qu'il est toujours possible de déchiffrer un message, même si celui-ci n'est pas un chiffré. Dans le modèle de Dolev-Yao originel ceci n'est pas possible. Par exemple, dans leurs travaux, décrits précédemment, F. Jacquemard et S. Delaune [DJ04] ajoutent au modèle la notion de destructeurs explicites. Il est alors désormais possible, dans ce modèle, de prendre en compte que n'importe quel message peut être déchiffré, même si ce n'est pas un message chiffré. Dans ce modèle la règle de déduction associée au déchiffrement est légèrement modifié de telle sorte que n'importe quelle hypothèse filtre (« match ») avec cette règle. Ainsi cette règle peut s'appliquer à n'importe quel terme. La règle (D) transformée devient la règle (D_{ex}) suivante :

$$(D_{ex})\frac{T \vdash u \qquad T \vdash k}{T \vdash dec(u,k)}$$

Pour respecter le fait que le déchiffrement d'un message m chiffré redonne le message initial m, nous devons modifier aussi la règle de chiffrement (C) par la règle (C_{ex}) de la manière suivante :

$$(C_{ex})\frac{T \vdash u \qquad T \vdash k}{T \vdash enc(u,k)}$$

Il faut alors ajouter l'équation $dec(enc(m,k),k) = m$ pour retrouver le message originel.

3.2 Le modèle de Dolev-Yao étendu avec des propriétés algébriques.

Nous utilisons la présentation usuelle du système de Dolev-Yao étendu qui prend en compte certaines propriétés algébriques représentées par une théorie équationnelle. Pour cela, nous considérons une nouvelle règle (Eq) pour passer d'un terme équivalent à un autre en appliquant la théorie équationnelle. Ensuite dans la section 3.2.2 page ci-contre, nous présentons un autre modèle pour des théories équationnelles qui sont représentées par un système de réécriture convergent et terminant. Nous montrons alors que ces deux systèmes de preuves sont équivalents. Ainsi nous pouvons travailler sur les formes normales des termes modulo la théorie équationnelle. Enfin, nous décrivons les différentes théories équationnelles que nous considérerons dans la suite de la thèse.

3.2.1 Le modèle de Dolev-Yao étendu par une théorie équationnelle.

Soit E une théorie équationnelle sur la signature finie Σ, où Σ partitionnée comme suit :

$$\Sigma = \{\langle \cdot, \cdot \rangle, \{\cdot\}_{\cdot}, \oplus\} \uplus \Sigma^{-}$$

La signature Σ est constituée de l'opérateur de paire $\langle \cdot, \cdot \rangle$, le chiffrement $\{\cdot\}_{\cdot}$, et du symbole de fonction $\oplus$, et d'autres symboles de fonction libres Σ^{-}. Nous utilisons dans le système de déduction des séquents de la forme $T \vdash_E t$, où T est un ensemble fini tel que $T \subseteq \mathcal{T}(\Sigma)$ et $t \in \mathcal{T}(\Sigma)$. Un séquent $T \vdash_E t$ signifie que l'intrus peut en fonction des ses capacités déduire le terme t de sa connaissance initiale T en se servant d'une théorie équationnelle E.

Définition 3 ((Σ, E)-séquent) *Un* (Σ, E)-séquent *est une expression de la forme* $T \vdash_E u$ *où* E *est une théorie équationnelle sur la signature* Σ, T *est un ensemble fini tel que* $T \subseteq \mathcal{T}(\Sigma)$, *et* $u \in \mathcal{T}(\Sigma)$.

Avec ces séquents, nous introduisons une nouvelle notion de preuve qui prend en compte la théorie équationnelle et un ensemble de règles pour engendrer de nouveaux symboles.

Définition 4 (Preuve) *Une* preuve *d'un* (Σ, E)*-séquent* $T \vdash_E u$ *est un arbre tel que :*
- *chaque feuille est étiquetée par une expression de la forme* $v \in T$*, et chaque nœud non feuille est étiqueté par un* (Σ, E)*-séquent.*
- *chaque nœud étiqueté par un séquent* $T \vdash_E v$ *a* n *fils étiqueté par* $T \vdash_E s_1, \ldots, T \vdash_E s_n$ *tel qu'il existe une instance d'une règle d'inférence qui a pour conclusion* $T \vdash_E v$ *et comme hypothèses* $T \vdash_E s_1, \ldots, T \vdash_E s_n$*.*
- *la racine de l'arbre de preuve est étiqueté par* $T \vdash_E u$*.*

Nous affaiblissons l'hypothèse de chiffrement parfait en donnant à l'intrus la capacités de raisonner avec la théorie équationnelle E. Pour cela nous considérons le système de déduction de la figure 3.1 page 47, dans lequel nous avons introduit une nouvelle règle (Eq) qui permet de prendre en compte explicitement la théorie équationnelle.

$$
\begin{array}{ll}
(A)\dfrac{u \in T}{T \vdash_E u} & (UL)\dfrac{T \vdash_E \langle u, v\rangle}{T \vdash_E u} \\[2ex]
(P)\dfrac{T \vdash_E u \qquad T \vdash_E v}{T \vdash_E \langle u, v\rangle} & (UR)\dfrac{T \vdash_E \langle u, v\rangle}{T \vdash_E v} \\[2ex]
(C)\dfrac{T \vdash_E u \qquad T \vdash_E v}{T \vdash_E \{u\}_v} & (D)\dfrac{T \vdash_E \{u\}_v \qquad T \vdash_E v}{T \vdash_E u} \\[2ex]
(Eq)\dfrac{T \vdash_E u \qquad u =_E v}{T \vdash_E v} & (F)\dfrac{T \vdash_E u_1 \quad \ldots \quad T \vdash_E u_n}{T \vdash_E f(u_1, \ldots, u_n)}
\end{array}
$$

FIG. 3.2 – Le modèle de preuve de Dolev-Yao étendu à une théorie équationnelle (E).

3.2.2 Le modèle Dolev-Yao étendu par réécriture.

Ce modèle n'est pas approprié pour un raisonnement automatique sur les preuves car la théorie équationnelle E peut intervenir à tout moment dans la preuve. Pour éviter cela nous coupons la théorie équationnelle en deux parties : une première partie S toujours présente dans les preuves (dans notre cas l'associativité et la commutativité), et une seconde partie que nous modélisons par un système de réécriture R.

Nous supposons que le système de réécriture R est confluent et termine modulo la théorie S (nous le montrerons dans la suite pour les théories qui nous intéressent). Nous avons besoin de définir une notion de forme normale associée au système de réécriture R modulo la théorie équationnelle S.

Définition 5 (Terme en forme normale) *Soit* (R, S) *une présentation sous forme de réécriture de la théorie équationnelle* E*. Un terme* t *est en* forme normale *s'il n'y a pas de terme* s *tel que* $t \rightarrow_{R/S} s$ *(*t *se réduit en* s *par une règle du système de réécriture* R *modulo* S*). Si* $t \rightarrow^* s$ *et* s *est en forme normale alors nous dirons que* s *est la forme normale de* t*, dénoté* $s = t\downarrow$*.*

Remarque les formes normales sont uniques à classe d'équivalence près.

Exemple 3 *Si nous avons la règle de réécriture suivante* $f(x \oplus y) \rightarrow_R f(x) \oplus f(y)$ *et* S *est la théorie associative et commutative associée au symbole* $\oplus$*, alors la forme normale de* $f((a \oplus b) \oplus c)$ *est* $f(a) \oplus f(b) \oplus f(c)$.

Les formes normales ont les propriétés suivantes :
- $\forall u, v : u =_E v \Rightarrow u\downarrow =_S v\downarrow$
- $\forall u : u =_E u\downarrow$

Définition 6 (Présentation sous forme de réécriture d'une théorie équationnelle) *Soient* E *et* S *une théorie équationnelle sur la signature* Σ*, et* R *un système de réécriture sur les* Σ*-termes.* (R, S) *est une* présentation sous forme de réécriture *de* E *si et seulement si*
- *R est localement confluent modulo S.*
- *R termine modulo S.*
- *Pour tous les termes clos Σ-termes u, v : $u =_E v$ si et seulement si $u\downarrow =_S v\downarrow$.où $t\downarrow$ dénote la forme normale de t pour le système de réécriture R modulo S.*

Nous considérons maintenant une extension du modèle de Dolev-Yao de la section 3.2.1 page 48, dans laquelle nous travaillons sur les formes normales modulo S à chaque étape de la preuve. L'idée est que l'équivalence modulo S est facilement décidable. Ainsi nous pouvons ôter la règle (Eq) et travailler seulement avec des classes d'équivalence modulo S. Nous considérons ici l'opérateur $\oplus$ qui est associatif et commutatif (AC).

Nous montrons maintenant l'équivalence entre les deux modèles introduits. Nous enlevons la règle (Eq) et considérons le nouveau système de déduction $\vdash$ présenté dans la figure 3.2 page précédente en utilisant les formes normales.

$$(A)\frac{u \in T}{T \vdash u\downarrow_{R/AC}} \qquad (UL)\frac{T \vdash r}{T \vdash u\downarrow_{R/AC}} \quad \textit{if } \langle u, v\rangle = r$$

$$(P)\frac{T \vdash u \qquad T \vdash v}{T \vdash \langle u, v\rangle\downarrow_{R/AC}} \qquad (UR)\frac{T \vdash r}{T \vdash v\downarrow_{R/AC}} \quad \textit{if } \langle u, v\rangle = r$$

$$(C)\frac{T \vdash u \qquad T \vdash v}{T \vdash \{u\}_v\downarrow_{R/AC}} \qquad (D)\frac{T \vdash r \qquad T \vdash v}{T \vdash u\downarrow_{R/AC}} \quad \begin{array}{l}\textit{if } r = \{u\}_v \\ \textit{and } u\downarrow_{R/AC} = u\end{array}$$

$$(F)\frac{T \vdash u_1\downarrow_{R/AC} \qquad \ldots \qquad T \vdash u_n\downarrow_{R/AC}}{T \vdash f(u_1, \ldots, u_n)\downarrow_{R/AC}}$$

FIG. 3.3 – Modèle de preuve de Dolev-Yao avec les formes normales issues d'un système de réécriture R modulo AC.

Théorème 1 *Soit* (R, S) *une présentation sous forme de réécriture de la théorie équationnelle* E*,* $T \subseteq \mathcal{T}(\Sigma)$*, et* $T \in \mathcal{T}(\Sigma)$*. Si la théorie équationnelle à les propriétés suivantes* $(\{u\}_v)\downarrow = \{u\downarrow\}_{v\downarrow}$*,* $(\langle u, v\rangle)\downarrow = \langle u\downarrow, v\downarrow\rangle$ *alors nous avons*

$$T \vdash_E u \textit{ si et seulement si } T \vdash u\downarrow$$

Preuve :

Les propriétés $(\{u\}_v)\downarrow = \{u\downarrow\}_{v\downarrow}$ et $(\langle u, v\rangle)\downarrow = \langle u\downarrow, v\downarrow\rangle$ nous assurent que les symboles de chiffrement et de paire ne dissparaissent pas avec la théorie équationnelle dans le système de réécriture.

Soit une preuve de $T \vdash u\!\downarrow$ nous pouvons facilement trouver une preuve de $T \vdash_E u$ en insérant des étapes par la règle (Eq).

Réciproquement, nous transformons une preuve qui utilise $\vdash_E$ en une preuve utilisant $\vdash$ par la transformation donnée dans la figure 3.4. Cette transformation ne change pas les feuilles de la preuve. Nous montrons par induction que s'il y a une preuve de $T \vdash_E u$ alors la transformation aboutit en une preuve de $T \vdash u\!\downarrow$.

$$(Eq)\dfrac{(R_1)\dfrac{P_1}{T \vdash_E u_1} \qquad u =_E v}{T \vdash_E v} \qquad \Rightarrow \qquad (R)\dfrac{P_1}{T \vdash v\!\downarrow}$$

$$(R)\dfrac{(R_1)\dfrac{P_1}{T \vdash_E u_1} \qquad \dots \qquad (R_1)\dfrac{P_1}{T \vdash_E u_1}}{T \vdash_E v} \quad \Rightarrow \quad (R)\dfrac{(R_1)\dfrac{P_1}{T \vdash_E u_1} \qquad \dots \qquad (R_1)\dfrac{P_1}{T \vdash_E u_1}}{T \vdash v\!\downarrow}$$

FIG. 3.4 – Transformations d'une preuve de $T \vdash_E u$ en une preuve de $T \vdash u\!\downarrow$.

Nous regardons les différents pour la dernière règle du système :

- (A) : Par définition de cette règle, le résultat est forcément en forme normale.
- $(Eq(E))$: Comme (R, S) est une présentation sous forme de réécriture de la théorie E nous avons $u\!\downarrow = v\!\downarrow$ (modulo S) nous obtenons bien une preuve de $T \vdash u\!\downarrow$
- (P), (C) ou (F) : Par hypothèse d'induction sur toutes les hypothèses de la règle et avec le fait que $f(u_1, \dots, u_n)\!\downarrow = f(u_1\!\downarrow, \dots, u_n\!\downarrow)\!\downarrow$ et les propriétés $(\{u\}_v)\!\downarrow = \{u\!\downarrow\}_{v\downarrow}$, $(\langle u, v\rangle)\!\downarrow = \langle u\!\downarrow, v\!\downarrow\rangle$ nous concluons.
- (D), par induction $T \vdash \{u\}_v\!\downarrow$ et $T \vdash v\!\downarrow$, avec $\{u\}_{v\downarrow} = \{u\}_{v\downarrow}\!\downarrow$ et la règle (D) nous obtenons $T \vdash u\!\downarrow$, par conséquent une preuve utilisant $\vdash$.
- (UL) or (UR) par induction nous obtenons le résultat.

□

Dans la suite, nous montrons qu'il existe bien une présentation sous forme de réécriture pour toutes les théories équationnelles que nous considérerons dans la partie II page 45. Ainsi, dans la suite, nous travaillerons qu'avec des formes normales.

3.2.3 Différentes théories équationnelles considérées.

Nous étudions différentes théories équationnelles interagissant avec un opérateur associatif et commutatif $\oplus$. Pour modéliser cela nous introduisons la règle (GX) dans le modèle de Dolev-Yao introduit précédemment dans la figure 3.5 page suivante, cela permet à un intrus s'il connaît un nombre fini de termes $u_1, \dots, u_n$ de construire l'application du symbole à cet ensemble de termes (ceci car le symbole $\oplus$ est associatif).

Nous étudierons, dans les sections futures, les théories équationnelles présentées dans la section 2.5.3 page 41 : ACh, ACUNh, AGh, $\mathsf{ACUN}\{.\}_.$, $\mathsf{AG}\{.\}_.$, où $\mathsf{ACUN}\{.\}_.$ dénote la théorie équationnelle dans laquelle le chiffrement distribue sur le « ou exclusif » et $\mathsf{AG}\{.\}_.$ la théorie dans laquelle le chiffrement distribue sur l'opérateur des groupes abéliens. Nous considérerons aussi les théories $\mathsf{ACUN}\{.\}_.$ et $\mathsf{AG}\{.\}_.$ où le chiffrement est commutatif. Pour chacune d'entre elles, nous avons associé un système de réécriture et montré la confluence et la terminaison de ce système [LLT04], en utilisant l'outil Cime [CM96]. Nous le montrons en détail uniquement ici pour le cas de la théorie ACUNh.

$$(A)\frac{u \in T}{T \vdash u\downarrow_{R/AC}} \qquad (UL)\frac{T \vdash r}{T \vdash u\downarrow_{R/AC}} \quad \textit{if } \langle u, v\rangle = r$$

$$(P)\frac{T \vdash u \qquad T \vdash v}{T \vdash \langle u, v\rangle\downarrow_{R/AC}} \qquad (UR)\frac{T \vdash r}{T \vdash v\downarrow_{R/AC}} \quad \textit{if } \langle u, v\rangle = r$$

$$(C)\frac{T \vdash u \qquad T \vdash v}{T \vdash \{u\}_v\downarrow_{R/AC}} \qquad (D)\frac{T \vdash r \qquad T \vdash v}{T \vdash u\downarrow_{R/AC}} \quad \begin{array}{l}\textit{if } r = \{u\}_v \\ \textit{and } u\downarrow_{R/AC} = u\end{array}$$

$$(GX)\frac{T \vdash u_1 \qquad \ldots \qquad T \vdash u_n}{T \vdash u_1 \oplus \ldots \oplus u_n\downarrow_{R/AC}} \qquad (F)\frac{T \vdash u_1\downarrow_{R/AC} \qquad \ldots \qquad T \vdash u_n\downarrow_{R/AC}}{T \vdash f(u_1, \ldots, u_n)\downarrow_{R/AC}}$$

FIG. 3.5 – Modèle de preuve de Dolev-Yao avec les formes normales issues d'un système de réécriture R modulo AC prenant en compte un opérateur $\oplus$ associatif et commutatif.

Rappelons d'abord les axiomes associés à la théorie équationnelle ACUNh, ensuite orientons ces axiomes pour obtenir un système de réécriture. Nous ajoutons certaines règles de réécriture nécessaire pour former un nouveau système confluent. Enfin, nous montrons la terminaison de ce nouveau système.

Théorie équationnelle. Nous considérons les symboles $\oplus, 0, h$ avec les axiomes suivants :

1. $(x \oplus y) \oplus z = x \oplus (y \oplus z)$ (Associativité)
2. $x \oplus y = y \oplus x$ (Commutativité)
3. $0 \oplus x = x$ (Unité)
4. $x \oplus x = 0$ (Nilpotence)
5. $h(x \oplus y) = h(x) \oplus h(y)$ (Homomorphisme)

Système de réécriture associé modulo AC. Nous orientons les équations des axiomes de la gauche vers la droite et obtenons le système de réécriture suivant modulo AC.

1. $0 \oplus x \to x$
2. $x \oplus x \to 0$
3. $h(x \oplus y) \to h(x) \oplus h(y)$

Confluence locale. Nous calculons les paires critiques de ce système :
$h(x) \oplus h(0) \leftarrow h(x \oplus 0) \to h(x)$
$h(x) \oplus h(x) \leftarrow h(x \oplus x) \to h(0)$

Grâce aux propriétés du « ou exclusif » et à la propriété suivante $h(x \oplus y) = h(x) \oplus h(y)$ nous déduisons $h(0) = h(x \oplus x) = h(x) \oplus h(x) = 0$ et donc $h(0) = 0$. En introduisant la nouvelle règle de réécriture : $h(0) \to 0$, nous obtenons que le nouveau système est confluent car il ne contient plus de paires critiques.

Terminaison. Nous donnons une interprétation par des polynômes pour montrer la terminaison de ce système, en associant à la :
- constante : $[0] = 0$
- variable : $[x] = x$
- fonction h : $[h(x)] = 2x + 1$
- fonction du « ou exclusif » : $[x \oplus y] = x + y + 1$

Nous montrons que toutes les règles du système modulo AC sont décroissantes, ce qui prouve la terminaison du système de réécriture.

- $0 \oplus x \to x$ implique : $[0 \oplus x] = x + 1 > x = [x]$
- $x \oplus x \to 0$ implique : $[x \oplus x] = 2x + 1 > 0 = [0]$
- $h(x \oplus y) \to h(x) \oplus h(y)$ implique : $[h(x \oplus y)] = 2x + 2y + 3 > 2x + 2y + 2 = [h(x) \oplus h(y)]$
- $h(0) \to 0$ implique : $[h(0)] = 1 > 0 = [0]$

Exemple 4 *Considérons la théorie équationnelle du « ou exclusif » pour E, ainsi à partir de $T = \{a \oplus b \oplus c, b \oplus d\}$ nous pouvons déduire le terme $a \oplus c \oplus d$ en utilisant le système de la figure 3.2 page 49 étendu par la règle (GX).*

$$(Eq)\dfrac{(GX)\dfrac{(A)\dfrac{a \oplus b \oplus c \in T}{T \vdash_E a \oplus b \oplus c} \qquad (A)\dfrac{b \oplus d \in T}{T \vdash_E b \oplus d}}{T \vdash_E a \oplus c \oplus d} \qquad a \oplus b \oplus c \oplus b \oplus d =_E a \oplus c \oplus d}{T \vdash_E a \oplus c \oplus d}$$

Comme nous avons montré que la théorie ACUNh *peut être représentée par un système de réécriture convergent modulo AC, nous pouvons passer au système de Dolev-Yao utilisant les formes normales. En remarquant que si $t[.]$ est un contexte et u un terme clos, alors $t[u\downarrow]\downarrow =_S t[u]\downarrow$, et en particulier $(u_1 \oplus \ldots \oplus u_n)\downarrow =_S (u_1\downarrow \oplus \ldots \oplus u_n\downarrow)\downarrow$, nous obtenons la preuve suivante :*

$$(GX)\dfrac{(A)\dfrac{a \oplus b \oplus c \in T}{T \vdash a \oplus b \oplus c} \qquad (A)\dfrac{b \oplus d \in T}{T \vdash b \oplus d}}{T \vdash a \oplus c \oplus d}$$

Chapitre 4

Localité simple.

« I have had my results for a long time :
but I do not yet know how I am to arrive at them. »
Carl Friedrich Gauss.

Sommaire

Maintenant, nous présentons l'approche usuelle pour montrer le problème de déduction de l'intrus basée sur l'idée localité introduite par McAllester. Nous appliquons cette technique pour démontrer un premier résultat simple de localité. Nous montrons comment ce résultat permet de redémontrer le problème de déduction de l'intrus pour le modèle de Dolev-Yao standard. Dans le chapitre suivant nous appliquerons ce résultat de localité simple afin de simplifier les preuves pour les théories équationnelles homomorphiques.

4.1 Approche par localité de McAllester.

Notre approche pour prouver le problème de déduction de l'intrus est basée sur l'approche plus générale, développée par McAllester [McA93] dans les années 90.

4.1.1 L'idée de McAllester : Une construction par saturation.

Il considère des systèmes de déductions représentés par un ensemble fini de clauses de Horn. Il propose un algorithme polynômial pour décider si un terme w est déductible à partir d'un ensemble fini de termes T pour un système de preuves possédant la propriété de *localité*. L'ensemble T est appelé *ensemble des hypothèses* et le terme w est lui appelé *le but* ou *la conclusion* ou *la racine* de la preuve. Nous notons $T \vdash w$ si le terme w est déductible à partir de l'ensemble de termes T. Un système de déduction est *local*, ou possède la propriété de localité, si pour toute preuve de $T \vdash w$ dans ce système de déduction il existe une *preuve locale* de $T \vdash w$. Une *preuve locale* est une preuve dont tous les nœuds sont contenus dans l'ensemble des sous-termes syntaxiques de $T \cup \{w\}$. Nous noterons pour plus de simplicité l'ensemble $T \cup \{w\}$ par T, w dans la suite du document.

Si le système de déduction est local, l'idée de McAllester est de construire la preuve de $T \vdash w$ par saturation restreinte aux termes de l'ensemble des sous-termes syntaxiques de $T \cup \{w\}$. Cette construction de la preuve de $T \vdash w$ est possible car le système de preuve est local *i.e.* il existe une preuve où chaque nœud de la preuve est dans l'ensemble des sous-termes syntaxiques de T, w. Comme l'ensemble des sous-termes syntaxiques est fini et calculable en temps polynômial, il obtient alors un algorithme efficace pour décider si un terme est déductible à partir d'un ensemble de termes donnés.

4.1.2 Notre extension de l'algorithme de McAllester.

L'approche de McAllester suppose deux restrictions importantes :
- La localité est définie uniquement pour la notion de sous-termes syntaxiques.
- Le système de règles de déduction doit être fini.

Nous souhaitons appliquer cette technique pour l'étude des théories équationnelles avec un symbole d'homomorphisme. Mais il n'est pas toujours possible de montrer un résultat de localité en ne considérant que des sous-termes syntaxiques. Nous généralisons donc la notion de localité en *S-localité* où S sera une fonction à définir pour chaque théorie équationnelle.

Définition 7 (Preuve S-locale) *Soit S fonction d'un ensemble de termes vers un ensemble de termes. Une preuve P de $T \vdash w$ est* S-locale *si tous les nœuds sont étiquetés par $T \vdash v$ où $v \in S(T, w)$. Un système de preuve est* S-local *s'il existe une preuve de $T \vdash w$ dans P alors il existe une preuve S-locale de $T \vdash w$ dans P.*

Le résultat de McAllester utilise donc un système de preuve S-local où la fonction S est la fonction de sous-termes syntaxiques.

Nous étudions des théories qui sont associatives et commutatives, si nous modélisons la règle de somme pour les groupes abéliens par une règle de déduction binaire alors, comme l'ont déjà remarqué H. Common *et alii* [CLS03], en général le nombre de sous-termes syntaxiques modulo AC est exponentiel. Une solution pour pallier à ce problème est de représenter par une seule règle de déduction n-aire la règle qui permet de faire l'addition, comme le montre notre extension du système de Dolev-Yao définie dans la figure 3.2 page 49. Malheureusement cette modélisation implique qu'il y a alors une infinité de règles à considérer. Nous modifions donc légèrement l'algorithme de McAllester originel en rajoutant une condition dans la construction par saturation. Nous présentons dans l'algorithme 1 page ci-contre, cette légère extension selon l'idée de McAllester.

Ce nouveau algorithme permet de construire pas à pas toutes les preuves locales à partir d'un ensemble de termes T et d'un terme but w, selon l'idée originelle de McAllester. Cet algorithme suppose que le système de preuve est S-local, où la fonction S calcule un ensemble des termes à

Algorithme 1 : Généralisation de l'algorithme de McAllester pour calculer l'existence d'une preuve S-locale de $T \vdash w$.

```
Données : T, w
Sub ← S(T, w);
répéter
    T_p ← T;
    pour chaque t ∈ Sub faire
        si T ⊢^{=1} t alors T ← T ∪ {t};
    fin
jusqu'à T_p = T;
retourner w ∈ T;
```

partir d'un ensemble de termes. De plus, il ne suppose pas qu'un nombre de règles du système de déduction soit borné. Le symbole $\vdash^{=1}$ signifie qu'une seule règle du système de déduction est employée pour déduire le but à partir de l'ensemble des hypothèses considérées, nous dirons alors que le terme but est *déductible en une étape* à partir de l'ensemble des hypothèses. Dans ce nouvel algorithme, il faut maintenant décider si un terme est déductible en une étape.

Le résultat de localité étendu généralise le résultat de McAllester. et s'énonce dans le théorème 2.

Théorème 2 (Localité) *Soit S une fonction d'un ensemble de termes vers un ensemble de termes, et P un système de preuve. Soit T un ensemble de termes hypothèses, w un terme but et n la cardinalité de $T \cup \{w\}$, dénoté par $|T, w|$. Si :*

1. *la déductibilité en une étape de $T \vdash^{=1} u$ dans P est décidable en temps $g(|T, u|)$ pour tout terme u et pour tout ensemble de termes T,*
2. *l'ensemble $S(T, w)$ est calculable en temps $f(n)$,*
3. *le système de preuve P est S-local,*

alors savoir si $T \vdash w$ est déductible avec le système de preuve P est décidable en temps $f(n) + f(n) * f(n) * g(f(n))$ (non déterministe si le critère (1) est non déterministe).

Preuve : Grâce à la S-localité du système de preuve P, savoir si $T \vdash w$ est prouvable dans P est équivalent à l'existence d'une preuve de S-locale de $T \vdash w$. En utilisant l'algorithme 1 nous pouvons construire une preuve S-locale de $T \vdash w$. Le calcul de *Sub* se fait en temps $f(n)$. En conséquence la cardinalité de *Sub* est bornée par $f(n)$, ainsi le nombre d'itérations des boucles de l'algorithme est donc fini. Comme le nombre d'éléments de T est borné par $f(n)$ le test s'effectue en temps $g(f(n))$, par une énumération de tous les cas possibles. Nous concluons qu'il est décidable en temps $f(n) + f(n) * f(n) * g(f(n))$ de savoir si $T \vdash w$ est déductible dans le système de preuve P. □

Pour montrer le problème de déduction de l'intrus en présence d'une théorie équationnelle dans le système de Dolev-Yao étendu, en utilisant l'approche de McAllester généralisée, nous devons pour chaque théorie équationnelle prouver les points suivants :

(i) Définir la fonction S de sous-termes appropriée avec une complexité en temps de $f(n)$.

(ii) Montrer la S-localité pour ce système de preuve P.

(iii) Montrer que la déductibilité en une étape pour le système de preuve considéré s'effectue en temps $g(n)$.

Nous suivons donc cette démarche pour chacune des théories que nous analysons dans la suite de la thèse. Pour une plus grande clarté, nous considérons uniquement un chiffrement symétrique. Le cas du chiffrement asymétrique se généralise en modifiant le système de Dolev-Yao pour symboliser les clefs inverses et en suivant les mêmes techniques, nous ne le présenterons pas dans ce document en détail, excepté pour un cas particulier dans la section 5.3.2 page 106.

4.2 Localité simple pour le modèle de Dolev-Yao étendu.

Nous définissons en premier la notion de sous-termes syntaxiques et quelques critères sur la forme des preuves. Ces éléments nous permettent d'effectuer une étude précise des règles de construction et destruction de la paire, ainsi nous démontrons un premier lemme simple de localité pour le système de Dolev-Yao de la figure 4.1, où nous pouvons construire un symbole homomorphique h et le symbole associatif commutatif $\oplus$ avec la règle (GX).

$$
\begin{array}{ll}
(A)\dfrac{u \in T}{T \vdash_E u} & (UL)\dfrac{T \vdash_E \langle u, v\rangle}{T \vdash_E u} \\
(P)\dfrac{T \vdash_E u \qquad T \vdash_E v}{T \vdash_E \langle u, v\rangle} & (UR)\dfrac{T \vdash_E \langle u, v\rangle}{T \vdash_E v} \\
(C)\dfrac{T \vdash_E u \qquad T \vdash_E v}{T \vdash_E \{u\}_v} & (D)\dfrac{T \vdash_E \{u\}_v \qquad T \vdash_E v}{T \vdash_E u} \\
(GX)\dfrac{T \vdash_E u_1 \qquad \ldots \qquad T \vdash_E u_n}{T \vdash_E u_1 \oplus \ldots \oplus u_n} & (h)\dfrac{T \vdash_E u}{T \vdash_E h(u)}
\end{array}
$$

FIG. 4.1 – Modèle de preuve de Dolev-Yao avec les formes normales issues d'un système de réécriture R modulo AC prenant en compte un opérateur $\oplus$ associatif et commutatif et un symbole de fonction h.

4.2.1 Définitions.

Nous dénotons par $na := a + a + \ldots + a$, n fois, lorsque le symbole $+$ est le symbole additif des groupes abéliens. Nous présentons des définitions dans le cadre des groupes abéliens, elles restent valables et sont facilement transposables au « ou exclusif ».

4.2.1.1 Sous-termes syntaxiques.

Nous caractérisons d'abord les termes en fonction du dernier symbole qui les constituent, puis donnons la définition de sous-termes syntaxiques.

Définition 8 (Terme en tête avec $+$ et terme en tête avec $-$) *Soit u un terme en forme normale, u est* en tête avec $+$ *si u est de la forme $u_1 + \ldots + u_n$ avec $n > 1$. Sinon u* n'est pas en tête avec $+$. *De plus, u est* en tête avec $-$ *si u est de la forme $-v$ où v est un terme non en tête avec $+$. Sinon u* n'est pas en tête avec $-$.

Exemple 5 *Le terme $t_1 = -2u + 3\langle v, w\rangle$ est en tête avec $+$ et n'est pas en tête avec $-$. Par contre le terme $t_2 = \{3v + w\}_k$ est en tête ni avec $+$ ni avec $-$. Selon nos notations le terme $-3u = (-u) + (-u) + (-u)$ est en tête avec $+$ et le terme $-a$ est en tête avec $-$ mais pas avec $+$.*

Définition 9 ($S(t)$) *L'ensemble des* sous-termes syntaxiques *d'un terme t est défini comme le plus petit ensemble $S(t)$ tel que :*

- $t \in S(t)$.
- *Si* $\langle u, v\rangle \in S(t)$ *alors* $u, v \in S(t)$.
- *Si* $\{u\}_v \in S(t)$ *alors* $u, v \in S(t)$.
- *Si* $u = u_1 + \ldots + u_n \in S(t)$ *et* u_i *n'est pas en tête ni avec* $+$ *ni avec* $-$ *alors* $u_i \subseteq S(t)$.

Cette notion de sous-termes S s'étend de manière naturelle aux ensembles de termes T en forme normale par $S(T) := \bigcup_{t\in T} S(t)$, et $S(T)$ se calcule en temps polynômial.

Exemple 6 *Pour le terme* $t = 2a + \langle 3b + c, d\rangle$ *nous obtenons l'ensemble des sous-termes suivant :*

$$S(t) = \{t, a, \langle 3b + c, d\rangle, 3b + c, b, c, d\}$$

Remarquons que les termes $2a$ *et* $3b$ *ne sont pas sous-termes syntaxiques de* t.

Proposition 1 *Soit* A *et* B *deux termes en forme normale, la fonction de sous-terme syntaxique* S *possède les propriétés suivantes :*
- $S(A \cup B) = S(A) \cup S(B)$.
- S *est idempotent :* $S(S(A)) = S(A)$.
- S *est monotone : si* $A \subseteq B$ *alors* $S(A) \subseteq S(B)$.
- S *est transitive.*

Preuve : Ces propriétés sont conséquences immédiates de la Définition 9 page précédente de sous-terme syntaxique. $\square$

4.2.1.2 Preuves.

Nous définissons plusieurs notions de preuves comme les preuves simples ou aplatie.

Définition 10 (Sous-preuve) *Soit* P *une preuve de* $T \vdash w$.
- *Une* sous-preuve P' *de* P *est un sous arbre de* P.
- *La* taille d'une preuve P, *notée* $|P|$, *est le nombre de nœuds de* P.
- *Une preuve* P *de* $T \vdash w$ *est* minimale *si pour toute preuve* P' *de* $T \vdash w$ *nous avons* $|P| \leq |P'|$.
- *Une preuve* P *est* simple *si chaque nœud* $T \vdash v$ *n'apparaît au plus qu'une seule fois dans chaque branche et qu'un nœud* $T \vdash v$ *ne soit présent au plus une seule fois dans les hypothèses de la règle* (GX).
- *Une preuve* P *est* aplatie *s'il n'y a jamais deux applications consécutives de la règle* (GX).

Puisque deux applications successives de la règle (GX) peuvent être regroupées en une seule règle (GX), comme le montre la figure 4.2, alors toute preuve peut être transformée en une preuve aplatie.

$$(GX)\frac{(GX)\dfrac{T \vdash x_1 \quad \ldots \quad T \vdash x_n}{T \vdash \alpha_1 x_1 + \ldots + \alpha_n x_n} \quad T \vdash y_1 \quad \ldots \quad T \vdash y_m}{T \vdash \beta(\alpha_1 x_1 + \ldots + \alpha_n x_n) + \beta_1 y_1 + \ldots + \beta_m y_m}$$
$$\Downarrow$$
$$(GX)\frac{T \vdash x_1 \quad \ldots \quad T \vdash x_n \quad T \vdash y_1 \quad \ldots \quad T \vdash y_m}{T \vdash \beta\alpha_1 x_1 + \ldots + \beta\alpha_n x_n + \beta_1 y_1 + \ldots + \beta_m y_m}$$

FIG. 4.2 – Fusion de deux règles (GX) en une seule.

Pour obtenir une preuve simple nous coupons simplement dans l'arbre de preuve la branche entre les deux occurrences du même nœud. Cette simplification fait évidemment diminuer la taille de la preuve.

Proposition 2 *Soit P une preuve simple alors :*

1. *Il n'y a pas de règle (D_v) juste après une règle (C_v) dans P.*
2. *Il n'y a pas de règle (C_v) juste après la règle (D_v) dans P.*

Preuve : C'est une conséquence immédiate de la simplicité, sinon il y aurait deux fois le même nœud dans une branche de la preuve P. □

Proposition 3 *Soit P une preuve de $T \vdash w$.*
- *Si P est minimale alors P est aplatie.*
- *Si P est minimale alors P est simple.*

Preuve : Supposons que la preuve P de $T \vdash w$ ne soit pas aplatie, alors la preuve n'est pas minimale car nous pouvons construire une preuve plus petite en nombre de nœuds en fusionnant les deux applications successives de la règle (GX). De même, si une preuve P de $T \vdash w$ n'est pas simple, alors la preuve n'est pas minimale car nous pouvons construire une preuve plus petite en nombre de nœuds en coupant le sous-arbre de preuve entre les deux mêmes nœuds qui apparaissent dans la même branche de la preuve. □

Nous utiliserons implicitement ces deux propriétés dans la suite.

4.2.2 Analyse de la règle de paire et des règles de projection.

Nous démontrons deux lemmes techniques à propos des règles de construction et de destruction du symbole de paire. Le premier, le lemme 2 page suivante, montre que dans une preuve simple tous nœuds issus d'une des deux règles de projection (UR) ou (UL) sont des sous-termes de la connaissance initiale de l'intrus. Le second, le lemme 2 page ci-contre, démontre que pour une preuve minimale tout nœud issu d'une règle de construction de paire (P) est un sous-terme syntaxique de la conclusion de la preuve et de la connaissance initiale de l'intrus.

Les différentes variations de théories équationnelles n'influent pas sur ces deux lemmes, qui restent vrais dans chacune des théories équationnelles considérées dans cette thèse.

Lemme 1 *Soit P une preuve simple de la forme suivante :*

$$P = \left\{ (R)\frac{P_1 \quad \ldots \quad P_n}{T \vdash w} \right.$$

Si $T \vdash u$ n'apparaît dans aucune preuve $P_1, \ldots, P_n$ et $\langle u, v\rangle \in S(w)$ alors il y a au moins un i tel que $\langle u, v\rangle \in S(w_i)$, où la racine de P_i est soit $T \vdash w_i$, soit $w_i \in T$.

Preuve : Nous considérons tous les cas de figure possibles pour la dernière règle (R) de la preuve P :
- Si la dernière règle (R) est la règle (A) : le lemme est évidemment vrai.
- Si la dernière règle (R) est la règle (UL) ou (UR) : Nous avons une seule règle au-dessus de la dernière règle et $w_1 = \langle u_1, u_2\rangle$ où w est soit u_1 soit u_2. Nous concluons donc par hypothèse d'induction car $\langle u, v\rangle \in S(w) \subseteq S(w_1)$.

– Si la dernière règle (R) est la règle (D) : il n'y a qu'une seule règle au-dessus de la règle (R) et $w_1 = \{u_1\}_{u_2}$ où w est soit u_1 soir u_2. Nous concluons donc par hypothèse d'induction car $\{u\}_v \in S(w) \subseteq S(w_1)$.
– Si la dernière règle (R) est la règle (GX) : Par hypothèse $\langle u, v\rangle \in S(w)$ et $w = (w_1 \oplus \ldots \oplus w_n)\downarrow$. Par définition de sous-terme $\langle u, v\rangle \in \cup_i S(w_i)$, il existe donc un i tel que $\langle u, v\rangle \in S(w_i)$, car le terme $\langle u, v\rangle$ n'est pas en tête avec $\oplus$. Nous concluons alors par hypothèse de récurrence.
– Si la dernière règle (R) est la règle (P) : Or $T \vdash w$ ne peut pas apparaître dans P par simplicité de la preuve P. De plus nous savons que $w = \langle w_1, w_2\rangle \neq \langle u, v\rangle$. Par hypothèse $\langle u, v\rangle \in S(w)$, nous obtenons donc que $\langle u, v\rangle \in S(w_1) \cup S(w_2)$ et concluons par hypothèse d'induction.
– Si la dernière règle (R) est la règle (C) : Dans ce cas nous savons que $w = \{w_1\}_{w_2}$. Or par hypothèse $\langle u, v\rangle \in S(w)$ nous obtenons donc que $\langle u, v\rangle \in S(w_1) \cup S(w_2)$ et concluons par hypothèse d'induction.
– Si la dernière règle (R) est la règle (h) : Dans ce cas nous savons que $w = h(w_1)$. Or par hypothèse $\langle u, v\rangle \in S(w)$ nous obtenons donc que $\langle u, v\rangle \in S(w_1)$ et concluons par hypothèse d'induction.

$\square$

Lemme 2 *Soit P une preuve simple de $T \vdash u$. Si P est de la forme suivante :*

$$(UL)\dfrac{\dfrac{\vdots}{T \vdash \langle u, v\rangle}}{T \vdash u} \qquad (UR)\dfrac{\dfrac{\vdots}{T \vdash \langle v, u\rangle}}{T \vdash u}$$

alors $\langle u, v\rangle \in S(T)$.

Preuve : Nous supposons que la dernière règle est (UL), le cas de (UR) se résout de la même manière.

$$P = \begin{cases} (UL)\dfrac{\dfrac{P_1 \ldots P_n}{T \vdash \langle u, v\rangle}}{T \vdash u} \end{cases}$$

La preuve P est simple donc $T \vdash u$ n'apparaît pas dans les sous-preuves $P_1, \ldots, P_n$. Par conséquent, nous pouvons appliquer le lemme 1 page précédente à la preuve $\dfrac{P_1 \ldots P_n}{T \vdash \langle u, v\rangle}$. Soit $\langle u, v\rangle \in T$, soit il existe une sous-preuve P_i avec pour racine $T \vdash w$ telle que $\langle u, v\rangle \in S(w)$ et $T \vdash u$ n'apparaît pas dans P_i. Nous appliquons autant de fois que nécessaire le lemme 1 page ci-contre et finalement obtenons $\langle u, v\rangle \in T$. $\square$

Lemme 3 *Soit P une preuve minimale de $T \vdash w$. Si contient une application de la règle de construction de paire (P) de la forme suivante :*

$$(P)\dfrac{\dfrac{\vdots}{T \vdash u} \quad \dfrac{\vdots}{T \vdash v}}{T \vdash \langle u, v\rangle}$$

alors $\langle u, v\rangle \in S(T, w)$.

Preuve : Nous prouvons le résultat par induction structurelle sur la preuve P de $T \vdash w$. Nous regardons les différentes possibilités pour la dernière règle de la preuve P :

– La dernière règle est (A). Le résultat est immédiat.
– La dernière règle est (UR) ou (UL).

$$(UL)\frac{\dfrac{P_1}{T \vdash \langle w, v\rangle}}{T \vdash w}$$

Par hypothèse d'induction nous savons que tous les nœuds issus d'une règle (P) dans la sous-preuve P_1 sont dans $S(T, \langle w, v\rangle)$. Comme la preuve P est minimale, elle est alors simple (lemme 3 page 60), nous pouvons donc appliquer le lemme 1 page 60 et donc $\langle w, v\rangle \in S(T)$. Ainsi tous les nœuds issus d'une règle (P) sont dans $S(T) \subseteq S(T, w)$.

– La dernière règle est (C).

$$(C)\frac{\dfrac{P_1}{T \vdash u} \quad \dfrac{P_2}{T \vdash k}}{T \vdash \{u\}_k = w}$$

Par hypothèse d'induction nous savons que tous les nœuds issus d'une règle (P) dans la sous-preuve P_1 sont dans $S(T, u)$ et tous les nœuds issus d'une règle (P) dans la sous-preuve P_2 sont dans $S(T, k)$. Par définition de S $u \in S(\{u\}_k) = S(w)$ et $k \in S(\{u\}_k) = S(w)$, ainsi tous les nœuds issus d'une règle (P) sont dans $S(T) \subseteq S(T, w)$.

– La dernière règle est (h).

$$(h)\frac{\dfrac{P_1}{T \vdash u}}{T \vdash h(u) = w}$$

Par hypothèse d'induction nous savons que tous les nœuds issus d'une règle (P) dans la sous-preuve P_1 sont dans $S(T, u)$. Par définition de S, $u \in S(h(u)) = S(w)$, ainsi tous les nœuds issus d'une règle (P) sont dans $S(T) \subseteq S(T, w)$.

– La dernière règle est (P). Pour la dernière occurrence de la règle (P) qui génère le terme final w ainsi le résultat est évidemment dans $S(T, w)$. Le reste de la démonstration est similaire au cas de la règle (C) précédemment analysée.

$$(P)\frac{\dfrac{P_1}{T \vdash u} \quad \dfrac{P_2}{T \vdash v}}{T \vdash \langle u, v\rangle = w}$$

Par hypothèse d'induction nous savons que tous les nœuds issus d'une règle (P) dans la sous-preuve P_1 sont dans $S(T, u)$ et tous les nœuds issus d'une règle (P) dans la sous-preuve P_2 sont dans $S(T, v)$. Par définition de S $u \in S(\langle u, v\rangle) = S(w)$ et $v \in S(\langle u, v\rangle) = S(w)$, ainsi tous les nœuds issus d'une règle (P) sont dans $S(T) \subseteq S(T, w)$.

– La dernière règle est (GX).

$$(GX)\frac{\dfrac{P_1}{T \vdash u_1} \quad \cdots \quad \dfrac{P_n}{T \vdash u_n}}{T \vdash w = (\alpha_1 u_1 \oplus \ldots \oplus \alpha_n u_n)\downarrow}$$

Nous considérons les occurrences de la règle (P) qui engendrent le terme $v = \langle v_1, v_2\rangle$ dans une sous-preuve P_i. Par hypothèse d'induction, nous obtenons $v \in S(T, u_i)$. Supposons que $v \notin S(T, w)$, ce qui implique que $v \notin S(T)$, et par conséquent $v \in S(u_i)$.

Les occurrences de v sont donc « éliminées » dans la « somme » faite par la règle (GX), ce qui implique qu'il existe un terme u_j tel que $v \in S(u_j)$ avec $j \neq i$. Nous construisons alors une preuve plus petite de $T \vdash w$. Nous analysons les deux sous-preuves P_i et P_j. Nous cherchons le chemin maximal à partir du nœud u_i, resp. u_j où tous les nœuds possèdent v comme sous-termes. Soit un des chemins finit sur une règle (A), dans ce cas nous concluons que $v \in S(T)$; ce qui contredit l'hypothèse $v \notin S(T, w)$. Soit les deux chemins finissent sur une application de la règle (P), alors nous remplaçons cette application de la règle par la sous-preuve qui se termine par $T \vdash v_1$, nous coupons donc toute la sous-preuve de v_2, et dans tous les nœuds des deux chemins, nous remplaçons le sous-terme v par v_1. Nous obtenons bien une preuve plus courte, ce qui contredit la minimalité de la preuve.

– La dernière règle est (D).

$$(D)\frac{\dfrac{P_1}{T \vdash \{w\}_k} \quad \dfrac{P_2}{T \vdash k}}{T \vdash w}$$

Nous raisonnons de même que dans le cas précédent pour la règle (GX).
Nous considérons les occurrences de la règle (P) qui engendrent le terme $v = \langle v_1, v_2 \rangle$ dans une sous-preuve P_1 et P_2. Par hypothèse d'induction, nous obtenons $v \in S(T, k)$ et $v \in S(T, \{w\}_k)$. Supposons que $v \notin S(T, w)$, ce qui implique que $v \in S(T, k)$ car v est une paire et ne peut pas être un chiffré égal à $\{w\}_k$.
Les occurrences de v sous-termes de k sont donc « éliminées » par ce déchiffrement. Nous construisons alors une preuve plus petite de $T \vdash w$. Nous analysons les deux sous-preuves P_1 et P_2. Nous cherchons le chemin maximal à partir du nœud $\{W\}_k$, resp. k où tous les nœuds possèdent v comme sous-termes. Soit un des chemins finit sur une règle (A), dans ce cas nous concluons que $v \in S(T)$ ce qui contredit l'hypothèse $v \notin S(T, w)$. Soit les deux chemins finissent sur une application de la règle (P), alors nous remplaçons cette application de la règle par la sous-preuve qui se termine par $T \vdash v_1$, nous coupons donc toute la sous-preuve de v_2, et dans tous les nœuds des deux chemins, nous remplaçons le sous-terme v par v_1. Nous obtenons bien une preuve plus courte, ce qui contredit la minimalité de la preuve $T \vdash w$.

□

Ces deux lemmes nous permettent d'établir une premier résultat de localité.

4.2.3 Localité simple.

Nous montrons un premier résultat de localité (lemme 4) que nous appelons simple car il fait intervenir uniquement la notion de sous-termes syntaxiques et les démonstrations supposent des preuves simples. Pour cela nous regroupons les autres règles en une seule « macro » règle, que nous appellerons (M). Cette règle (M) prend toutes les hypothèses des règles qu'elle regroupe comme ses propres hypothèses et génère la même conclusion que l'ensemble des règles qu'elle regroupe. La règle (M) regroupe les règles (GX), (h), (D), (C). Ce nouveau système de déduction, dénoté S_M, comporte uniquement les règles (A), (UR), (UL), (P) et (M) et est présenté dans la figure 4.3 page suivante.

Lemme 4 *Soit P une preuve de $T \vdash w$ dans le système S_M, alors il existe une preuve P' de $T \vdash w$ dans S_M telle que tous les nœuds de P' sont dans $S(T, w)$.*

Preuve : Soit P une preuve de $T \vdash w$ dans S_M, nous considérons une preuve de $T \vdash w$ dans le système initial qui existe forcément par construction de la « macro » règle (M). Nous construisons

$$(A)\frac{u \in T}{T \vdash_E u} \qquad (UL)\frac{T \vdash_E \langle u, v\rangle}{T \vdash_E u}$$

$$(P)\frac{T \vdash_E u \qquad T \vdash_E v}{T \vdash_E \langle u, v\rangle} \qquad (UR)\frac{T \vdash_E \langle u, v\rangle}{T \vdash_E v}$$

$$(M)\frac{T \vdash_E u_1 \qquad \ldots \qquad T \vdash_E u_n}{T \vdash_E C[u_1, \ldots, u_n]}$$

FIG. 4.3 – Modèle de Dolev-Yao avec la « macro » règle (M), où $C[u_1, \ldots, u_n]$ est un contexte de l'application des règles (GX), (h), (D), (C).

à partir de cette preuve minimale de $T \vdash w$ dans ce système. Par le lemme 1 page 60 et le lemme 3 page 61, tous les nœuds issus des règles (UR), (UL) ou (P) sont dans $S(T, w)$. Nous reconstruisons alors à partir de cette preuve une preuve dans le système S_M, nous obtenons donc que :

- tous les nœuds hypothèses et conclusions des règles (UR), (UL) ou (P) sont dans $S(T, w)$.
- toutes les hypothèses de toutes les règles (M) sont issues soit de T par une règle axiome (A) ou d'une règle (UR), (UL) ou (P). Par conséquent ces hypothèses sont dans $S(T, w)$.
- Toutes les conclusions des règles (M) sont soit une hypothèse d'une règle (UR), (UL) ou (P), soit w la conclusion de la preuve P, par conséquent ces nœuds sont des éléments de $S(T, w)$.

Nous pouvons donc conclure que tous les nœuds de cette nouvelle preuve de $T \vdash w$ dans S_M sont dans $S(T, w)$. $\square$

Nous appliquons ce résultat de localité simple dans le chapitre suivant pour montrer la décidabilité du problème de déduction de l'intrus dans deux cas :

Premièrement, le symbole homomorphique h n'a aucune interaction dans la théorie équationnelle avec le symbole de chiffrement. Ce cas couvre les théories équationnelles de ACh, ACUNh, AGh et le cas de la théorie équationnelle vide, *i.e.* Dolev-Yao étendu sans théorie équationnelle. Nous devons alors établir des résultats similaires aux lemmes de la section 4.2.2 page 60 pour les règles (C), (UL) et (UR). Pour montrer ce résultat, nous reprenons la technique utilisée S. Delaune [Del06a] en fusionnant la règle (GX) et la règle (h), en une « macro » règle (GXh). Nous rappelons que la règle (h) permet d'appliquer la fonction homomorphique h à un terme. Il faut aussi étudier la déductibilité en une étape des règles de ce nouveau système.

Le second cas prend en compte les théories équationnelles où le symbole homomorphique sur l'opérateur $\oplus$ est le symbole de chiffrement. Pour obtenir le résultat de localité simple dans ce cas nous considérons une « macro » règle qui regroupe les règles (GX), (C) et (D), appelée $(GXCD)$. Une analyse plus fine de la déductibilité en une étape de cette règle grâce au $\mathcal{Z}$-modules est nécessaire et permet de démontrer que le problème de déduction de l'intrus est décidable.

L'exemple 7 montre que cette notion de sous-termes syntaxiques n'est pas suffisant pour prouver ce résultat dans le cas où le chiffrement est le symbole homomorphique.

Exemple 7 *Nous considérons la théorie équationnelle* $\mathsf{AG}\{.\}_.$ *(théorie où le chiffrement est distributif par rapport au symbole des groupes abéliens) et la preuve suivante où* $T = \{a - \{b\}_k, \{b\}_k - c, \{c\}_k - d, k\}$ *et* $w = \{a\}_k - d$ *où* $\Sigma_0 = \{a, b, c, d, k\}$*. Nous calculons*

$$S(T, w) = T \cup \{w\} \cup \{\{a\}_k, a, \{b\}_k, b, \{c\}_k, c, d, k\}$$

$$(GX)\dfrac{(C_k)\dfrac{(A)\dfrac{a-\{b\}_k \in T}{T \vdash a-\{b\}_k} \quad (A)\dfrac{k \in T}{T \vdash k}}{T \vdash \{a\}_k - \{\{b\}_k\}_k} \qquad (C_k)\dfrac{(A)\dfrac{\{b\}_k - c \in T}{T \vdash \{b\}_k - c} \quad (A)\dfrac{k \in T}{T \vdash k}}{T \vdash \{\{b\}_k\}_k - \{c\}_k} \qquad (A)\dfrac{\{c\}_k - d \in T}{T \vdash \{c\}_k - d}}{T \vdash \{a\}_k - d}$$

Cette preuve de $T \vdash w$ n'est pas S-locale car $\{\{b\}_k\}_k$ n'est pas dans $S(T,w)$.

$$(GX)\dfrac{(C_k)\dfrac{(GX)\dfrac{(A)\dfrac{a-\{b\}_k \in T}{T \vdash a-\{b\}_k} \quad (A)\dfrac{\{b\}_k - c \in T}{T \vdash \{b\}_k - c}}{T \vdash a - c} \qquad (A)\dfrac{k \in T}{T \vdash k}}{T \vdash \{a\}_k - \{c\}_k} \qquad (A)\dfrac{\{c\}_k - d \in T}{T \vdash \{c\}_k - d}}{T \vdash \{a\}_k - d}$$

Dans la seconde preuve $T \vdash w$, le terme $a - c$ n'est pas $S(T,w)$, ainsi cette preuve n'est pas S-locale.

Dans cette exemple, dans la première preuve nous avons appliqué la règle (GX) le plus tard, tandis que dans la seconde preuve les additions sont effectuées le plus tôt possible. Grâce à cette dernière sorte de preuve, que nous définirons plus précisément dans la section suivante, nous pouvons limiter le nombre de symboles de chiffrement utilisé dans les termes de la preuve. Ce point est un ingrédient important de notre approche pour démontrer la décidabilité du problème de déduction de l'intrus dans cette théorie équationnelle.

Nous montrons maintenant le résultat de localité dans le cas de la théorie vide pour le modèle de Dolev-Yao standard, décrit dans la figure 3.1 page 47.

4.3 Application au modèle Dolev-Yao standard.

Nous présentons ce résultat comme une conséquence de la localité simple du lemme 4 page 63. Nous considérons dans cette section uniquement le modèle de Dolev-Yao standard avec les règles (A), (UR), (UL), (P), (C) et (D).

Tout d'abord, le lemme 1 page 60 et le lemme 3 page 61 se démontrent facilement, de même que le cas du modèle de Dolev-Yao standard. Ainsi, nous savons que tous les nœuds hypothèses et conclusions des règles (UR), (UL) et (P) sont dans (P).

4.3.1 Analyse des règles (D) et (C).

Nous analysons donc les règles (D) et (C) plus en détail dans les lemmes suivants.

Lemme 5 *Soit P une preuve simple de $T \vdash u$. Si P est*

$$(D)\dfrac{(R)\dfrac{\vdots}{T \vdash \{u\}_v\downarrow = r} \qquad \dfrac{\vdots}{T \vdash v\downarrow}}{T \vdash u}$$

alors $\{u\}_v\downarrow \in S(T)$.

Preuve : Nous faisons une induction sur la structure de la preuve P.

Le cas de base est constitué d'une preuve qui ne contient que la règle d'axiome (A), nous obtenons donc immédiatement le résultat.

Nous regardons alors la dernière règle (R) utilisée dans la sous-preuve de P avec une racine $\{u\}_v\downarrow$:

- (R) est (A), (UL) ou (UR) : le résultat est vrai par définition de la règle axiome et le lemme 2 page 61.
- (R) est (P) : impossible car $\{u\}_v \downarrow$ n'est pas une paire.
- (R) est une règle (C) : tout d'abord (C_v) impossible car P est simple et $(C_{v'})$) car $\{u\}_v = \{u'\}_{v'}$ avec $v \neq v'$ ce qui est impossible.
- (R) est une règle (D) tel que $\dfrac{T \vdash \{\{u\}_v\}_{v'} \quad T \vdash v'}{T \vdash \{u\}_v}$. Alors par hypothèse d'induction $\{\{u\}_v\}_{v'} \in (T)$, nous concluons $\{u\}_v \in S(T)$

$\square$

Nous montrons le lemme 6 à propos de la règle de chiffrement (C), similaire au lemme 3 page 61 pour la règle (P).

Lemme 6 *Soit P une preuve minimale de $T \vdash w$. Si P contient une application de la règle de construction de chiffrement (C) de la forme suivante :*

$$(C)\dfrac{\dfrac{\vdots}{T \vdash u} \quad \dfrac{\vdots}{T \vdash k}}{T \vdash \{u\}_k}$$

alors $\{u\}_k \in S(T, w)$.

Preuve : Nous prouvons le résultat par induction structurelle sur la preuve P de $T \vdash w$. Nous regardons les différentes possibilités pour la dernière règle de la preuve P :

- La dernière règle est (A). Le résultat est immédiat.
- La dernière règle est (UR) ou (UL).

$$(UL)\dfrac{\dfrac{P_1}{T \vdash \langle w, v\rangle}}{T \vdash w}$$

  Par hypothèse d'induction nous savons que tous les nœuds issus d'une règle (C) dans la sous-preuve P_1 sont dans $S(T, \langle w, v\rangle)$. Comme la preuve P est minimale elle est alors simple (lemme 3 page 60), nous pouvons donc appliquer le lemme 1 page 60 et donc $\langle w, v\rangle \in S(T)$. Ainsi tous les nœuds issus d'une règle (C) sont dans $S(T) \subseteq S(T, w)$.
- La dernière règle est (P).

$$(P)\dfrac{\dfrac{P_1}{T \vdash u} \quad \dfrac{P_2}{T \vdash v}}{T \vdash \langle u, v\rangle = w}$$

  Par hypothèse d'induction nous savons que tous les nœuds issus d'une règle (C) dans la sous-preuve P_1 sont dans $S(T, u)$ et tous les nœuds issus d'une règle (P) dans la sous-preuve P_2 sont dans $S(T, v)$. Par définition de S $u \in S(\langle u, v\rangle) = S(w)$ et $v \in S(\langle u, v\rangle) = S(w)$, ainsi tous les nœuds issus d'une règle (C) sont dans $S(T, w)$.
- La dernière règle est (C). Pour la dernière occurrence de la règle (C) qui génère le terme final w ainsi le résultat est évidemment dans $S(T, w)$. Le reste de la démonstration est similaire au cas de la règle (P) précédemment analysée.

$$(C)\dfrac{\dfrac{P_1}{T \vdash u} \quad \dfrac{P_2}{T \vdash k}}{T \vdash \{u\}_k = w}$$

Par hypothèse d'induction nous savons que tous les nœuds issus d'une règle (C) dans la sous-preuve P_1 sont dans $S(T, u)$ et tous les nœuds issus d'une règle (C) dans la sous-preuve P_2 sont dans $S(T, k)$. Par définition de S $u \in S(\{u\}_k) = S(w)$ et $k \in S(\{u\}_k) = S(w)$, ainsi tous les nœuds issus d'une règle (C) sont dans $S(T, w)$.

□

4.3.2 Problème de déduction de l'intrus pour le modèle de Dolev-Yao standard.

Grâce à tous ces lemmes préliminaires et aux différentes notions de preuves et sous-termes introduites dans ce début de chapitre, nous sommes maintenant capable de redémontrer dans le théorème suivant le problème de déduction de l'intrus pour le modèle de Dolev-Yao standard.

Théorème 3 *Soit w un terme et T un ensemble de terme, le problème de déduction de l'intrus pour le terme w à partir des connaissance initiales T est décidable en temps polynômial.*

Preuve : Nous appliquons les lemmes 6 page ci-contre, 3 page 61, 5 page 65, 1 page 60 pour montrer la $S(T, w)$-localité, où S est la fonction qui construit les sous-termes syntaxiques en temps polynômial. Pour conclure en appliquant le théorème 2 page 57, nous montrons que la déductibilité en une étape pour notre système de preuve est décidable en temps polynomial, ce qui est immédiat, car chaque règle du système est finie *i.e.* possède un nombre fini d'hypothèses. Ainsi, le problème de déduction de l'intrus pour le terme w à partir des connaissances initiales T est décidable en temps polynômial. □

Chapitre 5

Problème de déduction de l'intrus et Dolev-Yao étendu.

« On fait la science avec des faits, comme on fait une maison avec des pierres : mais une accumulation de faits n'est pas plus une science qu'un tas de pierres n'est une maison. »

Henri Poincaré.

Sommaire

À la fin de la section 3.2.3 page 51, nous avons énoncé différentes théories équationnelles, nous commençons dans ce chapitre par analyser les théories équationnelles en présence d'un symbole homomorphique h sur un opérateur associatif et commutatif, puis sur l'opérateur du « ou exclusif » et enfin sur celui des groupes abéliens. Comme nous l'avons illustré dans le chapitre 2, la fonction de vérification de somme utilisée dans le protocole WEP (décrit dans la section 2.4.2.4 page 30, plus précisément dans la section 2.5.2.1 page 38) possède cette propriété sur le « ou exclusif ». Ensuite nous considérerons le cas où le symbole de chiffrement est distributif sur le « ou exclusif » et les groupes abéliens. Enfin nous regardons le cas où le chiffrement est commutatif et distributif sur le symbole du « ou exclusif » et des groupes abéliens.

Grâce au lemme 4 page 63 nous savons que dans une preuve minimale de $T \vdash w$ tous les nœuds hypothèses et conclusions des règles (UR), (UL) et (P) sont des éléments de $S(T, w)$. Nous analysons maintenant les autres règles de notre système de déduction afin d'obtenir une procédure de décision pour le problème de déduction de l'intrus.

5.1 Homomorphisme.

Nous avons proposé une procédure de décision pour le problème de déduction de l'intrus en temps exponentiel [LLT05] pour les théories ACUNh et AGh. Notre approche consiste à transformer les preuves pour arriver à borner le nombre d'applications de la fonction homomorphique et obtenir un résultat de localité pour appliquer le résultat étendu de McAllester du théorème 2 page 57. S. Delaune [Del06a] améliora ces résultats et proposa une procédure en temps polynomial en introduisant une « macro » règle pour traiter à la fois la règle (h) de création du symbole h et la règle d'addition (GX). Nous présentons notre résultat pour le cas ACh et utilisons l'idée de la procédure en temps polynômial pour les théories ACUNh et AGh introduite par S. Delaune, pour redémontrer ce résultat en utilisant le résultat de localité simple du chapitre 4.

5.1.1 ACh.

Dans la figure 5.1, nous rappelons le modèle de Dolev-Yao utilisé et les axiomes de la théorie équationnelle ACh :
- $(x \oplus y) \oplus z = x \oplus (y \oplus z)$ (Associativité)
- $x \oplus y = y \oplus x$ (Commutativité)
- $h(x \oplus y) = h(x) \oplus h(y)$ (Homomorphisme)

Ainsi, aucun symbole ne peut être détruit ni par la règle (GX) ni par la règle (h).

$$(A)\frac{u \in T}{T \vdash_E u} \qquad (UL)\frac{T \vdash_E \langle u, v\rangle}{T \vdash_E u}$$

$$(P)\frac{T \vdash_E u \qquad T \vdash_E v}{T \vdash_E \langle u, v\rangle} \qquad (UR)\frac{T \vdash_E \langle u, v\rangle}{T \vdash_E v}$$

$$(C)\frac{T \vdash_E u \qquad T \vdash_E v}{T \vdash_E \{u\}_v} \qquad (D)\frac{T \vdash_E \{u\}_v \qquad T \vdash_E v}{T \vdash_E u}$$

$$(GX)\frac{T \vdash_E u_1 \qquad \ldots \qquad T \vdash_E u_n}{T \vdash_E u_1 \oplus \ldots \oplus u_n} \qquad (h)\frac{T \vdash_E u}{T \vdash_E h(u)}$$

FIG. 5.1 – Dolev-Yao étendu pour la théorie équationnelle ACh.

Dans un premier temps, nous redémontrons les lemmes 6 page 66 et 5 page 65 obtenus dans le cas de Dolev-Yao standard et de la théorie vide, pour ACh. Ensuite nous considérons les règles (h) et (GX) pour obtenir un résultat de localité. Dans un second temps nous montrons que la déduction en une étape dans ce système peut se ramener à la résolution d'un système d'équations linéaires. Ce procédé usuel est détaillé dans le cas ACh.

5.1.1.1 Localité.

Nous analysons maintenant le comportement des deux nouvelles règles introduites (GX) et (h) dans les preuves des lemmes pour la règle (C) et (D). La structure des preuves des lemmes par induction reste identique que dans le cas de Dolev-Yao standard.

Lemme 7 *Soit P une preuve simple de $T \vdash u$. Si P est*

$$(D)\dfrac{(R)\dfrac{\vdots}{T \vdash \{u\}_v\downarrow = r} \quad \dfrac{\vdots}{T \vdash v\downarrow}}{T \vdash u}$$

alors $\{u\}_v\downarrow \in S(T)$.

Preuve : Nous faisons une induction sur la structure de la preuve P. Les cas des règles (P), (UR), (UL), (C) et (D) sont identiques au lemme 5 page 65. Le cas où la dernière règle de la preuve P est la règle (h) ou (GX), est impossible car $\{u\}_v\downarrow$ n'est ni un terme avec le symbole h en tête, ni le symbole $\oplus$. $\square$

Lemme 8 *Soit P une preuve minimale de $T \vdash w$. Si P contient une application de la règle de construction de chiffrement (C) de la forme suivante :*

$$(C)\dfrac{\dfrac{\vdots}{T \vdash u} \quad \dfrac{\vdots}{T \vdash k}}{T \vdash \{u\}_k}$$

alors $\{u\}_k \in S(T, w)$.

Preuve :

Nous faisons une induction sur la structure de la preuve P. Les cas des règles (P), (UR), (UL), (C) et (D) sont identiques au lemme 6 page 66.

– Le cas où la dernière règle de la preuve P est la règle (h),

$$(h)\dfrac{\dfrac{P_1}{T \vdash u}}{T \vdash h(u) = w}$$

Par hypothèse d'induction nous savons que tous les nœuds issus d'une règle (C) dans la sous-preuve P_1 sont dans $S(T, u)$. Par définition de S, $u \in S(h(u)) = S(w)$, ainsi tous les nœuds issus d'une règle (C) sont dans $S(T) \subseteq S(T, w)$.

– Le cas où la dernière règle de la preuve P est la règle (GX),

$$(GX)\dfrac{\dfrac{P_1}{T \vdash u_1} \quad \dots \quad \dfrac{P_n}{T \vdash u_n}}{T \vdash u_1 \oplus \ldots \oplus u_n = w}$$

Par hypothèse d'induction nous savons que tous les nœuds issus d'une règle (C) dans la sous-preuve P_i sont dans $S(T, u_i)$ pour $i = 1, \ldots, n$. Par définition de S, $u_i \in S(u_1 \oplus \ldots \oplus u_n) = S(w)$, ainsi tous les nœuds issus d'une règle (C) sont dans $zS(T, w)$.

$\square$

Ces preuves dans le cas ACh sont très similaires à celles de Dolev-Yao car la théorie équationnelle associée ne détruit aucun terme. Nous pouvons démontrer de même que pour la règle (C) des lemmes pour les règles (GX) et (h). Ces lemmes non détaillés ici prouvent que les conclusions des règles (GX) et (h) sont dans $S(T, w)$.

Lemme 9 *Soit P une preuve minimale de $T \vdash w$, alors elle est $S(T, w)$-locale.*

Preuve : C'est une conséquence immédiate des lemmes 4 page 63, 8 page précédente, 7 page 70 et des deux lemmes pour les règles (GX) et (h) indiquant que les conclusions de ces règles sont dans $S(T, w)$. □

Nous avons donc une fonction de sous-terme calculable en temps polynômial et un résultat de localité, nous regardons maintenant en détail la déductibilité en une étape.

5.1.1.2 Déduction en une étape.

Nous transformons le problème de déduction en une étape de la règle (GX) en la résolution d'un système d'équations linéaires diophantiennes. Cette transformation ressemble à la méthode utilisée par P. Narendran [Nar96] pour résoudre des problèmes d'unification en présence de théories équationnelles modulo AC. Nous rappelons que seule la règle (GX) pose un problème car elle représente potentiellement une famille infinie de règles. La déductibilité en une étape pour les autres règles est facile à mettre en œuvre car ces règles n'ont qu'un nombre fini d'hypothèses possibles.

Définition 11 (atoms) *Soit u un terme, nous définissons la fonction $atoms(u)$ de la manière suivante :*
- *Si $u = u_1 \oplus \ldots \oplus u_n$, où chaque u_i n'est pas en tête avec $\oplus$, alors $atoms(u) = \{u_1, \ldots, u_n\}$. Les u_i sont appelés* atomes *de u.*
- *Si u n'est pas en tête avec $\oplus$ alors $atoms(u) = u$.*

Nous généralisons la définition d'atomes aux ensembles de termes T en formes normales par $\text{atoms}(T) := \bigcup_{t \in T} \text{atoms}(t)$.

Nous détaillons la construction du système d'équations linéaires diophantiennes à partir du problème de déduction en une étape pour la règle (GX).
- Données :
  - un ensemble fini de termes $T = \{t_0, \ldots, t_n\}$
  - un terme s
- Sortie
  - Un système $D(T, s)$ d'équations linéaires diophantiennes sur les variables $X = \{x_0, \ldots, x_n\}$ telles que $T \vdash s$ si et seulement si $D(T, s)$ possède une solution.
- Algorithme
  - À chaque terme t_i nous associons une variable x_i pour $i = 0, \ldots, n$.
  - Si u est un atome de t, nous dénotons par $\delta(u, t)$ le nombre d' occurrences de l'atome u dans le terme t.
  - Soit $A = \{a_1, \ldots, a_m\}$ l'ensemble des atomes de $T \cup \{s\}$.
  - Pour chaque atome a_i de s, nous introduisons l'équation suivante :

  $$\delta(a_i, s) = \sum_{j=0}^{n} \delta(a_i, t_j) * x_j$$

  Cette équation relie le nombre d'occurrences de l'atome a_i dans le terme s est égal à la somme du nombre d'occurrences de l'atome a_i dans les termes t_j.
  Le système d'équations $D(T, s)$ est la conjonction de toutes ces équations :

  $$D(T, s) := \bigwedge_{i=1}^{m} \sum_{j=0}^{n} \delta(a_i, t_j) * x_j = \delta(a_i, s)$$

Exemple 8 *Soit $T = \{a_1 \oplus a_2 \oplus a_3, a_1 \oplus a_4, a_2 \oplus a_4\}$ et $s = a_1 \oplus a_2$, où tous les a_i ne sont pas en tête avec $\oplus$. Nous introduisons trois variables x_0, x_1, x_2 une pour chaque élément de T :*

$$\begin{array}{lll} x_0 & \textit{pour} & a_1 \oplus a_2 \oplus a_3 \\ x_1 & \textit{pour} & a_1 \oplus a_4 \\ x_2 & \textit{pour} & a_2 \oplus a_4 \end{array}$$

Pour chaque atome a_i nous créons une équation, et obtenons le système suivant :

$$\left\{ \begin{array}{lll} a_1 & : & x_0 + x_1 = 1 \\ a_2 & : & x_0 + x_2 = 1 \\ a_3 & : & x_0 = 0 \\ a_4 & : & x_1 + x_2 = 0 \end{array} \right.$$

Lemme 10 *Soit S un système d'équations de la forme suivante :*

$$\left\{ \begin{array}{ccc} c_{1,1}x_1 + \ldots + c_{1,n}x_n & = & d_1 \\ \vdots & \vdots & \vdots \\ c_{m,1}x_1 + \ldots + c_{m,n}x_n & = & d_m \end{array} \right.$$

$$w_S = d_1 * A_1 \oplus \ldots \oplus d_m * A_m$$

$$\Sigma = \{A_1, \ldots, A_m\}, T_S = \{t_1, \ldots, t_m\}$$

où $\forall i, 1 \leq i \leq n, t_i = c_{1,i} * A_1 \oplus \ldots \oplus c_{m,i} * A_m$ $\forall i, 1 \leq i \leq m, A_i$ sont des atomes de T_S.

Le système d'équations (S) a une solution si et seulement si le terme w est déductible en une étape à partir de T_S par la règle (GX).

Preuve :

– Si (S) possède une solution alors il existe une solution α de (S) telle que :

$$\left\{ \begin{array}{ccc} c_{1,1}\alpha(x_1) + \ldots + c_{1,n}\alpha(x_n) & = & d_1 \\ \vdots & \vdots & \vdots \\ c_{m,1}\alpha(x_1) + \ldots + c_{m,n}\alpha(x_n) & = & d_m \end{array} \right.$$

Ainsi, nous calculons w_S à partir de T_S et α :

$$\begin{array}{ll} & \alpha(x_1) * t_1 \oplus \ldots \oplus \alpha(x_n) * t_n \\ = & \alpha(x_1) * (c_{1,1} * A_1 \oplus \ldots \oplus c_{m,1} * A_m) \oplus \ldots \oplus \alpha(x_n) * (c_{1,n} * A_1 \oplus \ldots \oplus c_{m,n} * A_m) \\ = & c_{1,1} * \alpha(x_1) * A_1 \oplus \ldots \oplus c_{1,n} * \alpha(x_n) * A_1 \oplus \ldots \oplus c_{m,1} * \alpha(x_1) * A_m \oplus \ldots \oplus c_{m,n} * \alpha(x_n) * A_m \\ = & d_1 * A_1 \oplus \ldots \oplus d_m * A_m \\ = & w_S \end{array}$$

– Réciproquement, soit P une preuve de $T_S \vdash w_S$ en une étape avec la règle (GX). Nous construisons le système (S) à partir de T_S et w_S.

$$T_S = \{t_1, \ldots, t_m\}, \Sigma = \{A_1, \ldots, A_m\} = \mathrm{atoms}(T_S)$$

$$w_S = d_1 * A_1 \oplus \ldots \oplus d_m A_m$$

$\forall i, 1 \leq i \leq n, \exists c_{j,i} 1 \leq j \leq m, t_i = c_{1,i} * A_1 \oplus \ldots \oplus c_{m,i} * A_m$

Nous déduisons le terme w_S de T_S, il existe donc une décomposition de d_i en $c_{i,j}$,par conséquent nous obtenons le système suivant :

$$\left\{\begin{array}{ccc} c_{1,1}x_1 + \ldots + c_{1,n}x_n & = & d_1 \\ \vdots & \vdots & \vdots \\ c_{m,1}x_1 + \ldots + c_{m,n}x_n & = & d_m \end{array}\right.$$

□

Nous obtenons un système linéaire d'équations diophantiennes à coefficients $\mathbb{N}$. Ce système a une solution si et seulement si le terme s est déductible à partir de T en une étape. Résoudre un système linéaire d'équations diophantiennes à coefficients $\mathbb{N}$ est un problème NP-complet [Pap94], ainsi nous connaissons la complexité de la déductibilité en une étape pour cette théorie équationnelle.

5.1.2 ACUNh et AGh.

Les preuves pour les lemmes concernant les règles (D), (C), (GX) et (h) ne sont plus aussi simples car la règle (GX) peut faire disparaître certains termes qui ne seront donc pas dans $S(T, w)$.

Exemple 9 *Considérons la théorie équationnelle ACUNh, la preuve suivante où* $\{a \oplus \{b\}_k, h(a) \oplus h(\{b\}_k)\}$ *et* $w = h(a) \oplus h(c)$ *avec* $\Sigma_0 = \{a, b, c, k\}$. *Nous calculons* $S(T, w) = T \cup \{w\} \cup \{a, c, \{b\}_k, b, k\}$.

$$(GX)\dfrac{(h)\dfrac{(A)\dfrac{a \oplus \{b\}_k \in T}{T \vdash a \oplus \{b\}_k}}{T \vdash h(a) \oplus h(\{b\}_k)} \qquad (h)\dfrac{(A)\dfrac{\{b\}_k \oplus c \in T}{T \vdash \{b\}_k \oplus c}}{T \vdash h(\{b\}_k) \oplus h(c)}}{T \vdash h(a) \oplus h(c)}$$

Cette preuve n'est pas $S(T, w)$*-locale car le terme* $h(a) \oplus h(\{b\}_k)$ *n'est pas dans* $S(T, w)$.

L'approche originelle que nous avons développée [LLT05] est basée sur des transformations de preuves qui assurent que les applications de règles (GX) sont faites le plus tôt possible. Comme le montre l'exemple 9 si nous ne faisons pas cette transformation de preuve nous pouvons trouver des nœuds dans la preuve qui ne sont pas dans les sous-termes de T, w. Nous présenterons cette technique en détail pour les théories équationnelles avec chiffrement distributif dans la section 5.2 page 77. Elle s'applique aussi dans ce cadre, nous préférons présenter l'idée de S. Delaune qui regroupe la règle (GX) et (h) en une seule « macro » règle dénotée par (GXh) en appliquant notre résultat de localité syntaxique obtenu au chapitre précédent. Sur l'exemple 9, la preuve se transforme en une preuve qui ne contient qu'une seule « macro » règle (GXh) ce qui rend la preuve $S(T, W)$-locale. Dans la figure 5.2 page suivante, nous rappelons le modèle de Dolev-Yao utilisé pour les théories équationnelles ACUNh et AGh, en choisissant la notation usuelle pour le « ou exclusif ».

5.1.2.1 Localité.

Le lemme 4 page 63 nous indique que pour une preuve P de $T \vdash w$ les hypothèses et la conclusion des règles (UR), (UL) et (P) sont dans $S(T, w)$. Nous démontrons un lemme pour la règle (D) et ensuite nous prouverons directement le résultat de localité.

$$(A)\frac{u \in T}{T \vdash_E u} \qquad (UL)\frac{T \vdash_E \langle u, v\rangle}{T \vdash_E u}$$

$$(P)\frac{T \vdash_E u \qquad T \vdash_E v}{T \vdash_E \langle u, v\rangle} \qquad (UR)\frac{T \vdash_E \langle u, v\rangle}{T \vdash_E v}$$

$$(C)\frac{T \vdash_E u \qquad T \vdash_E v}{T \vdash_E \{u\}_v} \qquad (D)\frac{T \vdash_E \{u\}_v \qquad T \vdash_E v}{T \vdash_E u}$$

$$(GXh)\frac{T \vdash_E u_1 \qquad \dots \qquad T \vdash_E u_n}{T \vdash_E C[u_1 \oplus \dots \oplus u_n]}$$

FIG. 5.2 – Dolev-Yao étendu pour la théorie équationnelle ACUNh ou AGh, où C est un contexte composé de symboles h ou $\oplus$.

Lemme 11 *Soit P une preuve simple et aplatie de $T \vdash u$. Si P est*

$$(D)\frac{(R)\dfrac{\vdots}{T \vdash \{u\}_v\downarrow = r} \qquad \dfrac{\vdots}{T \vdash v\downarrow}}{T \vdash u}$$

alors $\{u\}_v\downarrow \in S(T)$.

Preuve : Nous faisons une induction sur la structure de la preuve P. Le cas de base (A) est vrai. Les cas des règles (P), (UR), (UL) sont une conséquence du lemme 4 page 63. Si la règle (R) est :

- (D) : l'hypothèse d'induction et la définition de S nous permettent de conclure.
- (C) : impossible par simplicité de la preuve P.
- (GXh) : Nous analysons en détail la structure de la preuve dans ce cas.

$$(D)\frac{(GXh)\dfrac{(R_1)\dfrac{\vdots}{T \vdash u_1} \qquad \dots \qquad (R_n)\dfrac{\vdots}{T \vdash u_n}}{T \vdash \{u\}_v\downarrow = r} \qquad \dfrac{\vdots}{T \vdash v\downarrow}}{T \vdash u\downarrow}$$

  - Si (R_i) est (A), (P), (UR) ou (UL) alors le résultat de ces règles est dans $S(T, w)$ grâce aux lemmes précédents.
  - La règle (R_i) ne peut être (GXh) car la preuve est aplatie.
  - Si (R_i) est la règle (C), par simplicité de la preuve, le résultat de cette règle (C) ne peut être $\{u\}_v$. Si le résultat de cette règle de chiffrement est un autre terme u_i chiffré par une autre clef, ce terme sera forcément éliminé par le résultat u_j d'une autre règle (R_j). La règle (R_j) n'est pas (GXh) car la preuve est aplatie, ni la règle (P) car u_j doit annuler u_i qui est un chiffré. Si (R_j) est une règle (A), (UR) ou (UL) nous concluons que u_j est dans $S(T)$ et donc u_i aussi. Nous concluons donc $\{u\}_v \in S(T, w)$

$\square$

Lemme 12 *Soit P une preuve minimale de $T \vdash w$, alors P est $S(T, w)$-locale.*

Preuve :

Nous faisons une induction sur la structure de la preuve P.

Le cas de base (A) est vrai.

- Si la dernière règle (R) est : (P), (UR), (UL) alors le lemme 4 page 63 nous permet de conclure.
- Le lemme 11 page 74 résout le cas où la dernière règle est (D), car une preuve minimale est simple et aplatie d'après la proposition 3 page 60 .
- (C) : l'hypothèse d'induction et la propriété de S (1 page 59) nous permettent de conclure, car $u, k \in S(\{u\}_k) = S(w)$.
- (GXh) : la règle se présente sous cette forme :

$$(GXh)\frac{(R_1)\dfrac{P_1}{T \vdash u_1} \quad \dots \quad (R_n)\dfrac{P_n}{T \vdash u_n}}{T \vdash w}$$

  L'hypothèse d'induction nous dit que toutes les u_i hypothèses de cette dernière règle (GXh) sont dans $S(T, u_i)$. Si $u_i \in S(w)$ le résultat est obtenu, sinon $u_i \notin S(w)$, donc nous montrons que $u_i \in S(T)$. Regardons plus précisément les règles (R_i).
  - (R_i) n'est pas (GXh) car (P) est minimale donc aplatie.
  - Si (R_i) est la règle (A), (UR) ou (UL) nous concluons que u_i est dans $S(T)$.
  - Si (R_i) est la règle (P) nous utilisons le lemme 4 page 63 pour conclure.
  - Si (R_i) est la règle (C), par simplicité de la preuve, le résultat de cette règle (C) ne peut être $\{u\}_v$. Si c'est un autre terme u_i chiffré par une autre clef, ce terme sera forcément éliminé par le résultat u_j d'une autre règle (R_j). La règle (R_j) n'est pas (GXh) car la preuve est aplatie, ni la règle (P) car u_j doit annuler u_i qui est un chiffré. Si (R_j) est une règle (A), (UR) ou (UL) nous concluons que u_j est dans $S(T)$ et donc u_i aussi.

□

Grâce à ce résultat de localité, pour la fonction de sous-terme syntaxique nous avons rempli les deux premiers points du lemme 2 page 57. Ce résultat s'obtient de même pour le cas e la théorie équationnelle AGh. Nous analysons maintenant la déduction en une étape pour ces deux théories.

5.1.2.2 Déduction en une étape.

Nous utilisons la même transformation que dans le cas ACh. Ramenons le problème à la résolution de système d'équations linéaires diophantiennes, mais cette fois, nous ne les résolvons pas sur $\mathbb{N}$.

ACUNh Dans ce cas nous résolvons les équations sur $\mathbb{Z}/2\mathbb{Z}[h]$. Ce problème se résout en temps polynomial [KKS87]. Nous obtenons donc un algorithme P-TIMEpour la déductibilité en une étape.

AGh Pour les groupes abéliens, nous devons d'abord étendre la notion de sous-terme et d'atomes. Nous considérons que $+$ est l'opérateur additif des groupes abéliens. Nous rappelons que nous dénotons par $na := a + a + \ldots + a$, n fois.

Nous redéfinissons maintenant fonction δ :

- si u est un atome de t qui n'est pas en tête avec $-$ alors $\delta(u, t)$ correspond au nombre d'occurrences de l'atome u dans le terme t,
- sinon, si $-u$ est un atome de t alors nous notons $\delta(u, t)$ l'opposé du nombre d occurrences de $-u$ dans t.

Nous ramenons donc le problème de déductibilité en une étape pour la règle (GXh) au problème de résoudre un système d'équations linéaires diophantiennes sur $\mathbb{Z}[h]$. La résolution d'un tel système s'effectue en temps polynômial [Sch86, PS82]. Nous obtenons donc un algorithme P-TIMEpour la déductibilité en une étape.

Exemple 10 *Soit $T = \{a_1 + a_2 - a_3 - a_3, a_1 + a_4, a_2 + -a_4\}$ et $s = a_1 + a_2$, où tous les a_i ne sont pas en tête avec + ni avec −. Pour chaque terme de T nous associons une variable :*

$$\begin{array}{lll} x_0 & \textit{pour} & a_1 + a_2 - a_3 - a_3 \\ x_1 & \textit{pour} & a_1 + a_4 \\ x_2 & \textit{pour} & a_2 - a_4 \end{array}$$

Pour tous les atomes a_i nous générons une équation, pour finalement obtenir le système d'équations suivant : système :

$$\begin{cases} a_1 & : & x_0 + x_1 = 1 \\ a_2 & : & x_0 + x_2 = 1 \\ a_3 & : & -2x_0 = 0 \\ a_4 & : & x_1 - x_2 = 0 \end{cases}$$

5.2 Chiffrement homomorphique.

Comme le montre l'exemple 7 page 64, considérer uniquement les sous-termes syntaxiques ne permet pas d'obtenir un résultat de localité pour les théories équationnelles $\mathsf{ACUN}\{.\}_.$ et $\mathsf{AG}\{.\}_.$ avec chiffrement commutatif ou non. Nous devons étudier plus précisément l'interaction entre le chiffrement, le déchiffrement et l'opérateur sur lequel le chiffrement distribue. Pour cela nous montrons un résultat de localité pour les atomes des nœuds en manipulant les arbres de preuve afin d'effectuer les règles (GX) le plus tôt possible dans l'arbre de preuve. Ce résultat nous permettra de conclure directement dans le cas du « ou exclusif », pour les groupes abéliens où une analyse plus fine est nécessaire pour contrôler les coefficients devant les termes. La règle de construction pour le symbole h n'est plus utilisé, nous travaillons donc avec le modèle de Dolev-Yao de la figure 5.3, pour la théorie équationnelle $\mathsf{ACUN}\{.\}_.$.

$$(A)\frac{u \in T}{T \vdash_E u} \qquad (UL)\frac{T \vdash_E \langle u, v\rangle}{T \vdash_E u}$$

$$(P)\frac{T \vdash_E u \qquad T \vdash_E v}{T \vdash_E \langle u, v\rangle} \qquad (UR)\frac{T \vdash_E \langle u, v\rangle}{T \vdash_E v}$$

$$(C)\frac{T \vdash_E u \qquad T \vdash_E v}{T \vdash_E \{u\}_v} \qquad (D)\frac{T \vdash_E \{u\}_v \qquad T \vdash_E v}{T \vdash_E u}$$

$$(GX)\frac{T \vdash_E u_1 \qquad \dots \qquad T \vdash_E u_n}{T \vdash_E u_1 \oplus \dots \oplus u_n}$$

FIG. 5.3 – Dolev-Yao étendu pour la théorie équationnelle $\mathsf{ACUN}\{.\}_.$.

Notre approche consiste à considérer une seule « macro » règle qui regroupe les règles (GX), (C) et (D). Le théorème 2 page 57 et le lemme 4 page 63 nous permettent de nous concentrer uniquement sur la déduction en une étape pour cette nouvelle règle car, nous avons la localité syntaxique pour le système $S_{(GXCD)}$ qui ne contient que les règles (A), (UR), (UL), (P) et $(GXCD)$.

Pour montrer la déductibilité en une étape de cette règle ($GXCD$) il est nécessaire d'analyser plus finement les preuves utilisées. Cette étude plus exacte, effectuée dans la prochaine section, permet d'obtenir un résultat de localité « atomique ». Grâce à ce résultat nous déduisons immédiatement deux conséquences indispensables pour la résolution du problème de déductibilité en une étape :

- Tous les nœuds sont constitués de termes présents dans les connaissances initiales de l'intrus et dans le but de la preuve (Lemme 15 page 82).
- Toutes les clefs utilisées comme hypothèse des règles (C) et (D) sont également calculables à partir de T et w (Lemme 16 page 83).

5.2.1 Localité pour les atomes.

Nous présentons ce résultat de localité « atomique » uniquement pour la théorie AGh, toutes les preuves et les résultats de cette section sont beaucoup plus simples à obtenir dans le cas ACUNh, ainsi nous ne les détaillerons pas.

5.2.1.1 Définitions.

Tout d'abord, nous raffinons la notion d'atomes introduite dans la définition 11 page 72.

Définition 12 (Atomes) *L'ensemble des* atomes *d'un terme t en forme normale est défini par :*

- $atoms(t_1 \oplus t_2) = atoms(t_1) \cup atoms(t_2)$
- $atoms(-t) = atoms(t)$
- $atoms(\{t_1\}_{t_2}) = \{\{t_1\}_{t_2}\} \cup atoms(t_1) \cup atoms(t_2)$
- $atoms(t) = \{t\}$ *si t n'est pas en tête avec $\oplus$ ni en tête avec $-$.*

Nous étendons cette notion de manière naturelle à un ensemble de termes T en forme normale par $atoms(T) = \bigcup_{t \in T} atoms(t)$, de plus $atoms(T, t)$ dénotera $atoms(T \cup \{t\})$.

Nous dégageons deux propriétés sur les atomes.

Proposition 4 *Pour tout terme t, $atoms(t) \subseteq S(t)$.*

Proposition 5 *Soit $atoms(t) = \{a_1, \ldots, a_n\}$. Alors il existe $\alpha_1, \ldots, \alpha_n \in \mathbb{Z} \setminus \{0\}$ tel que $t = \sum_{i=1}^{i=n} \alpha_i a_i$.*

Preuve : Ces deux propriétés sont des conséquences de la définition 9 page 58 de sous-termes syntaxiques et de la définition 12 d'atomes. $\square$

Remarquons qu'il existe une instance de la règle ($GXCD$) :

$$(GXCD)\frac{T \vdash t_1 \qquad \ldots \qquad T \vdash t_n}{T \vdash t}$$

si et seulement s'il existe une preuve de $T \vdash t$ qui n'utilise que les règles (A), (GX), (C) et (D).

Nous introduisons une nouvelle notion de preuve qui caractérise les atomes des nœuds d'une preuve.

Définition 13 (Preuve atomique) *Une preuve de $T \vdash w$ qui n'utilise que les règles (A), (C), (D) et (GX) est une preuve* atomique*, si pour chaque nœud de $T \vdash u$ nous avons $atoms(u) \subseteq atoms(T, w)$.*

Nous montrons dans cette section que pour toute preuve de $T \vdash w$ qui n'utilise que les règles (A), (C), (D) et (GX) il existe une preuve atomique de $T \vdash w$.

Nous caractérisons maintenant les preuves qui effectuent le plus tôt possible la règle (GX), nous les appelons des preuves $\oplus$-*eager*.

Définition 14 (Preuve $\oplus$-*eager*) *Soit P une preuve aplatie de $T \vdash w$. P est une preuve $\oplus$-**eager** si :*

1. *Pour tout v il y a au plus une règle (C_v) avec la clef v immédiatement au-dessus d'une règle (GX)dans P,*
2. *et il n'y pas de règle (D_v) juste après une règle (GX) avec une règle (C_v) au-dessus de la règle (GX).*

Revenons sur l'exemple 7 page 64. La première preuve est simple mais non $\oplus$-*eager* car il y a deux applications de la règle (C_k) au-dessus de la règle (GX), alors que la seconde est $\oplus$-*eager* est simple.

Exemple 11 *Soit la preuve suivante de $T \vdash b$ n'est pas $\oplus$-eager.*

$$(D_k)\dfrac{(GX)\dfrac{(A)\dfrac{a+\{b\}_k}{T\vdash a\oplus\{b\}_k} \qquad (A)\dfrac{\{c\}_k - a}{T\vdash \{c\}_k - a} \qquad (C_k)\dfrac{(A)\dfrac{c\in T}{T\vdash c}\ (A)\dfrac{k\in T}{T\vdash k}}{T\vdash\{c\}_k}}{T\vdash \{b\}_k\downarrow} \qquad (A)\dfrac{k\in T}{T\vdash k}}{T\vdash b}$$

Alors que la preuve suivante est $\oplus$-eager.

$$(GX)\dfrac{(D_k)\dfrac{(GX)\dfrac{(A)\dfrac{a+\{b\}_k}{T\vdash a\oplus\{b\}_k}\qquad (A)\dfrac{\{c\}_k-a}{T\vdash\{c\}_k-a}}{T\vdash\{b\}_k+\{c\}_k}\qquad (A)\dfrac{k\in T}{T\vdash k}}{T\vdash b+c}\qquad (A)\dfrac{c\in T}{T\vdash c}}{T\vdash b}$$

5.2.1.2 Transformations de preuves.

Nous présentons les transformations de preuves locales nécessaires pour montrer que toute preuve peut être transformée en une preuve atomique.

Proposition 6 *Toutes les transformations de preuves décrites dans les figures 4.2 page 59, 5.4 page suivante et 5.5 page suivante font décroître globalement le nombre de nœuds de la preuve.*

Preuve : Nous dénotons par π_x la sous-preuve de P avec pour racine $T \vdash x$. Toutes ces transformations de preuves transforment une preuve avec certaines hypothèses et une conclusion en une preuve avec les mêmes hypothèses et la même conclusion.

– Dans la figure 4.2 page 59, nous observons facilement que le nombre de nœuds décroît.
– Dans la figure 5.4 page suivante, le nombre de nœuds initial est de : $\Sigma_{i=1}^{i=m}|\pi_{z_i}| + \Sigma_{i=1}^{i=n}|\pi_{x_i}| + n|\pi_v| + n + 1$ et la preuve finale contient $\Sigma_{i=1}^{i=m}|\pi_{z_i}| + \Sigma_{i=1}^{i=n}|\pi_{x_i}| + |\pi_v| + 3$ nœuds. le nombre global de nœuds a diminué car $n \geq 2$.

$$
(GX)\,\dfrac{(C_v)\dfrac{T \vdash x_1 \qquad T \vdash v}{T \vdash \{x_1\}_v} \ldots (C_v)\dfrac{T \vdash x_n \qquad T \vdash v}{T \vdash \{x_n\}_v}\,(R_1)\dfrac{\vdots}{T \vdash z_1} \ldots (R_m)\dfrac{\vdots}{T \vdash z_m}}{T \vdash \alpha_1\{x_1\}_v \oplus \ldots \oplus \alpha_n\{x_n\}_v \oplus \beta_1 z_1 \oplus \ldots \beta_m \oplus z_m}
$$

$$\Downarrow$$

$$
(GX)\,\dfrac{(C_v)\dfrac{(GX)\dfrac{T \vdash x_1 \quad \ldots \quad T \vdash x_n}{T \vdash \alpha_1 x_1 \oplus \ldots \oplus \alpha_n x_n} \qquad T \vdash v}{T \vdash \alpha_1\{x_1\}_v \oplus \ldots \oplus \alpha_n\{x_n\}_v}\,(R_1)\dfrac{\vdots}{T \vdash z_1} \ldots (R_m)\dfrac{\vdots}{T \vdash z_m}}{T \vdash \alpha_1\{x_1\}_v \oplus \ldots \oplus \alpha_n\{x_n\}_v \oplus \beta_1 z_1 \oplus \ldots \oplus \beta_m z_m}
$$

FIG. 5.4 – Permutation des règles (C_v) et (GX) si toutes les règles (R_i) sont différentes de (C_v) et si $n \geq 2$.

$$
(D_v)\,\dfrac{(GX)\dfrac{(R_1)\dfrac{T \vdash B_1}{T \vdash B_1'} \quad \ldots \quad (R_n)\dfrac{T \vdash B_n}{T \vdash B_n'} \qquad (C_v)\dfrac{T \vdash B \qquad T \vdash v}{T \vdash \{B\}_v}}{T \vdash \alpha_1 B_1' \oplus \ldots \oplus \alpha_n B_n' \oplus \alpha\{B\}_v = \{u\}_v} \qquad T \vdash v}{T \vdash u}
$$

$$\Downarrow$$

$$
(D_v)\,\dfrac{(GX)\dfrac{(GX)\dfrac{(R_1)\dfrac{T \vdash B_1}{T \vdash B_1'} \quad \ldots \quad (R_n)\dfrac{T \vdash B_n}{T \vdash B_n'}}{T \vdash \alpha_1 B_1' \oplus \ldots \oplus \alpha_n B_n' = \{c\}_v} \qquad (C_v)\dfrac{T \vdash B \qquad T \vdash v}{T \vdash \{B\}_v}}{T \vdash \{c\}_v \oplus \alpha\{B\}_v = \{u\}_v} \qquad T \vdash v}{T \vdash c \oplus \alpha B = u}
$$

$$\Downarrow$$

$$
(GX)\,\dfrac{(D_v)\dfrac{(GX)\dfrac{(R_1)\dfrac{T \vdash B_1}{T \vdash B_1'} \quad \ldots \quad (R_n)\dfrac{T \vdash B_n}{T \vdash B_n'}}{T \vdash \alpha_1 B_1' \oplus \ldots \oplus \alpha_n B_n' = \{c\}_v} \qquad T \vdash v}{T \vdash c} \qquad T \vdash B}{T \vdash c \oplus \alpha B = u}
$$

FIG. 5.5 – Simplification de la règle (C_v) et (D_v) s'il existe une règle (C_v) au-dessus de la règle (GX) et la règle (D_v) juste après la même règle (GX), avec $n \geq 2$.

– Dans In Figure 5.5 page précédente, la preuve initiale possède $\Sigma_{i=1}^{i=n}|\pi_{B'_i}| + |\pi_B| + 2|\pi_v| + 3$ nœuds et la preuve finale a $\Sigma_{i=1}^{i=n}|\pi_{B'_i}| + |\pi_B| + |\pi_v| + 3$ nœuds. Par conséquent,le nombre de nœuds décroît globalement.

□

Lemme 13 *Soit P une preuve de $T \vdash w$ alors il existe une preuve $\oplus$-eager et simple de $T \vdash w$.*

Preuve : Soit P une preuve de $T \vdash w$, nous appliquons à P les transformations de règles des figures 4.2 page 59, 5.4 page précédente et 5.5 page ci-contre qui diminuent la taille de $|P|$, donc l'application de ces transformations termine. Comme le nombre d'occurrence de cas problématique pour obtenir une règle simple et $\oplus$-*eager* décroît également, nous obtenons bien au final une preuve $\oplus$-*eager* et simple de $T \vdash w$. □

5.2.1.3 Lemme pour la règle (D).

Le point clef de ce raisonnement consiste à montrer le lemme suivant qui assure que pour toute règle de déchiffrement d'un message chiffré $\{m\}_k$, les atomes de ce message sont en fait des éléments des atomes des connaissances initiales de l'intrus.

Lemme 14 *Soit P une preuve $\oplus$-eager et simple de $T \vdash u$ de la forme suivante :*

$$(D)\dfrac{(R)\dfrac{\vdots}{T \vdash \{u\}_v \downarrow = r} \quad \dfrac{\vdots}{T \vdash v \downarrow}}{T \vdash u}$$

alors atoms$(\{u\}_v) \subseteq$ atoms(T).

Preuve : Nous raisonnons par induction structurelle sur P.

Le cas de base correspond à la règle (A), le lemme est vrai dans ce cas.

Nous continuons notre analyse de cas en regardant la dernière (R) de la sous-preuve de P qui se termine sur $\{u\}_v \downarrow$.

– (R) est (A) : Le résultat se déduit de la définition de la règle (A).
– (R) est une règle de chiffrement (C) : ce cas est impossible car soit (C) est la règle (C_v) ce qui contredirait la simplicité de P, soit (C) est la règle $(C_{v'})$ donc $\{u\}_v = \{u'\}_{v'}$ avec $v \neq v'$ ce qui est impossible.
– (R) est une règle (D) telle que : $\dfrac{T \vdash \{\{u\}_v\}_{v'} \quad T \vdash v'}{T \vdash \{u\}_v}$. Par hypothèse d'induction atoms$(\{\{u\}_v\}_{v'}) \subseteq$ atoms(T), et par définition d'atoms nous savons que atoms$(\{u\}_v) \subseteq$ atoms(T).
– (R) est la règle (GX) : Les différentes possibilités pour les règles au-dessus de la règle (GX) sont décrites dans la figure 5.6 page suivante. Nous analysons les différents cas possibles pour les règles (R_i) :
  Nous montrons que chaque atome de $\{u\}_v \downarrow$ est dans atoms(T). Soit $a \in$ atoms$(\{u\}_v \downarrow)$, nous en déduisons que a est de la forme $\{a'\}_v$, et il existe un i tel que $a \in$ atoms(u_i). Nous énumérons tous les cas possibles pour (R_i) :
  – (R_i) est la règle (A), par conséquent $a \in$ atoms(T).
  – (R_i) est la règle $(D_{v'})$ telle que $(D_{v'})\ \dfrac{T \vdash \{w_1\}_{v'}\ T \vdash v'}{T \vdash w_1 = u_i}$.
    Par induction atoms$(\{w_1\}_{v'}) \subseteq$ atoms(T), avec la définition d'atoms nous concluons que atoms$(u_i) \subseteq$ atoms(T) et donc $a \in$ atoms(T).

$$
(D_v)\dfrac{(GX)\dfrac{(R_1)\dfrac{\vdots}{T \vdash u_1} \quad \dots \quad (R_n)\dfrac{\vdots}{T \vdash u_n}}{T \vdash \{u\}_v \downarrow} \qquad \dfrac{\vdots}{T \vdash v \downarrow}}{T \vdash u \downarrow}
$$

FIG. 5.6 – Illustration du lemme 14 page précédente.

- (R_i) est la règle (C_v) ou (GX) : Impossible car la preuve est $\oplus$-*eager* et aplatie.
- (R_i) est la règle $(C_{v'})$ avec $v \neq v'$, alors $u_i = \{u'\}_{v'} \downarrow$. Comme $v' \neq v$ aucun élément de atoms$(\{u'\}_{v'} \downarrow)$ n'est égal à a, tous ces atomes sont éliminés par d'autres occurrences du même atome dans un autre u_j avec $j \neq i$. Comme la preuve est $\oplus$-*eager* et aplatie il n'est pas possible que les autres termes soient issus d'une règle (GX) ou d'une règle (C), par conséquent les autres termes sont issus uniquement des règles (A) ou (D). Ainsi dans le premier cas les atomes sont donc dans atoms(T), dans le second cas nous appliquons l'hypothèse de récurrence, pour conclure que atoms$(u_i) \subseteq$ atoms(T).

□

5.2.1.4 Localité pour les atomes.

Nous possédons maintenant tous les éléments pour démontrer l'existence d'une preuve atomique, et donc prouver les deux lemmes sur les clefs des preuves indispensables pour assurer la déductibilité en une étape de la règle $(GXCD)$ dans la prochaine section.

Lemme 15 *Toute preuve simple et $\oplus$-eager est atomique.*

Preuve : Nous procédons par induction structurelle sur la preuve P et faisons une distinction de cas pour la dernière règle (R) de P une preuve simple et $\oplus$-*eager* de $T \vdash u$:

- (R) est (A) : P est une preuve atomique.
- (R) est la règle (D) telle que $\dfrac{T \vdash \{u\}_v \quad T \vdash v}{T \vdash u}$. D'après l'hypothèse d'induction la preuve P_1 qui se termine par $T \vdash \{u\}_v$ et la preuve P_2 qui génère $T \vdash v$ sont atomiques, c'est à dire que tous les nœuds $T \vdash w$ de P_1, resp. P_2, sont tels que atoms$(w) \subseteq$ atoms$(T, \{u\}_v)$, resp. atoms$(w) \subseteq$ atoms(T, v). Or la preuve P est $\oplus$-*eager*, d'après le lemme 14 page précédente nous savons que atoms$(\{u\}_v) \subseteq$ atoms$(T) \subseteq$ atoms(T, w). Nous concluons donc que atoms$(w) \subseteq$ atoms(T, u) pour tous nœuds $T \vdash w$ de P.
- (R) est une règle de chiffrement (C) : nous avons donc $u = \{u_1\}_{u_2}$ tel que $\dfrac{T \vdash u_1 \quad T \vdash u_2}{T \vdash \{u_1\}_{u_2}}$.
  L'hypothèse d'induction assure que la preuve P_1 qui se termine par le nœud $T \vdash u$ et la preuve P_2 qui se termine par le nœud $T \vdash v$ sont deux preuves atomiques, ainsi tous les nœuds $T \vdash w$ de P_1, resp. P_2, sont tels que atoms$(w) \subseteq$ atoms(T, u_1), resp. atoms$(w) \subseteq$ atoms(u_2). Nous concluons par le fait que atoms$(T, u_1) \subseteq$ atoms$(T, \{u_1\}_{u_2})$ et atoms$(T, u_2) \subseteq$ atoms$(T, \{u_1\}_{u_2})$.
- (R) est une règle (GX) telle que :

$$
(GX)\dfrac{(R_1)\dfrac{\vdots}{T \vdash u_1} \quad \dots \quad (R_n)\dfrac{\vdots}{T \vdash u_n}}{T \vdash u}
$$

Par induction chaque preuve P_i qui se termine par $T \vdash u_i$ est atomique, ainsi tous les nœuds $T \vdash w$ de P_i sont tels que $\mathrm{atoms}(w) \subseteq \mathrm{atoms}(T, u_i)$. Nous montrons que $\mathrm{atoms}(u_i) \subseteq \mathrm{atoms}(T, u)$ pour tous les i. Nous regardons les différents cas possibles pour les règles (R_i) de la preuve P_i.

– (R_i) est (GX) : Impossible car P est $\oplus$-*eager*, donc aplatie.
– (R_i) est (A), (D) : Par le lemme 14 page 81, $\mathrm{atoms}(u_i) \subseteq \mathrm{atoms}(T) \subseteq \mathrm{atoms}(T, u)$.
– (R_i) est (C_k) : Soit $a \in \mathrm{atoms}(u_i)$, et supposons que $a \notin \mathrm{atoms}(T, u)$. Comme u est issu de l'application de la règle (C_k) nous savons que a est de la forme $\{a'\}_k$.
Comme $a \notin \mathrm{atoms}(u)$, le terme a est « éliminé » par un autre terme u_j, $j \neq i$, avec $a \in \mathrm{atoms}(u_j)$. Comme nous avons supposé que $a \notin \mathrm{atoms}(T)$ nous en déduisons que la dernière règle (R_j) de la preuve P_j ne peut être ni (A), ni (D), et ni (C). Par conséquent, R_j est une règle de chiffrement(C). Plus précisément, comme $a = \{a'\}_k$ nous savons que R_j est (C_k), nous obtenons donc deux applications de la règle (C_k) au-dessus d'une règle (GX), ce qui contredit l'hypothèse que P soit une preuve $\oplus$-*eager*.

□

Définition 15 (Terme en position clef) *Un terme v est en* position clef *d'un terme w en forme normale s'il existe un terme t tel que $\{t\}_v \in S(w)$.*

Lemme 16 *Soit P une preuve $\oplus$-eager et simple de $T \vdash w$. Tous les termes qui apparaissent en position clef dans un nœud de P sont dans $S(T, w)$.*

Preuve : Soit P une preuve $\oplus$-*eager* et simple de $T \vdash w$. Si un terme k apparaît en position clef d'un terme t qui apparaît dans la preuve P alors c'est un sous-terme syntaxique d'un atome de t. Grâce au lemme 15 page précédente, un sous-terme t est un un atome de T, w. En utilisant la proposition 4 page 78, nous obtenons $k \in S(T, w)$. □

Le lemme 16 nous permet de raffiner la notion de « macro » règle contenant (GX), (C) and (D). :

Définition 16 (GCD-arbre de preuve) *Un arbre de preuve P est un* GCD-arbre de preuve *l'ensemble L de feuilles, l'ensemble K des clefs, et al racine u si :*

1. *P consiste en un unique nœud $T \vdash u$ et $L = \{u\}$, $K = \emptyset$,*
2. *P est de la forme (C) $\dfrac{P \qquad T \vdash k}{T \vdash u}$ où P est un GCD-arbre de preuve avec racine u', ensemble de feuilles L et ensemble de clefs K', $K = K' \cup \{k\}$, et $\{u'\}_k\downarrow = u$,*
3. *P est de la forme (D) $\dfrac{P \qquad T \vdash k}{T \vdash u}$ où P est un GCD-arbre de preuve avec racine u', ensemble de feuilles L et ensemble de clef K', $K = K' \cup \{k\}$, et $\{u\}_k\downarrow = u'$,*
4. *P est de la forme (GX) $\dfrac{P_1 \ \cdots \ P_n}{u}$ avec $n \geq 1$ tel que chaque P_i est un GCD-arbre de preuve avec*
respectivement racine u_i, ensemble de feuilles L_i, et ensemble de clefs K_i, et $K = \bigcup_{i=1}^{n} K_i \cup K'$ et $L = \bigcup_{i=1}^{n} L_i$.

En particulier, les instances des règles (GX), (C), ou (D) est un GCD-arbre de preuve.

5.2.2 Étude de la déductibilité en une étape pour $(GXCD)$.

Nous montrons la déductibilité en une étape pour la règle $(GXCD)$.

Définition 17 (Terme déductible atomiquement) *Soit w un terme en forme normale, w est* déductible atomiquement *à partir d'un ensemble fini de termes L s' il existe un GCD-arbre de preuve dont les feuilles sont $L' \subseteq L$, l'ensemble des clefs utilisées $K \subseteq L$, et la racine w, tels que pour tous les nœuds $T \vdash t$ de la preuve nous avons : $atoms(t) \subseteq atoms(L, w)$.*

Nous considérons d'abord le cas du « ou exclusif » avec un chiffrement distributif, le résultat est une conséquence immédiate du résultat sur les preuves atomiques de la section précédente. Ensuite nous regardons le cas beaucoup plus complexe de la théorie équationnelle $\mathsf{AG}\{.\}_.$, car les coefficients des termes peuvent être potentiellement infiniment grands. pour résoudre cette difficulté, nous utilisons la notion de $\mathbb{Z}$-module.

5.2.2.1 ACUN$\{.\}_.$.

Dans le cas du « ou exclusif » nous utilisons le résultat de localité atomique pour construire une procédure qui permet de calculer l'ensemble fini de toutes les preuves possibles. Par définition, les nœuds d'un GCD-arbre de preuve sont de la forme $a_1 \oplus \ldots \oplus a_n$ où $a_i \in \mathrm{atoms}(L, w)$ et $a_i \neq a_j$ pour $i \neq j$. Grâce au résultat de localité atomique, il n'y a qu'un nombre exponentiel de nœuds possible (dans la taille de $|L| + |w|$), par conséquent un nombre fini de preuves possibles.

Il reste alors à prouver la déduction en une étape pour les règles (D), (C) et (GX). Seule la règle (GX) est problématique. La déduction en une étape pour la règle (GX) dans ce cas est équivalent à la résolution d'équations diophantiennes linéaires sur $\mathbb{Z}/2\mathbb{Z}[h]$, comme nous l'avons détaillée pour ACh dans la section 5.1.1.2 page 72. Cette approche est similaire à la méthode classiquement utilisée dans ce domaine [CLS03, CKRT03].

5.2.2.2 AG$\{.\}_.$.

Le raisonnement que nous avons fait sur le « ou exclusif » ne s'applique pas dans le cas des groupes abéliens car nous ne pouvons a priori borner les coefficients des éléments d'une somme de termes $\Sigma_{i=1}^{i=n}\alpha_i a_i$. Nous posons

$$\mathrm{atoms}(L, w) = \{a_1, \ldots, a_n\}$$

Un terme t est dit (L, w)-*atomique* si $\mathrm{atoms}(t) \subseteq \mathrm{atoms}(L, w)$. Grâce à la proposition 5 page 78, un terme (L, w)-atomique t peut s'écrire sous la forme $\alpha_1 a_1 + \ldots + \alpha_n a_n$ avec $\alpha_i \in \mathbb{Z}$. Nous appelons le vecteur $\overline{t} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{Z}^n$ la représentation de t et de manière naturelle $\overline{t_1 + t_2} = \overline{t_1} + \overline{t_2}$, et $\overline{-t} = -\overline{t}$, où l'addition de deux représentations s'effectue composante par composante. De plus, par définition, $\overline{a_i} = e_i$ où e_i est le i-ème vecteur unité.

Nous rappelons qu'un $\mathbb{Z}$-*module*, ou simplement un *module*, est un groupe abélien M muni d'une multiplication par scalaire telle que $\alpha(x + y) = \alpha x + \alpha y$ et $\alpha(-x) = -\alpha x$ pour tout $\alpha \in \mathbb{Z}$ and $x, y \in M$. Nous travaillons avec un module sur $\mathbb{Z}^n$, où la multiplication par un entier est définie de manière usuelle.

Pour $x_1, \ldots, x_m \in \mathbb{Z}^n$ nous dénotons par $\langle x_1, \ldots, x_m\rangle$ le sous-module de $\mathbb{Z}^n$ engendré par $x_1, \ldots, x_m$, *i.e.*

$$\langle x_1, \ldots, x_m\rangle = \{\alpha_1 x_1 + \ldots \alpha_m x_m \mid \alpha_i \in \mathbb{Z}\}$$

Il est décidable de savoir si $y \in \langle x_1, \ldots, x_m\rangle$ étant donné $y, x_1, \ldots, x_m$, car cette question revient à résoudre un système linéaire d'équations sur $\mathbb{Z}$.

Nous construisons pour un ensemble fini L et un terme w, un ensemble fini de générateurs pour l'ensemble des termes (L, w)-atomiques qui sont déductibles de L. Nous montrons que cette construction admet un point fixe qui est accessible en un nombre fini d'étapes. Pour cela nous avons besoin d'étendre un sous-module par un nouveau générateur, ceci uniquement si le générateur est nécessaire. Le sous-module résultant doit être minimal selon un certain ordre. Nous précisons l'ordre considéré de la manière suivante : Pour $x = (x_1, \ldots, x_n), y = (y_1, \ldots, y_n) \in \mathbb{Z}^n$ nous notons $x \trianglelefteq y$ si $|x_i| \leq |y_i|$ pour tout i, et $x \triangleleft y$ si $x \trianglelefteq y$ et $x \neq y$.

Définition 18 ($Ext((x_1, \ldots, x_n), x, l)$) *Soit $l_1 = (x_1, \ldots, x_n)$ et l_2 deux listes finies d'éléments de $\mathbb{Z}^n$, et $x \in \mathbb{Z}^n$, la relation $Ext((x_1, \ldots, x_n), x, l)$ est définie dans les cas suivants :*
- *$x \in \langle x_1, \ldots, x_n \rangle$ et $l = (x_1, \ldots, x_n)$.*
- *$x \notin \langle x_1, \ldots, x_n \rangle$ et $l = (x_1, \ldots, x_n, y)$ où y est minimal pour l'ordre $\trianglelefteq$ tel que $y \in x + \langle x_1, \ldots, x_n \rangle$.*

Remarquons que dans le deuxième cas de cette définition, il peut y avoir différents choix possibles pour y, ces choix différent uniquement sur le signe des composants.

Nous rappelons que l'arithmétique de Presburger est la théorie du premier ordre de $(\mathbb{Z}, +, >)$, qui est donc décidable. La valeur absolue de $y \in \mathbb{Z}$ est définie par la formule en utilisant z une variable libre : $(y \geq 0 \implies z = y) \wedge (y < 0 \implies z = -y)$. L'expression $x \trianglelefteq y$ est également définissable par une formule de Presburger. L'appartenance de x a $\langle x_1, \ldots, x_n \rangle$ s'exprime grâce à la formule suivante $\exists \alpha_1, \ldots, \alpha_n \ (x = \Sigma_i \alpha_i x_i)$. De plus $x \notin \langle x_1, \ldots, x_n \rangle$ est simplement la négation de la formule précédente d'appartenance. L'ensemble des éléments minimaux, selon l'ordre $\trianglelefteq$, x dans un ensemble $S \subseteq \mathbb{Z}^n$ est défini par la formule de Presburger : $x \in S \wedge \neg(\exists y \ (y \in S \wedge y \trianglelefteq x))$. En combinant toutes ces formules ensemble, nous définissons tous les l_2 tels que $Ext(l_1, x, l_2)$ pour un ensemble donné l_1 et un x. Nous pouvons donc calculer facilement une liste particulière l_2, parmi toutes celles envisageables.

Nous rappelons le lemme de Dickson [Dic13], dans ce lemme les tuples sont des tuples d'entiers naturels.

Lemme 17 (Lemme de Dickson) *Pour chaque séquence infinie $t_1, t_2, \ldots$ de n-tuples distincts d'entiers naturels il existe $i < j$ tel que $t_i \triangleleft t_j$.*

Nous montrons qu'il n'y a pas de chaîne infinie dans le relation Ext :

Lemme 18 *Soit $x_1, \ldots, x_i, \ldots \in \mathbb{Z}^n$ une suite infinie, l_0 la suite vide, et $Ext(l_i, x_{i+1}, l_{i+1})$ pour tout i. Alors il existe un i tel que $l_i = l_j$ pour tout $j \geq i$.*

Preuve : Supposons que la séquence $(l_i)_{i \in \mathbb{N}}$ ne soit pas stationnaire. Alors nous pouvons extraire une sous-suite de $(l_i)_{i \in \mathbb{N}}$ telle que chaque dernier élément possède le même signe (il n'y que 2^n possibilités). Nous choisissons alors une sous-suite de cette forme non stationnaire. Grâce au lemme de Dickson il existe l_i, l_j dans cette sous-suite telle que $i < j$ et $l_i \trianglelefteq l_j$. Nous décomposons $l_i = l'_i \cdot x'_i$ et $l_j = l'_j \cdot x'_j$, où x'_i est le dernier élément de l_i et x'_j est le dernier élément de l_j. Par construction, l_i est un préfixe de la suite l'_j, et donc $x'_i \in \langle l'_j \rangle$. Comme $x'_j \in x_j + \langle l'_j \rangle$ pour un x_j, cela signifie que $x'_j - x'_i \in x_j + \langle l'_j \rangle$ par définition de $\langle l'_j \rangle$. Or x'_i et x'_j ont le même signe par construction, donc $x'_j - x'_i$ est strictement plus petit que x'_j, selon l'ordre $\trianglelefteq$. Ce qui contredit la minimalité de x'_j dans la définition de Ext. □

Nous utilisons les $\mathbb{Z}$-modules pour représenter tous les termes que la règle (GX) peut construire dans le cas de la théorie AGh. Nous analysons plus précisément comment les générateurs d'un $\mathbb{Z}$-module sont modifiés par l'application de la règle (D_k) dans le Lemme 19 page suivante.

Lemme 19 *Soit* $g^1, \ldots, g^m \in \mathbb{Z}^n$. *Nous pouvons calculer un ensemble fini de générateurs de*

$$D_k(g^1, \ldots, g^m) = \{\bar{t} \mid \overline{\{t\}_k} \in \langle g^1, \ldots, g^m \rangle\}$$

Preuve : Premièrement nous calculons l'ensemble de générateurs de

$$K_k(g^1, \ldots, g^m) = \{\overline{\{t\}_k} \mid \overline{\{t\}_k} \in \langle g^1, \ldots, g^m \rangle\}$$

Soit I_k l'ensemble des indices correspondants aux atomes chiffrés par la clef k, *i.e.*

$$I_k = \{i \mid \exists j \text{ tel que } a_i = \{a_j\}_k\}$$

où $(a_1, \ldots, a_n)$ est une énumération des éléments de $\text{atoms}(L, w)$ choisis au début de la section. Un terme (L, w)-atomique t, dont la représentation $\bar{t}$ est $(\alpha_1, \ldots, \alpha_n)$, est de la forme $\{u\}_k$ si et seulement si $\alpha_i = 0$ pour chaque $i \notin I_k$. Par conséquent, $(\alpha_1, \ldots, \alpha_n) \in K_k(g^1, \ldots, g^m)$ si et seulement si

$$\begin{cases} (\alpha_1, \ldots, \alpha_n) = \beta_1 g^1 + \ldots + \beta_m g^m \\ \text{et } \alpha_i = \beta_1 g_i^1 + \ldots \beta_m g_i^m = 0 \text{ pour chaque } i \in I_k \end{cases}$$

L'ensemble des m-tuples $(\beta_1, \ldots, \beta_m)$ satisfaisant ce système d'équations constitue bien un sous-module de $\mathbb{Z}^m$. Nous calculons un ensemble fini B de générateurs de ce sous-module : Chaque β_i s'écrit $\gamma_i - \delta_i$ où $\gamma_i, \delta_i \geq 0$. Donc les équations $\beta_1 g_i^1 + \ldots \beta_m g_i^m = 0$ pour $i \in I_k$ définissent un système homogène (H) où les inconnus γ_i, δ_i appartiennent à $\mathbb{N}$. Un $2m$-tuple $(\gamma_1, \ldots, \gamma_m, \delta_1, \ldots, \delta_m)$ est une solution de (H) si et seulement si c'est une combinaison linéaire avec coefficients dans $\mathbb{N}$ de solutions minimales de (H). Toute combinaison linéaire à coefficients dans $\mathbb{Z}$ est aussi une solution de (H). Toute solution β s'exprime comme une combinaison linéaire à coefficients dans $\mathbb{Z}$ d'éléments de l'ensemble fini

$$B = \{(\gamma_1^\mu - \delta_1^\mu, \ldots, \gamma_m^\mu - \delta_m^\mu) \mid (\gamma_1^\mu, \ldots, \gamma_m^\mu, \delta_1^\mu, \ldots, \delta_m^\mu) \text{ solution minimale de (H)}\}$$

Réciproquement toute combinaison linéaire à coefficients dans $\mathbb{Z}$ d'éléments de B est une solution de l'ensemble original d'équations. Ainsi B est un ensemble fini de générateurs.

Nous obtenons

$$G = \{b_1 g^1 + \ldots + b_m g^m \mid (b_1, \ldots, b_m) \in B\}$$

est une ensemble fini de générateurs de $K_k(g^1, \ldots, g^m)$.

Nous obtenons donc un ensemble fini de générateurs de $D_k(g^1, \ldots, g^m)$ en « décalant » les éléments de G. Soit $\mathit{shift}_k(\alpha_1, \ldots, \alpha_n)$ un vecteur $(\beta_1, \ldots, \beta_n)$ défini par

$$\beta_i = \begin{cases} \alpha_j & \text{si } \{a_i\}_k = a_j \in \text{atoms}(L, w) \\ 0 & \text{si } \{a_i\}_k \notin \text{atoms}(L, w) \end{cases}$$

L'ensemble fini de générateurs de $D_k(g^1, \ldots, g^m)$ est donc $\mathit{shift}_k(G)$. $\square$

Le lemme 20 correspond à l'analogue du lemme 19 pour la règle (D_k).

Lemme 20 *Soit* $g^1, \ldots, g^m \in \mathbb{Z}^n$. *nous pouvons calculer un ensemble fini de générateurs de :*

$$C_k(g^1, \ldots, g^m) = \{\overline{\{t\}_k} \mid \bar{t} \in \langle g^1, \ldots, g^m \rangle, \mathit{atoms}(\{t\}_k) \subseteq \mathit{atoms}(L, w)\}$$

Preuve : La preuve fonctionne de la même manière que celle du lemme 19. Maintenant nous construisons l'ensemble fini B de générateurs de l'ensemble

$$K_k^{-1}(g^1, \ldots, g^m) = \{\bar{t} \mid \bar{t} \in \langle g^1, \ldots, g^m \rangle \text{ et } \text{atoms}(\{t\}_k) \subseteq \text{atoms}(L, w)\}$$

Cela représente l'ensemble des termes dont le chiffrement avec k est encore un terme (L,w)-atomique. L'ensemble fini des générateurs de $C_k(g^1,\ldots,g^m)$ est obtenu par $\mathit{shift}_k^{-1}(B)$, où $\mathit{shift}_k^{-1}(\alpha_1,\ldots,\alpha_n)$ est le vecteur $(\beta_1,\ldots,\beta_n)$ défini par

$$\beta_i = \begin{cases} \alpha_j & \text{si } a_i = \{a_j\}_k \text{ avec } a_j \in \mathrm{atoms}(L,w) \\ 0 & \text{si } a_i \text{ n'est pas de la forme } \{a_j\}_k \text{ où } a_j \in \mathrm{atoms}(L,w) \end{cases}$$

□

Lemme 21 *Nous pouvons calculer, étant donné un ensemble fini de termes L et un terme w, un ensemble fini de générateurs de l'ensemble des termes (L,w)-atomiques qui sont déductibles avec la règle $(GXCD)$ de L.*

Preuve : Soit $\mathrm{atoms}(L,w) = \{a_1,\ldots,a_n\}$ et $L = \{t_1,\ldots,t_p\}$, nous définissons une relation Φ entre des listes finies de vecteurs de l manière suivante :

Étant donné un ensemble fini de vecteurs l, soit $x_1,\ldots,x_p$ l'union des ensembles finis de générateurs de $D_k(l)$ avec $k \in L$ calculés par le lemme 19 page ci-contre, et $x_{p+1},\ldots,x_q$ l'union des ensembles finis de générateurs de $C_k(l)$ avec $k \in L$ issus du lemme 20 page précédente, alors $\Phi(l,l')$ existe si et seulement s'il existe des listes finies de vecteurs $l_0,\ldots,l_q$ telles que $l = l_0$, $l' = l_q$, et $Ext(l_{i-1},x_i,l_i)$ pour tout i.

Soit $l_0 = (\overline{t_p},\ldots,\overline{t_p})$, et $\Phi(l_i,l_{i+1})$ pour tout i. Par le lemme 18 page 85, il existe un i tel que $l_{i+1} = l_i$. Par construction ce l_i est un ensemble fini de générateurs de l'ensemble des termes (L,w)-atomiques qui sont atomiquement déductibles à partir de L. □

5.2.3 Cas binaire.

Nous utilisons un résultat de réécriture préfixe pour obtenir un résultat en temps polynomial dans le cas d'une preuve binaire. Nous définissons d'abord une preuve binaire pour le cas $\mathsf{AG}\{.\}.$, puis nous montrons qu'une preuve de $T \vdash w$ où tous les éléments de T et w sont binaires alors il existe une preuve de $T \vdash w$ où tous les nœuds sont binaires. Pour conclure nous utilisons la réécriture préfixe pour montrer la déduction en une étape de la règle (GCD).

5.2.3.1 Définitions.

Définition 19 (Terme binaire en tête et terme au plus binaire) *Soit t un terme en forme normale, t est* binaire en tête *si t est la différence de deux termes distincts non en tête avec $+$ ni avec $-$. Un terme est* au plus binaire *si tous ces sous-termes syntaxiques sont soit binaires en tête, soit ni en tête avec $+$ ni avec $-$. Un ensemble de termes est* au plus binaire *si chacun de ces éléments l'est. Un arbre de preuve P de $T \vdash u$ est au plus binaire si tous ses nœuds sont au plus binaires.*

Exemple 12 *le terme $a - b$ est au plus binaire par contre les termes $2a$ et $3a - 2b$ ne le sont pas.*

Remarquons que si un ensemble de terme T est au plus binaire alors l'ensemble $S(T)$ des sous-termes syntaxiques l'est également.

Nous proposons une procédure de décision en temps polynomial pour le problème de déduction de l'intrus lorsque l'ensemble des hypothèses et la conclusion sont des termes au plus binaire.

5.2.3.2 Preuves au plus binaires.

Nous montrons comment transformer une preuve dont les hypothèses et la conclusion sont au plus binaires en une preuve au plus binaire (preuve où tous les nœuds sont au plus binaires).

Exemple 13 *La preuve suivante de* $\{c\}_k - \{b\}_k$ *où* $L = \{a - b, c - d, \{d\}_k - \{a\}_k\}$ *et* $K = \{k\}$ *n'est pas au plus binaire.*

$$\dfrac{\dfrac{\dfrac{\dfrac{a-b \in T}{T \vdash a-b}(A) \quad \dfrac{c-d \in T}{T \vdash c-d}(A)}{T \vdash a-b+c-d}(GX) \quad \dfrac{k \in T}{T \vdash k}(A)}{T \vdash \{a\}_k - \{b\}_k + \{c\}_k - \{d\}_k}(C) \quad \dfrac{\{d\}_k - \{a\}_k \in T}{T \vdash \{d\}_k - \{a\}_k}(A)}{T \vdash \{c\}_k - \{b\}_k}(GX)$$

Nous transformons cette preuve en une preuve au plus binaire.

$$\dfrac{\dfrac{\dfrac{a-b \in T}{T \vdash a-b}(A) \quad \dfrac{k \in T}{T \vdash k}(A)}{T \vdash \{a\}_k - \{b\}_k}(C) \quad \dfrac{\dfrac{c-d \in T}{T \vdash c-d}(A) \quad \dfrac{k \in T}{T \vdash k}(A)}{T \vdash \{c\}_k - \{d\}_k}(C) \quad \dfrac{\{d\}_k - \{a\}_k \in T}{T \vdash \{d\}_k - \{a\}_k}(A)}{T \vdash \{c\}_k - \{b\}_k}(GX)$$

Intuitivement, $\langle U \rangle$ est l'ensemble de termes en forme normale à partir desquels nous pouvons déduire tous les sous-ensembles de U en appliquant uniquement la règle (GX).

Définition 20 (**$\langle U \rangle$**) *Soit* U *un ensemble de termes nous dénotons par* $\langle U \rangle$ *l'ensemble de termes tel que :*

$$\langle U \rangle = \{(\alpha_1 u_1 + \ldots + \alpha_n u_n)\downarrow \mid \alpha \in \mathcal{Z}, u_i \in U\}$$

Dans le cas $\mathsf{ACUN}\{.\}_.$, la proposition 7 est plus simple à obtenir et fonctionne sur la même idée de découpage de preuve. Ce résultat nous permet de prouver que toute preuve dont les hypothèses et la conclusion sont au plus binaires peut être transformée en une preuve au plus binaire avec les mêmes hypothèse et la même conclusion,dans le lemme 22 page ci-contre.

Proposition 7 *Soit* U *un ensemble fini de termes au plus binaires et* $u \in \langle U \rangle$*. Alors il existe des termes au plus binaires* $v_1, \ldots, v_k \in \langle U \rangle$ *où* $u \in \langle v_1, \ldots, v_k \rangle$ *et* $atoms(v_1, \ldots, v_k) \subseteq atoms(u)$*.*

Preuve : Nous effectuons une preuve par induction sur le nombre d'éléments de U. Cas de bas : Si $u = 0$ alors nous choisissons $k = 0$. Sinon il existe un $a \in \text{atoms}(u)$, et un terme $u_0 \in U$ tels que $a \in \text{atoms}(u_0)$. Soit αle facteur avec lequel u_0 apparaît dans le terme u, ainsi $u = \alpha u_0 + u'$ où $u' \in \langle U \setminus \{u_0\} \rangle$.

Par hypothèse d'induction il existe des termes au plus binaires $v'_1, \ldots, v'_l \in \langle U \setminus \{u_0\} \rangle$ tels que $u' \in \langle v'_1, \ldots, v'_l \rangle$ et $\text{atoms}(v'_1, \ldots, v'_l) \subseteq \text{atoms}(u')$.

Nous distinguons trois cas :

1. u_0 n'est pas en tête avec $+$, ainsi $u_0 = a$. Nous prenons $k = l + 1$, $v_i = v'_i$ pour $i < k$, et $v_k = u_0$. Or $u_i \in \langle U \setminus \{u_0\} \rangle$ pour $i < k$ nous obtenons $u_i \in \langle U \rangle$ pour $i < k$, et $v_k \in \langle U \rangle$ car $u_0 \in U$.

   Par construction, $u \in \langle v_1, \ldots, v_k \rangle$, et comme $a \in \text{atoms}(u)$ nous avons $\text{atoms}(u) = \text{atoms}(u') \cup \{a\}$. Par conséquent, $\text{atoms}(v_i) \subseteq \text{atoms}(u') \subseteq \text{atoms}(u)$ pour $i < k$ par hypothèse d'induction, et $\text{atoms}(v_k) = \{a\} \subseteq \text{atoms}(u)$ par choix de a.

2. $u_0 = a - b$ ou $u_0 = b - a$ pour $b \in \mathrm{atoms}(u)$. Ce cas est semblable au premier cas : nous choisissons $k = l + 1$, $v_i = v'_i$ pour $i < k$, et $v_k = u_0$. Comme précédemment, $u_i \in \langle U \rangle$ pour $i \leq k$, ainsi nous obtenons $u \in \langle v_1, \ldots, v_k \rangle$. Concernant les atomes de u nous avons $\mathrm{atoms}(u) = \mathrm{atoms}(u') \cup \{a, b\}$ et donc $\mathrm{atoms}(v_1, \ldots, v_k) \subseteq \mathrm{atoms}(u)$.

3. $u_0 = a - b$ ou $u_0 = b - a$ pour un atome b, et $b \notin \mathrm{atoms}(u)$. Nous supposons sans perdre de généralité que $u_0 = a - b$.

   Chaque v'_i avec $1 \leq i \leq l$ peut s'écrire $v'_i = \lambda_i a + \gamma_i b + v''_i$ où $a, b \notin \mathrm{atoms}(v''_i)$. Nous choisissons $k = l$, et $v_i = (\lambda_i + \gamma_i)a + v''_i$ pour $1 \leq i \leq k$.

   Pour tout i il existe deux atomes au plus binaires dans v''_i car v'_i est au plus binaire.
   – Si v''_i contient deux atomes alors $\lambda_i = \gamma_i = 0$, et $v_i = v'_i$ est au plus binaire.
   – Sinon, si v''_i contient un atome au plus binaire, alors v est au plus binaire car $b \notin \mathrm{atoms}(v_i)$.
   Pour tout i, $v'_i + \gamma_i u_0 = (\lambda_i + \gamma_i)a + v''_i = v_i$, et donc $v_i \in \langle U \rangle$.

   Par construction, $\mathrm{atoms}(v_i) \subseteq \mathrm{atoms}(v'_i) \cup \{a\} \setminus \{b\} \subseteq \mathrm{atoms}(u)$.

   Nous montrons donc que $u \in \langle v_1, \ldots, v_k \rangle$. Soit $u' = \Sigma_{i=1}^{i=n} \alpha_i v'_i$. Remarquons d'abord que comme $u = \alpha u_0 + \Sigma_{i=1}^{i=n} \alpha_i v'_i = \alpha a - \alpha b + \Sigma_{i=1}^{i=n} (\lambda_i a + \gamma_i b + v''_i)$, $b \notin \mathrm{atoms}(u, v''_1, \ldots, v''_n)$ et $\alpha = \Sigma_{i=1}^{i=n} \alpha_i \gamma_i$, nous obtenons donc :

$$\Sigma_{i=1}^{i=n} \alpha_i v_i = \Sigma_{i=1}^{i=n} \alpha_i (v'_i + \gamma_i u_0) = \Sigma_{i=1}^{i=n} \alpha_i v'_i + (\Sigma_{i=1}^{i=n} \alpha_i \gamma_i) u_0 = u' + \alpha u_0 = u$$

□

Lemme 22 *Soit P un GCD-arbre de preuve avec l'ensemble de feuilles L, l'ensemble de clefs K, et la racine r. Si L et r sont au plus binaires alors il existe un GCD-arbre de preuve au plus binaire P' avec ensemble de feuilles $L' \subseteq L$, ensemble de $K' \subseteq K$, et racine r.*

Preuve : Remarquons d'abord que toutes instances de la règle (C) ou (D), peuvent être vues comme un cas spécial d'un GCD-arbre de preuve, la racine est au plus binaire si et seulement si les feuilles sont au plus binaires. Par conséquent, si tous les résultats des instances de la règle (GX) dans une preuve P sont au plus binaires alors P est au plus binaire, en supposant que l'ensemble de feuilles L, l'ensemble de clefs K, et la racine r soient au plus binaires.

Sinon, il existe une instance de la règle (GX) dont le résultat n'est pas au plus binaire et où toutes les feuilles sont au plus binaires. Comme la racine de P est au plus binaire para hypothèse, le chemin de la racine de P à la conclusion de la règle (GX) non au plus binaire passe forcément par une autre règle (GX). Ainsi la preuve est de la forme suivante :

$$\text{(GX)}\ \dfrac{\dfrac{\dfrac{\text{(GX)}\ \dfrac{T \vdash u_1 \qquad \cdots \qquad T \vdash u_n}{\text{(C,D)}\ \dfrac{T \vdash u \quad (\in \langle u_1, \ldots, u_n \rangle)}{\vdots}}}{}}{\text{(C,D)}\ \dfrac{}{T \vdash u'}} \qquad P_1 \qquad \cdots \qquad P_n}{P}$$

où (C, D) est n'importe quelle instance de règle (C) ou (D), et où les clefs sont omises pour plus de clarté. Grâce à la proposition 7 page précédente il existe des termes au plus binaires $v_1, \ldots, v_k \in \langle u_1, \ldots, u_n \rangle$ tels que $u \in \langle v_1, \ldots, v_k \rangle$ et $\mathrm{atoms}(v_1, \ldots, v_k) \subseteq \mathrm{atoms}(u)$. Par conséquent, nous obtenons une preuve de la forme suivante, où nous notons par $T \vdash U$ l'ensemble des séquents

$\{T \vdash z \mid z \in U\}$:

$$
\text{(GX)}\ \dfrac{\text{(C,D)}\ \dfrac{\text{(C,D)}\ \dfrac{\text{(GX)}\ \dfrac{\text{(GX)}\ \dfrac{T \vdash U}{T \vdash v_1} \quad \cdots \quad \text{(GX)}\ \dfrac{T \vdash U}{T \vdash v_n}}{T \vdash u}\ (\in \langle v_1, \ldots, v_n \rangle)}{\vdots}}{T \vdash u'} \qquad P_1 \quad \cdots \quad P_n}{P}
$$

Dans cette arbre de preuve nous savons que, pour tout i, $v_i \in \langle U \rangle$ est un terme au plus binaire. Nous pouvons donc commuter la règle (GX) avec la règle (C) suivante, car il est toujours possible de faire un chiffrement avant une règle (GX). Comme $\text{atoms}(v_i) \subseteq \text{atoms}(u)$ pour $1 \leq i \leq k$ nous pouvons aussi commuter cette application de la règle (GX) avec la règle (D) suivante. Ainsi nous obtenons une preuve de la forme suivante :

$$
\text{(GX)}\ \dfrac{\text{(C,D)}\ \dfrac{\text{(C,D)}\ \dfrac{\text{(GX)}\ \dfrac{T \vdash U}{T \vdash v_1}}{\vdots}}{T \vdash u'_1} \qquad \text{(C,D)}\ \dfrac{\text{(C,D)}\ \dfrac{\text{(GX)}\ \dfrac{T \vdash U}{T \vdash v_n}}{\vdots}}{T \vdash u'_n} \qquad P_1 \quad \cdots \quad P_n}{P}
$$

où $u' \in \langle u'_1, \ldots, u'_n \rangle$. Nous concluons en appliquant l'hypothèse d'induction à cet arbre de preuve, car le nombre d'applications de la règle (GX) a diminué et tous les nouveaux nœuds sont au plus binaires. $\square$

Nous définissons la règle (GCD) comme une instance des règles (GX), (C) et (D) cette forme.

Définition 21 (Règle (GCD)) *La règle (GCD) contient exactement une instance de (GX), où toutes les instances de (C) sont au-dessus de la règle (GX), et toutes les instances de la règle (D) sont au-dessous de la règle (GX).*

Définition 22 ($S_{\oplus 2}(T)$) *Soit T un ensemble de termes en forme normale, nous définissons $S_{\oplus 2}(T)$ par :*

$$
S_{\oplus 2}(T) = \mathit{atoms}(T) \cup \{a_1 - a_2 \mid a_1, a_2 \in \mathit{atoms}(T), a_1 \neq a_2\}
$$

Remarquons que $S_{\oplus 2}(T)$ se calcule en temps polynômial en T..Nous prouvons maintenant une résultat de localité pour les preuves représentées par un GCD-arbre de preuve.

Lemme 23 *Soit P un GCD-arbre de preuve avec comme ensemble de feuilles L, de clefs K, et comme racine r. Si $L \cup \{r\} \in S_{\oplus 2}(T)$ pour un ensemble de termes T alors il existe un arbre preuve tel que tous les nœuds soient dans $S_{\oplus 2}(T)$.*

Preuve : Le lemme 13 page 81 assure l'existence d'une preuve simple et $\oplus$-*eager*. Nous appliquons alors la transformation de preuve décrite dans le lemme 22 page précédente, ainsi tous les nœuds de la preuve sont donc au plus binaires. Comme P est simple, la seule possibilité entre deux règles (GX) est de trouver un séquence de règles (D) suivie d'une séquence de règles (C). La frontière entre deux instances de la règle (GCD) se termine forcément par une séquence d'applications de la règle (D). Comme $r \in S_{\oplus 2}(T)$ par hypothèse, nous savons donc que $v_i \in S_{\oplus 2}(T)$ car pour tout

i, $atoms(v_)$ $\subseteq$ atoms(u) et v_i sont des termes au plus binaires. Pas conséquent, tous termes issus d'une séquence de déchiffrement à partir de v_i sont donc dans $S_{\oplus 2}(T)$, ainsi tous les nœuds de la preuve sont dans $S_{\oplus 2}(T)$. $\square$

Il ne reste plus qu'a montrer la déductibilité en une étape pour cette règle.

5.2.3.3 Réécriture préfixe pour la déductibilité en une étape de (GCD).

Dans cette section nous noterons, pour plus de clarté, une séquence de chiffrements par $\{m\}_{x_1 \cdots x_n}$ à la place de $\{\cdots \{m\}_{x_1} \cdots\}_{x_n}$.

Nous devons maintenant décider la déductibilité en une étape de la règle (GCD), définition 21 page ci-contre : Étant donné un ensemble U de termes au plus binaires et un terme au plus binaire r, existe-t-il une règle (GCD) avec les feuilles et les clefs contenues dans U et la racine r ? Nous distinguons donc trois cas possibles :

1. r n'est pas en tête avec $+$, et il existe une suite de termes au plus binaires $(\{a_i\}_{v_i} - \{b_i\}_{v_i})_{i=1,\ldots,N}$ telle que, pour tout i soit $\{a_i\}_{v_i} - \{b_i\}_{v_i}$ soit $\{b_i\}_{v_i} - \{a_i\}_{v_i}$ est dans U, un terme $a_{N+1} \in U$ non en tête avec $+$, et une suite $(h_i)_{i=0,\ldots,N+1}$ de mots de U^* telle que $\{r\}_{h_0} = \{a_1\}_{v_1 h_1}$ et $\{b_i\}_{v_i h_i} = \{a_{i+1}\}_{v_{i+1} h_{i+1}}$ pour tout $i = 1, \ldots, N$.
2. r est un terme au plus binaire $r_1 - r_2$, et il existe deux instances de la règle (GCD) comme dans le premier cas avec racines r_1, resp. r_2, et avec la même suite de clefs h_i.
3. r est un terme au plus binaire $r_1 - r_2$, et il existe une suite de termes au plus binaires $(\{a_i\}_{v_i} - \{b_i\}_{v_i})_{i=1,\ldots,N}$ telle que pour tout i soit $\{a_i\}_{v_i} - \{b_i\}_{v_i}$ soit $\{b_i\}_{v_i} - \{a_i\}_{v_i}$ est dans U, et une suite $(h_i)_{i=0,\ldots,N}$ de mots U^* telle que $\{r_1\}_{h_0} = \{a_1\}_{v_1 h_1}$, $\{b_i\}_{v_i h_i} = \{a_{i+1}\}_{v_{i+1} h_{i+1}}$ pour $i = 1, \ldots, N-1$, et $\{b_N\}_{v_N h_N} = \{r_2\}_{h_0}$.

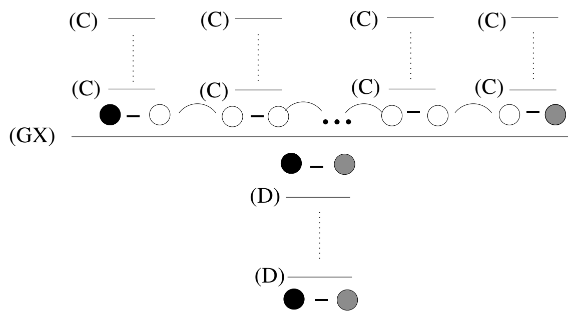


FIG. 5.7 – Illustration du troisième cas.

Nous traitons dans la suite de cette section uniquement le dernier cas, illustré dans la figure 5.7, les deux premiers cas étant semblables. Dans cette figure, la suite de clefs de chiffrement n'est pas précisée. Nous avons un terme binaire comme conclusion de la règle (GX), sur lequel une suite de règles (D) est appliquée, et une suite de termes binaires comme hypothèses de la règle (GX), dont chaque terme binaire est issu d'une séquence d'applications de la règle de chiffrement (C). Dans cette suite de termes, les atomes s'éliminent deux à deux (représentés par les arcs de la figure), ne préservant uniquement le premier et le dernier terme dans le résultat de la règle (GX).

L'idée de notre approche consiste à ramener la déduction en une étape de la règle (GCD) au problème d'accessibilité (« reachability ») dans les systèmes de réécriture préfixe, étudiés par D. Caucal [Cau92].

Les termes chiffrés comme $\{a\}_{xyz}$ seront notés $axyz$, où a n'est pas en tête ni avec un $+$, ni avec un symbole de chiffrement. Cette notation est vue comme une chaîne de caractères où le premier terme représente le message chiffré suivi de la suite de chiffrements appliqués du plus profond au plus superficiel. Ceci peut aussi être vu comme une configuration d'un automate à pile (« pushdown process ») avec a comme état et xyz comme pile, la clef la plus récente se trouve au sommet de pile.

Nous présentons d'abord la construction dans un cas simplifié. Si nous ignorons pour le moment les instances possibles de la règle (D), et si nous supposons aussi que tous les termes en position clef dans un terme de U sont aussi contenus dans U. Nous construisons alors simplement le système de réécriture préfixe qui pour tous termes binaires $\{a\}_v - \{b\}_w \in L$, réécrit tous termes avx en bwx, et réciproquement :

$$\{av \to bw \mid \{a\}_v - \{b\}_w \in U \text{ or } \{b\}_w - \{a\}_v \in U\}$$

Si nous souhaitons vérifier l'existence de la règle (GCD) avec racine $\{a\}_v - \{b\}_w$ alors nous avons juste à vérifier si la chaîne av se réécrit en bw dans le système de réécriture préfixe.

Exemple 14 *Soit $U = \{a - \{b\}_{12}, \{b\}_1 - \{c\}_3, 1, 2, 3, 4\}$, et $r = \{a\}_4 - \{c\}_{324}$ (nous utilisons ici comme dans l'exemple 15 nombres comme clefs). Il existe un GCD-arbre de preuve avec feuilles et clefs dans L et la racine r :*

$$(GX)\dfrac{(C_4)\dfrac{T \vdash a - \{b\}_{12}}{T \vdash \{a\}_4 - \{b\}_{124}} \qquad (C_4)\dfrac{(C_2)\dfrac{T \vdash \{b\}_1 - \{c\}_3}{T \vdash \{b\}_{12} - \{c\}_{32}}}{T \vdash \{b\}_{124} - \{c\}_{324}}}{T \vdash \{a\}_4 - \{c\}_{324}}$$

Le système de réécriture préfixe obtenu à partir de L est :

$$\{a \to b12, b12 \to a, b1 \to c3, c3 \to b1\}$$

Nous obtenons donc

$$a4 \mapsto b124 \mapsto c324$$

Ainsi le terme $\{a\}_4 - \{c\}_{324}$ est déductible en une étape par la règle (GCD).

La première difficulté est que certaines clefs peuvent ne pas être contenues dans U. Pour ce faire, nous réécrivons avx en bwx seulement quand $x \in U^*$, *i.e.* quand x est une suite de symboles de U. Nous ajoutons un marqueur, représenté par le symbole #, dans les termes pour se souvenir que tous les symboles au-dessous de # dans la pile sont dans U.

Formellement, soit $\mathit{left}(x)$ et $\mathit{right}(x)$, pour toute chaîne x, telle que $x = \mathit{left}(x) \cdot \mathit{right}(x)$, et $\mathit{right}(x)$ est le suffixe maximal x dans U^*. nous construisons alors le système de réécriture comme suit :

$$\begin{array}{l}\{ \quad a\ \mathit{left}(v)\#\mathit{right}(v) \to b\ \mathit{left}(w)\#\mathit{right}(w) \mid \\ \qquad\qquad \{a\}_v - \{b\}_w \in U \text{ or } \{b\}_w - \{a\}_v \in U \quad \}\end{array}$$

Exemple 15 *Nous appliquons cette nouvelle construction sur l'exemple 14 où nous ôtons la clef 1 de l'ensemble U. Nous obtenons le même (GCD)-arbre de preuve que dans l'exemple 14 car le chiffrement par 1 n'est pas utilisé dans cette preuve. Le système de réécriture devient donc :*

$$\{a\# \to b1\#2, b1\#2 \to a, b1\# \to c\#3, c\#3 \to b1\#\}$$

Nous pouvons alors réécrire $a\#4$ en $c\#324$ par la séquence suivante :

$$a\#4 \mapsto b1\#24 \mapsto c\#324$$

Il existe bien une preuve du terme $\{a\}_4 - \{c\}_{324}$ en une étape avec la règle (GCD). Par contre, si nous supprimons la clef 2 de U le système de réécriture devient :

$$\{a\# \to b12\#, b12\# \to a, b1\# \to c\#3, c\#3 \to b1\#\}$$

Nous ne pouvons pas avec ce système réécrire $a\#4$ en $c\#324$ il n'est donc pas possible de déduire en une étape (GCD) le terme $\{a\}_4 - \{c\}_{324}$.

Considérons maintenant le cas général où le résultat d'une règle (GCD) est issu d'une séquence de règles (D) après la règle (GX). Nous séparons le processus de réécriture en deux processus consécutifs. Lors de la première étape nous avons une pile x et souhaitons appliquer une règle dont le côté gauche contient x comme préfixe alors nous inscrivons le symbole $\overline{a}$ sur la pile, si le symbole a est manquant. Cette notation permet de distinguer clairement les deux étapes disjointes, notons que si $x = x_1 \cdots x_n$ alors $\overline{x} = \overline{x_n} \cdots \overline{x_1}$. Dans le second processus, nous effectuons l'opération inverse ; si nous rencontrons un symbole négatif au sommet de pile et si un côté droit d'une règle de réécriture produit ces symboles, alors nous dépilons ces symboles négatifs du sommet de la pile. Le symbole $\perp$ dénote la fin d'une chaîne (*i.e.* , la fin de la pile).

Définition 23 (*sta* et *keys*) *Nous définissons les fonctions sta et keys telles que :*
- $sta(\{t\}_k) = sta(t)$,
- $sta(t) = \bigcup_{a \in atoms(t)} sta(a)$,
- $sta(t) = \{t\}$ *si t n'est pas en tête avec $+$ et ni de la forme $\{x\}_y$.*
- $keys(\{t\}_k) = keys(t) \cup \{k\}$
- $keys(t) = \bigcup_{a \in atoms(t)} keys(a)$,
- $keys(t) = \emptyset$*si t n'est pas en tête avec $+$ et ni de la forme $\{x\}_y$.*

Nous étendons ces notions aux ensembles de termes T par $sta(T) = \bigcup_{t \in T} sta(t)$ et $keys(T) = \bigcup_{t \in T} keys(t)$.

Exemple 16 *Soit $T = \{\{a\}_{bc} - \{d\}_e, \{d\}_{ce}\}$, alors $sta(T) = \{a, d\}$ et $keys(T) = \{b, c, e\}$.*

Soit U un ensemble de termes au plus binaires et un terme r au plus binaire, nous définissons deux systèmes de réécriture préfixe. Soit $Q = sta(U, r)$ et $C = keys(U, r)$.

1. Le système de réécriture préfixe PR_1 est défini par :

$$\begin{aligned}\{\quad & a\ left(v)\#right(v) \to b\ left(w)\#right(w)\\ & a\ left(v)\#v_1\gamma \to b\ left(w)\#right(w)\overline{v_2}\gamma \mid\\ &\quad \{a\}_v - \{b\}_w \in U \text{ or } \{b\}_w - \{a\}_v \in U,\\ &\quad v_1 v_2 = right(v),\\ &\quad \gamma \in \{\perp\} \cup \{\overline{u} \mid u \in U\}\ \}\end{aligned}$$

2. Le système de réécriture préfixe PR_2 est défini par :

$$\begin{aligned}\{\quad & \hat{a}\ left(v)\#right(v) \to \hat{b}\ left(w)\#right(w)\\ & \hat{a}\ left(v)\#right(v)\overline{w_2} \to \hat{b}\ left(w)\#w_1 \mid\\ &\quad \{a\}_v - \{b\}_w \in U \text{ or } \{b\}_w - \{a\}_v \in U,\\ &\quad w_1 w_2 = right(w) \quad \}\end{aligned}$$

Ces deux systèmes de réécriture sont symétriques l'un de l'autre, au détail prés que le symbole γ dans le système PR_1 assure qu'aucun symbole négatif n'apparaît à gauche d'un symbole non négatif. Le système PR_2 préserve cet invariant car il ne peut pousser de symbole négatif. Remarquons que nous avons dans le premier cas une règle de réécriture pour toutes décompositions de $right(v)$ en v_1 et v_2, et dans le second cas, une règle de réécriture pour toutes décompositions de $right(w)$ en w_1 et w_2.

Nous définissons donc le système de réécriture complet de la manière suivante :

$$PR_1 \cup PR_2 \cup \{a \to \hat{a} \mid a \in Q\}$$

Exemple 17 *Soit* $U = \{\{a\}_{12} - b, \{b\}_{34} - c, c - \{d\}_{234}, 1, 2, 3, 4\}$, *et* $r = a - \{d\}_1$.*Nous donnons uniquement les règles de réécriture pertinentes pour cet exemple : Le système* PR_1 *contient parmi ses règles, la règle* $a\#\bot \to b\#\overline{21}\bot$ *et la règle* $b\#\overline{2} \to c\#\overline{432}$. *Le système* PR_2 *contient la règle* $\hat{c}\#\overline{432} \to \hat{d}\#$. *Par conséquent, nous obtenons la séquence de réécriture suivante :*

$$a\#\bot \mapsto b\#\overline{21}\bot \mapsto c\#\overline{4321}\bot \to \hat{c}\#\overline{4321}\bot \to \hat{c}\#\overline{1}\bot$$

Définition 24 (Chaîne admissible) *Étant donné* Q, C, U, *une chaîne est* admissible *si elle est de la forme* $qx_1\#x_2\overline{x_3}\bot$ *où*
- *il existe un* $a \in Q$ *tel que* $q = a$ *ou* $q = \hat{a}$
- *soit* $x_1 = \epsilon$ *ou* $x_1 \in C^*(C \setminus U)$
- $x_2, x_3 \in U^*$

Proposition 8 *Le système de réécriture préfixe de la section 5.2.3 page 87 réécrit les chaînes admissibles en chaînes admissibles..*

Preuve : Cette proposition découle immédiatement de la définition de chaîne admissible et du système de réécriture préfixe. □

La proposition suivante liste les propriétés simples de l'inversion et de la décomposition de chaîne, nous les utilisons sans y faire référence dans la suite.

Proposition 9 *Pour tout* $x, y \in (C \cup U)^*$ *:*

1. $\overline{xy} = \overline{y}\ \overline{x}$
2. *Si* $y \in U^*$ *alors* $left(xy) = left(x)$ *et* $right(xy) = right(x)y$

Les deux lemmes suivants montrent la propriété centrale de ces deux systèmes de réécriture.

Lemme 24 *Les assertions suivantes sont équivalentes pour tout* $a, b \in Q$, $x_1, y_1 \in \{\epsilon\} \cup C^*(C \setminus U)$, $x_2, y_2, y_3 \in U^*$ *:*

1. *Il y a une séquence de réécriture préfixe par* PR_1

$$ax_1\#x_2\bot \mapsto^* by_1\#y_2\overline{y_3}\bot$$

2. *soit* $a = b$, $x_1 = y_1$, $x_2 = y_2$, $y_3 = \epsilon$,
   soit il existe une séquence de termes binaires $\{a_i\}_{v_i} - \{b_i\}_{w_i} \in U$, $i = 1, \ldots, n$, *et une séquence de chaîne* $h_i \in U^*$, $i = 1, \ldots, n$, *telle que*
   (a) $\{a\}_{x_1x_2y_3} = \{a_1\}_{v_1h_1}$
   (b) $\{b_i\}_{w_ih_i} = \{a_{i+1}\}_{v_{i+1}h_{i+1}}$ *pour* $i = 1, \ldots, n-1$
   (c) $\{b_n\}_{w_nh_n} = \{b\}_{y_1y_2}$

et pour un i le plus long suffixe commun entre y_3 et h_i soit ϵ.

Lemme 25 *Les assertions suivantes sont équivalentes pour tout $a, b \in Q$, $x_1, y_1 \in \{\epsilon\} \cup C^*(C \setminus U)$, $x_2, y_2, y_3 \in U^*$:*

1. *Il y a une séquence de réécriture préfixe par PR_2*

$$\hat{a}x_1\#x_2\overline{x_3}\bot \mapsto^* \hat{b}y_1\#y_2\bot$$

2. *soit $a = b$, $x_1 = y_1$, $x_2 = y_2$, $y_3 = \epsilon$,*
   soit il existe une séquence de termes binaires $\{a_i\}_{v_i} - \{b_i\}_{w_i} \in U$, $i = 1, \ldots, n$, et une séquence de chaîne $h_i \in U^*$, $i = 1, \ldots, n$, telle que
   (a) $\{a\}_{x_1x_2} = \{a_1\}_{v_1h_1}$
   (b) $\{b_i\}_{w_ih_i} = \{a_{i+1}\}_{v_{i+1}h_{i+1}}$ *pour $i = 1, \ldots, n-1$*
   (c) $\{b_n\}_{w_nh_n} = \{b\}_{y_1y_2x_3}$
   et pour un i le plus long suffixe commun entre y_3 et h_i soit ϵ.

Preuve : Remarquons d'abord que ces deux systèmes de réécriture préfixe PR_1 et PR_2 sont symétriques (Le seul but des occurrences de γ dans PR_1 est de garantir l'admissibilité de toutes les configurations accessibles). Par conséquent, nous prouvons uniquement le lemme 24 page précédente, correspondant au système de réécriture PR_1.

Pour prouver (1) implique (2), nous raisonnons par induction sur la longueur de la séquence de règle de réécriture. Si la longueur est nulle alors $a = b$, $x_1 = y_1$, $x_2 = y_2$, et $y_3 = \epsilon$. S'il y a exactement une étape de réécriture il y a deux cas possibles :

1. La règle de réécriture est de la forme :

$$a\ \mathit{left}(r)\#\mathit{right}(r) \to b\ \mathit{left}(s)\#\mathit{right}(s)$$

alors il existe un u tel que

$$\begin{array}{ll} x_1 = \mathit{left}(r) & y_1 = \mathit{left}(s) \\ x_2 = \mathit{right}(r)u & y_2 = \mathit{right}(s)u \\ & y_3 = \epsilon \end{array}$$

Nous concluons en choisissant :

$$\begin{array}{rcl} \{a_1\}_{v_1} \oplus \{b_1\}_{w_1} & := & \{a\}_r \oplus \{b\}_s \\ h_1 & := & u \end{array}$$

nous obtenons donc

$$\begin{array}{l} \{a\}_{x_1x_2y_3} = \{a\}_{x_1x_2} = \{a\}_{ru} = \{a_1\}_{v_1h_1} \\ \{b_1\}_{w_1h1} = \{b\}_{su} = \{b\}_{y_1y_2} \end{array}$$

2. Le système de réécriture est de la forme :

$$a\ \mathit{left}(r)\#r_1\bot \to b\ \mathit{left}(s)\#\mathit{right}(s)\overline{r_2}\bot$$

avec $\mathit{right}(r) = r_1r_2$. Nous avons alors

$$\begin{array}{ll} x_1 = \mathit{left}(r) & y_1 = \mathit{left}(s) \\ x_2 = r_1 & y_2 = \mathit{right}(s) \\ & \overline{y_3} = \overline{r_2},\ \text{donc } y_3 = r_2 \end{array}$$

Nous concluons en choisissant :

$$\begin{array}{rcl} \{a_1\}_{v_1} \oplus \{b_1\}_{w_1} & := & \{a\}_r \oplus \{b\}_s \\ h_1 & := & \epsilon \end{array}$$

nous obtenons donc

$$\begin{array}{l} \{a\}_{x_1x_2y_3} = \{a\}_r = \{a_1\}_{v_1h_1} \\ \{b_1\}_{w_1h_1} = \{b\}_s = \{b\}_{y_1y_2} \end{array}$$

Dans les deux cas, le plus long suffixe commun de h_1 et y_3 est ϵ.

S'il y a N étapes de réécriture ($N > 1$), la chaîne obtenue après $N-1$ étapes de réécriture est d'après la proposition 8 page 94 admissible. Par conséquent, il y a $b \in Q$, $y_1 \in \{\epsilon\} \cup C^*(C \setminus U)$, et $y_2, y_3 \in U^*$ tels que :

$$ax_1\#x_2\perp \to^* by_1\#y_2\overline{y_3}\perp \to cz_1\#z_2\overline{z_3}\perp$$

Par hypothèse d'induction, il existe une séquence de termes binaires $\{a_i\}_{v_i} \oplus \{b_i\}_{w_i} \in U$, $i = 1, \ldots, n$, et une suite de chaînes $h_i \in U^*$, $i = 1, \ldots, n$, telles que

1. $\{a\}_{x_1x_2y_3} = \{a_1\}_{v_1h_1}$
2. $\{b_i\}_{w_ih_i} = \{a_{i+1}\}_{v_{i+1}h_{i+1}}$ pour $i = 1, \ldots, n-1$
3. $\{b_n\}_{w_nh_n} = \{b\}_{y_1y_2}$

et le plus long suffixe commun de y_3 et d'un h_i est ϵ. Nous montrons qu'il existe $\{a_{n+1}\}_{v_{n+1}} \oplus \{b_{n+1}\}_{w_{n+1}} \in U$, et une suite de chaînes de clefs $k_i \in K^*$, $i = 1, \ldots, n+1$ tels que :

1. $\{a\}_{x_1x_2z_3} = \{a_1\}_{v_1k_1}$
2. $\{b_i\}_{w_ik_i} = \{a_{i+1}\}_{v_{i+1}k_{i+1}}$ pour $i = 1, \ldots, n$
3. $\{b_{n+1}\}_{w_{n+1}k_{n+1}} = \{c\}_{z_1z_2}$

et le plus long suffixe commun de y_3 et d'un k_j est ϵ. Il y a deux cas possibles pour la règle de réécriture utilisée à la dernière étape :

1. La règle de réécriture est de la forme :

$$b\ \mathit{left}(r)\#\mathit{right}(r) \to c\ \mathit{left}(s)\#\mathit{right}(s)$$

Alors il existe u tel que

$$\begin{array}{ll} y_1 = \mathit{left}(r) & z_1 = \mathit{left}(s) \\ y_2 = \mathit{right}(r)u & z_2 = \mathit{right}(s)u \\ & \overline{z_3} = \overline{y_3},\ \text{donc } z_3 = y_3 \end{array}$$

Nous concluons en choisissant :

$$\begin{array}{rcl} \{a_{n+1}\}_{v_{n+1}} \oplus \{b_{n+1}\}_{w_{n+1}} & := & \{b\}_r \oplus \{c\}_s \\ k_i & := & h_i \quad (i = 1, \ldots, n) \\ k_{n+1} & := & u \end{array}$$

donc

$$\begin{array}{l} \{a\}_{x_1x_2z_3} = \{a\}_{x_1x_2y_3} = \{a_1\}_{v_1h_1} = \{a_1\}_{v_1k_1} \\ \{b_i\}_{w_ik_i} = \{b_i\}_{w_ih_i} = \{a_{i+1}\}_{v_{i+1}h_{i+1}} = \{a_{i+1}\}_{v_{i+1}k_{i+1}} \quad (i = 1, \ldots, n-1) \\ \{b_n\}_{w_nk_n} = \{b\}_{y_1y_2} = \{b\}_{ru} = \{a_{n+1}\}_{v_{n+1}k_{n+1}} \\ \{b_{n+1}\}_{w_{n+1}k_{n+1}} = \{c\}_{su} = \{c\}_{z_1z_2} \end{array}$$

Si le plus long suffixe commun de y_3 et h_i, $1 \leq i \leq n$, est ϵ alors le plus long suffixe commun de $z_3 = y_3$ et $k_i = h_i$ est ϵ.

2. La règle de réécriture est de la forme :

$$b \; \mathit{left}(r)\#r_1\gamma \rightarrow c \; \mathit{left}(s)\#\mathit{right}(s)\overline{r_2}\gamma$$

avec $\mathit{right}(r) = r_1 r_2$, et $\gamma \in \{\overline{u} \mid u \in U\} \cup \{\perp\}$. Nous avons donc

$$\begin{array}{ll} y_1 = \mathit{left}(r) & z_1 = \mathit{left}(s) \\ y_2 = r_1 & z_2 = \mathit{right}(s) \\ & \overline{z_3} = \overline{r_2}\;\overline{y_3}, \text{ donc } z_3 = y_3 r_2 \end{array}$$

Nous concluons en choisissant :

$$\begin{array}{rcl} \{a_{n+1}\}_{v_{n+1}} \oplus \{b_{n+1}\}_{w_{n+1}} & := & \{b\}_r \oplus \{c\}_s \\ k_i & := & h_i r_2 \quad (i = 1, \ldots, n) \\ k_{n+1} & := & \epsilon \end{array}$$

donc

$$\begin{array}{l} \{a\}_{x_1 x_2 z_3} = \{a\}_{x_1 x_2 y_3 r_2} = \{a_1\}_{v_1 h_1 r_2} = \{a_1\}_{v_1 k_1} \\ \{b_i\}_{w_i k_i} = \{b_i\}_{w_i h_i r_2} = \{a_{i+1}\}_{v_{i+1} h_{i+1} r_2} = \{a_{i+1}\}_{v_{i+1} k_{i+1}} \quad (i = 1, \ldots, n-1) \\ \{b_n\}_{w_n k_n} = \{b_n\}_{w_n h_n r_2} = \{b\}_{y_1 y_2 r_2} = \{b\}_r = \{a_{n+1}\}_{v_{n+1} k_{n+1}} \\ \{b_{n+1}\}_{w_{n+1} k_{n+1}} = \{b_{n+1}\}_{w_{n+1}} = \{c\}_s = \{c\}_{z_1 z_2} \end{array}$$

le plus long suffixe commun de z_3 et $k_{n+1} = \epsilon$ est ϵ.

Pour l'implication de (2) vers (1), si $a = b$, $x_1 = y_1$, $x_2 = y_2$, et $y_3 = \epsilon$ alors nous avons directement que $ax_1\#x_2\perp \mapsto^* by_1\#y_2\overline{y_3}\perp$. Sinon, nous procédons par induction sur n.

Si $n = 1$ alors il existe $\{a_1\}_{v_1} \oplus \{b_1\}_{w_i} \in U$ et $h_1 \in U^*$ tels que :

1. $\{a\}_{x_1 x_2 y_3} = \{a_1\}_{v_1 h_1}$
2. $\{b_1\}_{w_1 h_1} = \{b\}_{y_1 y_2}$

et le plus long suffixe commun de y_3 et h_1 est ϵ,tel que soit $y_3 = \epsilon$ soit $h_1 = \epsilon$.

1. Si $y_3 = \epsilon$ alors $x_1\#x_2 = \mathit{left}(v_1)\#\mathit{right}(v_1)h_1$ et $y_1\#y_2 = \mathit{left}(w_1)\#\mathit{right}(w_2)h_1$, donc $ax_1\#x_2\perp \mapsto by_1\#y_2\perp$ car le terme binaire $\{a_1\}_{v_1} \oplus \{b_1\}_{w_2}$ est dans U.
2. Si $h_1 = \epsilon$ alors $x_1\#x_2 = \mathit{left}(v_1)\#v_1^1$ et $y_3 = v_1^2$ pour $\mathit{right}(v_1) = v_1^1 v_1^2$, et $y_1\#y_2 = \mathit{left}(w)\#\mathit{right}(w)$. Donc $ax_1\#x_2\perp \mapsto by_1\#y_2\overline{y_3}\perp$ car $\{a_1\}_{v_1} \oplus \{b_1\}_{w_2} \in U$.

Si $n \geq 2$ alors il existe une suite de termes binaires $\{a_i\}_{v_i} \oplus \{b_i\}_{w_i} \in U$, $i = 1, \ldots, n$, et une suite de chaînes $h_i \in U^*$, $i = 1, \ldots, n$, telles que

1. $\{a\}_{x_1 x_2 y_3} = \{a_1\}_{v_1 h_1}$
2. $\{b_i\}_{w_i h_i} = \{a_{i+1}\}_{v_{i+1} h_{i+1}}$ pour $i = 1, \ldots, n-1$
3. $\{b_n\}_{w_n h_n} = \{b\}_{y_1 y_2}$

et pour un i le plus long suffixe commun de y_3 et h_i est ϵ.

1. S'il y a $i < n$ tel que le plus long suffixe commun de y_3 et h_i est ϵ alors par induction

$$ax_1\#x_2 \mapsto^* b_{n-1} \; \mathit{left}(w_{n-1})\#\mathit{right}(w_{n-1})h_{n-1}\overline{y_3}$$

Maintenant, nous avons $b_{n-1}\mathit{left}(w_{n-1})\#\mathit{right}(w_{n-1})h_{n-1} = a_n\mathit{left}(v_n)\#\mathit{right}(v_n)h_n$ et $\{b\}_{y_1y_2} = \{b_n\}_{w_nh_n}$. Donc,

$$\begin{array}{rl} & b_{n-1}\ \mathit{left}(w_{n-1})\#\mathit{right}(w_{n-1})h_{n-1}\overline{y_3} \\ = & a_n\ \mathit{left}(v_n)\#\mathit{right}(v_n)h_n\overline{y_3} \\ \mapsto & b_n\ \mathit{left}(w_n)\#\mathit{right}(w_n)h_n\overline{y_3} \\ = & by_1\#y_2\overline{y_3} \end{array}$$

2. Sinon, le plus long suffixe commun de h_n et y_3 est ϵ. Soit s le plus long suffixe commun de y_3 et h_i pour $i < n$, et y_3', h_i' $(1 \leq i < n)$ b tel que $y_3 = y_3's$ et $h_i' = h_is$. Par conséquent, nous avons :
   (a) $\{a\}_{x_1x_2y_3'} = \{a_1\}_{v_1h_1'}$
   (b) $\{b_i\}_{w_ih_i'} = \{a_{i+1}\}_{v_{i+1}h_{i+1}'}$ pour $i = 1, \ldots, n-2$

   et pour $i < n$ le plus long suffixe commun de y_3' et h_i' est ϵ. Donc par hypothèse d'induction,

$$ax_1\#x_2 \mapsto^* b_{n-1}\ \mathit{left}(w_{n-1})\#\mathit{right}(w_{n-1})h_{n-1}'\overline{y_3}'$$

   Maintenant, nous avons $\{b_{n-1}\}_{w_{n-1}h_{n-1}} = \{a_n\}_{v_nh_n}$, c'est à dire $w_{n-1}h_{n-1}'s = v_nh_n$. Comme s est suffixe de y_3, et comme le plus long suffixe commun de y_3 et h_n est ϵ, nous concluons que $h_n = \epsilon$, et s est un suffixe de v_n. Nous décomposons $v_n = v_n^1s$ et obtenons le résultat :

$$\begin{array}{rl} & b_{n-1}\ \mathit{left}(w_{n-1})\#\mathit{right}(w_{n-1})h_{n-1}'\overline{y_3'} \\ = & a_n\ \mathit{left}(v_n)\#v_n^1\overline{y_3}' \\ \mapsto & b_n\ \mathit{left}(w_n)\#\mathit{right}(w_n)\overline{s}\overline{y_3'} \\ = & b_n\ \mathit{left}(w_n)\#\mathit{right}(w_n)h_n\overline{y_3} \\ = & by_1\#y_2\overline{y_3} \end{array}$$

□

Par conséquent, si t et s sont tous les deux de la forme $\{m\}_k$ alors il existe une preuve de $T \vdash \{t\}_v - \{s\}_w$ si et seulement si pour u, x_1, x_2, x_3 :

$$t\ \mathit{left}(v)\#\mathit{right}(v)\bot \mapsto^* u\ x_1\#x_2\overline{x_3} \mapsto \hat{u}\ x_1\#x_2\overline{x_3} \mapsto^* \hat{s}\ \mathit{left}(w)\#\mathit{right}(w)\bot$$

Lemme 26 *Soit L un ensemble de termes au plus binaires, K un ensemble de termes, et r un terme au plus binaire. Il est décidable en temps polynômial s'il existe une instance de la règle (GCD) avec pour feuilles L, clefs K, et racine r.*

Preuve : Grâce aux lemmes 24 page 94 et 25 page 95, vérifier l'existence d'une instance de la règle (GCD) se réduit à l'accessibilité dans un système de réécriture préfixe de taille polynomiale. Ceci peut être fait en temps polynômial d'après les travaux de D. Caucal [Cau92]. □

Comme conséquence immédiate de ce résultat, grâce au résultat de localité du lemme 23 page 90 et car l'ensemble $S_{\oplus 2}(T)$ se calcule en temps polynômial nous obtenons le théorème suivant :

Théorème 4 *Le problème de déduction de l'intrus dans le cas binaire pour la théorie équationnelle des groupes abéliens avec chiffrement distributif est décidable en temps polynomial*

5.3 Chiffrement homomorphique commutatif.

Nous rajoutons la propriété suivante au chiffrement $\{\{x\}_{k_1}\}_{k_2} = \{\{x\}_{k_2}\}_{k_1}$. Dans le cas général, il faut modifier légèrement le système considéré et renforcer la notion de preuve $\oplus$-*eager* pour obtenir le résultat de localité atomique. Dans le cas binaire pour un chiffrement homomorphique et commutatif pour le « ou exclusif » nous obtenons un résultat de complexité exact. Nous travaillons toujours avec le modèle de Dolev-Yao de la figure 5.3 page 77.

5.3.1 Cas général.

Dans un premier temps, nous présentons le nouveau modèle utilisé pour le cas où le symbole homomorphique est un chiffrement commutatif. Ensuite nous montrerons le résultat clef concernant la localité atomique, ceci nous permettra de conclure immédiatement en utilisant le même procédé que dans le cas non commutatif.

5.3.1.1 Nouveau modèle.

Pour prendre en compte la commutativité de la méthode de chiffrement, nous considérons que le terme $\{u\}_K$ représente le terme u chiffré par un multi-ensemble (« multi-set ») fini de clefs $K = \{k_1^{\alpha_1}, \ldots, k_n^{\alpha_n}\}$. Nous utilisons un multi-ensemble car l'ordre dans lequel les clefs sont appliquées n'a aucun importance, mais la multiplicité de chaque clef est cruciale. Notons que $\{u\}_\emptyset$ est égal au terme u. Nous modifions donc uniquement la règle de chiffrement et la règle de déchiffrement dans le modèle de Dolev-Yao étendu de la manière suivante.

$$(C_K)\frac{T \vdash u \qquad T \vdash K}{T \vdash \{u\}_K\downarrow} \qquad (D_K)\frac{T \vdash r \qquad T \vdash K}{T \vdash u\downarrow} \quad \textit{if } r =_E \{u\}_K$$

FIG. 5.8 – $K = \{k_1^{\alpha_1}, \ldots, k_n^{\alpha_n}\}$.

La règle (C_K) permet à l'intrus de chiffrer un message u par le multi-ensemble de clefs $K = \{k_1^{\alpha_1}, \ldots, k_n^{\alpha_n}\}$. Réciproquement s'il connaît le multi-ensemble de clefs utilisé dans le message chiffré il peut alors déchiffrer le message par la règle (D_K).

Exemple 18 *Considérons $T = \{k_1, k_2, \{u\}_{\{k_1^3,k_2,k_3\}}\}$ nous pouvons donc déduire le terme $\{u\}_{\{k_3\}}$.*

$$(D_K)\frac{(A)\dfrac{\{u\}_{\{k_1^3,k_2,k_3\}} \in T}{T \vdash \{u\}_{\{k_1^3,k_2,k_3\}}} \qquad (A)\dfrac{\{k_1^3,k_2\} \in T}{T \vdash \{k_1^3,k_2\}}}{T \vdash \{u\}_{\{k_3\}}}$$

Le résultat sur la localité syntaxique est encore vrai pour le chiffrement commutatif. Ainsi nous ne regardons dans la suite uniquement les règles de chiffrement, de déchiffrement et de somme et cherchons à prouver un résultat de localité pour les atomes.

5.3.1.2 Localité atomique.

Termes et sous-termes. Nous devons modifier un peu les définitions de termes et sous-termes afin de prendre en compte la commutativité des clefs.

Définition 25 (Terme en tête avec $\mathcal{E}_K$) *Un terme u en forme normale est en* tête avec $\mathcal{E}_K$ *si u est de la forme $u = \{t\}_K$. Sinon u n'est pas* en tête avec $\mathcal{E}_K$.

La notion d'atomes de la définition 11 page 72 sera utilisée dans cette section. Nous étendons la notion de sous-termes syntaxiques en sous-termes syntaxiques « commutatifs ». Nous dénotons par $\mathcal{P}[K]$ l'ensemble des partitions de l'ensemble K.

Définition 26 (Sous-termes syntaxiques commutatifs) *L'ensemble des* sous-termes syntaxiques commutatifs *d'un terme t est le plus petit ensemble $S_c(t)$ tel que :*
- $t \in S(t)$.
- *Si* $\langle u, v\rangle \in S(t)$ *alors* $u, v \in S(t)$.
- *Si* $\{u\}_K \in S(t)$ *et* $K = \{k_1^{\alpha_1}, \ldots, k_p^{\alpha_p}\}$ *alors* $u \in S(t)$ *et* $k_i \in S(t)$ *pour tout* i $1 \leq i \leq p$.
- *Si* $u = u_1 \oplus \ldots \oplus u_n \in S(t)$ *alors* $atoms(u) \subseteq S(t)$.

La définition de S_c s'étend aux ensembles de termes T en forme normale par $S_c(T) := \bigcup_{t\in T} S_c(t)$. Comme le chiffrement est commutatif le nombre de sous-termes est exponentiel en la taille de l'ensemble de clefs de T, il faut prendre en compte toutes les combinaisons possibles pour les clefs d'un terme chiffré. La notion de sous-termes syntaxiques pour le cas non commutatif est contenue dans les sous-termes syntaxiques commutatifs $S(T) \subseteq S_c(T)$, cela nous assure le résultat de localité syntaxique, ainsi nous pouvons traiter uniquement les règles (GX), (D), (C) et (A)

Exemple 19 *Si* $u = \{a\}_{k1,k2,k3}$ *alors l'ensemble des sous-termes syntaxiques commutatifs de u est :* $S_c(u) = \{u, a, k_1, k_2, k_3, \{a\}_{k1}, \{a\}_{k2}, \{a\}_{k3}, \{a\}_{k1,k2}, \{a\}_{k2,k3}, \{a\}_{k1,k3}\}$.

Différentes sortes de preuves Les différentes caractéristiques des preuves, définies précédemment, sont maintenant étendues pour tenir compte de la commutativité du chiffrement.

Définition 27 (Preuve aplatie et simple) *Soit P une preuve de $T \vdash w$. P, P est :*
- aplatie *s'il n'y a pas deux applications consécutives des règles (GX), (C) et (D).*
- simple *si :*
  1. *chaque nœud de $T \vdash v$ apparaît au plus une fois dans chaque branche de la P.*
  2. *chaque nœud de $T \vdash v$ apparaît au plus une fois comme hypothèse d'une règle (GX).*
  3. *Il n'y a pas d'application consécutive des règles (C_K) et $(D_{K'})$ si $K \cap K' \neq \emptyset$.*

Il est toujours possible de fusionner deux applications consécutives des règles (C_K), (D_K) et (GX)et d'obtenir une preuve aplatie. De plus, en éliminant certaines branches, ou certaines hypothèses, ou certaines règles ((D) ou (C)) nous pouvons toujours transformer une preuve en une preuve simple.

Proposition 10 *Soit K et K' deux ensembles de clefs tels que $K \cap K' = \emptyset$, appliquer d'abord la règle (D_K) au terme u puis la règle $(C_{K'})$ est équivalent qu'appliquer d'abord la règle $(C_{K'})$ au terme u et ensuite la règle (D_K).*

Preuve : Le fait que $K \cap K' = \emptyset$ permet d'obtenir directement le résultat. □

Cette fois, nous devons d'abord effectuer les déchiffrements le plus tôt possible dans les preuves *D-eager* et ensuite faire les sommes le plus tôt possible dans les preuves *⊕-eager*.

Définition 28 (Preuve *⊕-eager* et *D-eager*) *Soit P une preuve de $T \vdash w$, P est :*
- ***D-eager*** *si :*
  1. *il n'y a pas d'hypothèse de la règle (GX) en tête avec $\{.\}_K$ et de règle $(D_{K'})$ juste après cette application de la règle (GX) telle que $K \cap K' \neq \emptyset$,*

2. *Il n y a pas de (C) jute sous une règle (D), si c'est le cas, nous permutons en utilisant la proposition 10 page ci-contre.*

– ***$\oplus$-eager*** *si toutes les règles (C_{K_i}) immédiatement au-dessus d'une règle (GX) dans P sont telles que $K_i \cap K_j = \emptyset$ pour i, j tel que $i \neq j$.*

Transformations de preuves. Nous dénotons par π_x la sous-preuve d'une preuve P dont la racine est $T \vdash x$.

Lemme 27 *Soit P une preuve simple et aplatie de $T \vdash w$. Alors il existe une preuve P' de $T \vdash w$ telle que P' soit simple, aplatie et D-eager.*

Preuve : Soit P une preuve simple et aplatie de $T \vdash w$. Nous transformons cette preuve en une preuve simple, aplatie et D-*eager* de $T \vdash w$ par induction sur le nombre de nœuds de P. Nous regardons quelle est la dernière règle de P :

- (A) : la preuve est évidement simple, aplatie et D-*eager*.
- (GX), (C) : nous appliquons directement l'hypothèse d'induction sur toutes les sous-preuves et concluons.
- (D_{K_2}) : nous appliquons l'hypothèse d'induction sur la partie clef de la règle (D_{K_2}), et regardons plus en détails d'où provient le message chiffré de la règle (D_{K_2}) :
  - (A) : le résultat est immédiat.
  - (C) : comme la preuve est simple nous permutons donc les deux règles en utilisant la proposition 10 page précédente et appliquons l'hypothèse de récurrence.
  - (GX) :
    - si toutes les hypothèses de (GX) sont chiffrées par l'ensemble de clefs K_i tel que $K_i \cap K_2 = \emptyset$ alors nous appliquons l'hypothèse d'induction sur les sous-preuves et concluons.
    - Sinon nous partageons les hypothèses de la règle (GX) en sommes plus petites qui produisent toutes un terme chiffré et nous appliquons la transformation de la figure 5.10 page suivante. Dans certaines situations d'autres transformations sont nécessaires pour préserver la simplicité : Nous devons couper les hypothèses en double dans les règles (GX) ou des branches entières de la preuve pour les nouveaux nœuds introduits par la transformation. De plus, si la règle (GX) ne possède qu'une hypothèse, nous ôtons cette règle. Comme $K_2 \cap K_1 \neq \emptyset$ et $n \geq 2$, la taille de la preuve initiale est $\Sigma_{i=1}^{i=n}|\pi_{B_i}| + |\pi_{K_2}| + 2$, ce qui est plus grand ou égal que la taille de la sous-preuve se terminant par la règle $(D_{K_2 \cap K_1})$: $\Sigma_{i=1}^{i=n_1}|\pi_{B_i}| + |\pi_{K_2 \cap K_1}| + 2$. Nous concluons en appliquant l'hypothèse de récurrence sur cette sous-preuve.

□

$$
(GX)\frac{(GX)\dfrac{T \vdash x_1 \quad \ldots \quad T \vdash x_n}{T \vdash x_1 \oplus \ldots \oplus x_n} \qquad T \vdash y_1 \quad \ldots \quad T \vdash y_m}{T \vdash x_1 \oplus \ldots \oplus x_n \oplus y_1 \oplus \ldots \oplus y_m}
$$

$$\Downarrow$$

$$
(GX)\frac{T \vdash x_1 \quad \ldots \quad T \vdash x_n \quad T \vdash y_1 \quad \ldots \quad T \vdash y_m}{T \vdash x_1 \oplus \ldots \oplus x_n \oplus y_1 \oplus \ldots \oplus y_m}
$$

FIG. 5.9 – Fusion de deux règles (GX).

$$(D_{K_2})\dfrac{(GX)\dfrac{(R_1)\dfrac{\vdots}{T\vdash B_1}\qquad\ldots\qquad(R_n)\dfrac{\vdots}{T\vdash B_n}}{T\vdash\{u\}_{K_1}}\quad T\vdash K_2}{T\vdash\{u\}_{K_1\setminus K_2}}$$

$$\Downarrow$$

$$(D_{K_2\setminus K_1})\dfrac{(D_{K_2\cap K_1})\dfrac{(GX)\dfrac{(GX)\dfrac{(R_1)\dfrac{\vdots}{T\vdash B_1}\quad\ldots\quad(R_{n_1})\dfrac{\vdots}{T\vdash B_{n_1}}}{T\vdash\{u_1\}_{K_1}}\qquad\ldots\qquad(GX)\dfrac{(R_{n_{l-1}+1})\dfrac{\vdots}{T\vdash B_{n_{l-1}+1}}\quad\ldots\quad(R_{n_l})\dfrac{\vdots}{T\vdash B_{n_l}}}{T\vdash\{u_l\}_{K_1}}}{T\vdash\{u_1\}_{K_1}\oplus\ldots\oplus\{u_l\}_{K_1}=\{u\}_{K_1}}\quad T\vdash K_2\cap K_1}{T\vdash\{u\}_{K_1\setminus K_2\cap K_1}}\quad T\vdash K_2\setminus K_1}{T\vdash\{u\}_{(K_1\setminus K_2\cap K_1)\setminus(K_2\setminus K_1)}=\{u\}_{K_1\setminus K_2}}$$

$$\Downarrow$$

$$(D_{K_2\setminus K_1})\dfrac{(GX)\dfrac{(D_{K_2\cap K_1})\dfrac{(GX)\dfrac{(R_1)\dfrac{\vdots}{T\vdash B_1}\quad\ldots\quad(R_{n_1})\dfrac{\vdots}{T\vdash B_{n_1}}}{T\vdash\{u_1\}_{K_1}}\quad T\vdash K_2\cap K_1}{T\vdash\{u_1\}_{(K_1\setminus K_2\cap K_1)}}\qquad\ldots\qquad(D_{K_2\cap K_1})\dfrac{(GX)\dfrac{(R_{n_{l-1}+1}1)\dfrac{\vdots}{T\vdash B_{n_{l-1}+1}}\quad\ldots\quad(R_{n_l})\dfrac{\vdots}{T\vdash B_{n_l}}}{T\vdash\{u_l\}_{K_1}}\quad T\vdash K_2\cap K_1}{T\vdash\{u_1\}_{(K_1\setminus K_2\cap K_1)}}}{T\vdash\{u\}_{(K_1\setminus K_2\cap K_1)}}\quad T\vdash K_2\setminus K_1}{T\vdash\{u\}_{(K_1\setminus K_2\cap K_1)\setminus(K_2\setminus K_1)}=\{u\}_{K_1\setminus K_2}}$$

FIG. 5.10 – Transformation *D-eager* avec $K_2\cap K_1\neq\emptyset$ et $n\geq 2$.

$$(GX)\dfrac{(C_{K_1})\dfrac{T\vdash x_1 \qquad T\vdash K_1}{T\vdash \{x_1\}_{K_1}}\;(C_{K_2})\dfrac{T\vdash x_2 \qquad T\vdash K_2}{T\vdash \{x_2\}_{K_2}}\;(R_1)\dfrac{\vdots}{T\vdash z_1}\ldots(R_m)\dfrac{\vdots}{T\vdash z_m}}{T\vdash \{x_1\}_{K_1}\oplus\{x_2\}_{K_2}\oplus z_1\oplus\ldots\oplus z_m}$$

$$K_1\cap K_2\neq\emptyset$$

$$\Downarrow$$

$$(GX)\dfrac{(C_{K_1\cap K_2})\dfrac{(GX)\dfrac{(C_{K_1\setminus K_2})\dfrac{T\vdash x_1 \qquad T\vdash K_1\setminus K_2}{T\vdash \{x_1\}_{K_1\setminus K_2}}\qquad (C_{K_2\setminus K_1})\dfrac{T\vdash x_2 \qquad T\vdash K_2\setminus K_1}{T\vdash \{x_2\}_{K_2\setminus K_1}}}{T\vdash \{x_1\}_{K_1\setminus K_2}\oplus\{x_2\}_{K_2\setminus K_1}}\qquad T\vdash K_1\cap K_2}{T\vdash \{x_1\}_{K_1}\oplus\{x_2\}_{K_2}}\;(R_1)\dfrac{\vdots}{T\vdash z_1}\ldots(R_m)\dfrac{\vdots}{T\vdash z_m}}{T\vdash \{x_1\}_{K_1}\oplus\{x_2\}_{K_2}\oplus z_1\oplus\ldots\oplus z_m}$$

FIG. 5.11 – Transformation $\oplus$-*eager*, $K_1\cap K_2\neq\emptyset$ et toutes les règles (R_i) sont différentes de (C).

Proposition 11 *Les transformations de preuves des figures 5.9 page 101 et 5.11 page précédente font décroître le nombre de nœuds par rapport au nombre de nœuds de la preuve initiale.*

Preuve : Remarquons que toutes les transformations de preuves préservent les hypothèses et la conclusion de la preuve initiale. Nous considérons les transformations une par une :

- Figure 5.9 page 101 : le résultat est immédiat.
- Figure 5.11 page précédente : le nombre de nœuds de la preuve initiale est de :
  $\alpha_I = \Sigma_{i=1}^{i=m}|\pi_{z_i}| + |\pi_{x_1}| + |\pi_{x_2}| + |\pi_{K_1}| + |\pi_{K_2}| + 3$
  Le nombre de nœuds de la preuve après transformation vaut :
  $\alpha_T = \Sigma_{i=1}^{i=m}|\pi_{z_i}| + |\pi_{x_1}| + |\pi_{x_2}| + |\pi_{K_1\setminus K_2}| + |\pi_{K_2\setminus K_1}| + |\pi_{K_1\cap K_2}| + 5$
  Observons que $|\pi_{K_1}| = |\pi_{K_1\cap K_2}| + |\pi_{K_1\setminus K_2}|$ et $|\pi_{K_2}| = |\pi_{K_1\cap K_2}| + |\pi_{K_2\setminus K_1}|$.

$$\begin{aligned}
\alpha_I - \alpha_T &= |\pi_{K_1}| + |\pi_{K_2}| - |\pi_{K_1\setminus K_2}| - |\pi_{K_2\setminus K_1}| - |\pi_{K_1\cap K_2}| - 2\\
&= |\pi_{K_1\cap K_2}| + |\pi_{K_1\setminus K_2}| + |\pi_{K_2}| - |\pi_{K_1\setminus K_2}| - |\pi_{K_2\setminus K_1}| - |\pi_{K_1\cap K_2}| - 2\\
&= |\pi_{K_1\cap K_2}| + |\pi_{K_2\setminus K_1}| - |\pi_{K_2\setminus K_1}| - 2\\
&= |\pi_{K_1\cap K_2}| - 2
\end{aligned}$$

  Comme $K_1 \cap K_2 \neq \emptyset$, nous obtenons $|\pi_{K_1\cap K_2}| \geq 2$ et le nombre de nœuds diminue.

$\square$

Toute la subtilité du résultat réside dans la construction de la bonne forme de preuve. Ce travail est réalisé dans le lemme 28 suivant.

Lemme 28 *S'il existe une preuve simple, aplatie et D-eager de $T \vdash w$ alors il existe également une preuve simple, aplatie, D-eager et $\oplus$-eager de $T \vdash w$.*

Preuve : Soit P une preuve simple, aplatie et *D-eager* de $T \vdash w$, nous appliquons plusieurs fois les transformations de preuves décrites dans les figures 5.9 page 101 et 5.11 page précédente. Ce processus termine car ces applications font décroître le nombre de nœuds d'après la proposition 11. Nous obtenons bien une preuve $\oplus$-*eager* de $T \vdash w$, que nous pouvons rendre simple et aplatie facilement. Remarquons que ces transformations ne font pas apparaître de règle (D) après une règle (GX) et ni de règle (D) juste après une règle $C)$), par conséquent la preuve est également *D-eager*. $\square$

Lemme technique pour (D_K) Nous prouvons le lemme 29 qui permet de dire que tous les atomes des nœuds issus d'une règle (D) sont bien dans les atomes de $S_c(T, w)$ pour une preuve simple, aplatie, *D-eager* et $\oplus$-*eager* de $T \vdash w$. Ce lemme nous permettra ensuite de prouver le lemme de localité atomique pour le chiffrement commutatif.

Nous effectuons, cette fois, la preuve du lemme 29 pour le cas du « ou exclusif », la preuve est similaire pour le cas des groupes abéliens.

Lemme 29 *Soit P une preuve simple, aplatie, D-eager et $\oplus$-eager de $T \vdash u$. Si P est de la forme*

$$(D_K)\dfrac{(R)\dfrac{\vdots}{T \vdash \{u\}_K\downarrow = r} \qquad \dfrac{\vdots}{T \vdash K\downarrow}}{T \vdash u}$$

alors $atoms(\{u\}_K) \subseteq atoms(S_c(T))$.

Preuve : Nous faisons une induction structurelle sur P.

Le cas de base : (R) est la règle (A) : le résultat est immédiat.

Nous focalisons notre attention sur la dernière règle (R) utilisée dans la sous-preuve de P se terminant par $\{u\}_v\downarrow$

- (R) est la règle $(C_{K'})$: Comme P est une preuve D-*eager* cela n'est pas possible.
- (R) est la règle $(D_{K'})$: impossible car la preuve P est aplatie.
- (R) est la règle (GX) : nous regardons donc d'où proviennent les hypothèses de la règle (GX) dans la figure 5.12. Nous distinguons donc les différentes possibilités pour les règles (R_i) suivant la structure de $\{u\}_K\downarrow$.

$$(D_K)\dfrac{(GX)\dfrac{(R_1)\dfrac{T\vdash B_1}{T\vdash B_1'}\quad \ldots \quad (R_n)\dfrac{T\vdash B_n}{T\vdash B_n'}}{T\vdash \{u\}_K\downarrow}\qquad \dfrac{\vdots}{T\vdash K\downarrow}}{T\vdash u\downarrow}$$

FIG. 5.12 – Illustration du cas (D_K) dans le lemme 29 page ci-contre.

Nous montrons que tous les atomes de $\{u\}_K\downarrow$ sont des atomes de $S_c(T)$. Soit $a\in \text{atoms}(\{u\}_K\downarrow)$.Remarquons que a est forcément de la forme $\{a'\}_K$, et il existe un i tel que $a\in\text{atoms}(B_i')$. Nous considérons alors les différents cas possibles pour la règle (R_i) :

- (R_i) est la règle (A).
- (R_i) est la règle $(D_{K'})$ telle que $(D_{K'})\ \dfrac{T\vdash \{w_1\}_{K'}\ T\vdash K'}{T\vdash w_1=B_i'}$. Par hypothèse d'induction atoms$(\{w_1\}_{K'})\subseteq$ atoms$(S_c(T))$, donc atoms$(w_1)=$ atoms$(B_i')\subseteq$ atoms$(S_c(T))$.
- (R_i) est la règle (C), comme P est une preuve D-*eager* nous savons que B_i' est en tête avec $\{.\}_{K'}$ tel que $K\cap K'=\emptyset$. Par conséquent B_i' s'annule par un autre terme hypothèse B_j' de la règle (GX) tel que atoms$(B_j')\subseteq$ atoms$(S_c(B_i'))$. Le terme B_j' ne provient pas d'une règle (C) car P est $\oplus$-*eager*, ni d'une règle (GX) car la preuve est aplatie. Dans les autres cas B_j' provient d'une règle (A) ou (D) ainsi atoms$(B_j')\subseteq$ atoms$(S_c(T))$. Nous concluons donc que atoms$(B_i')\subseteq$ atoms$(S_c(T))$.

□

Localité atomique. Nous disposons maintenant de tous les éléments pour prouver le lemme de localité atomique suivant.

Lemme 30 *Soit P une preuve aplatie, simple, D-eager et $\oplus$-eager de $T\vdash w$ alors tous les atomes des nœuds de P sont dans atoms$(S_c(T,w))$.*

Preuve : Nous raisonnons par induction structurelle sur la preuve P : Le cas de base correspond à la règle (A), le résultat est alors immédiat. Regardons en détail la dernière règle de P :

- (D) : nous utilisons le lemme 29 page précédente et l'hypothèse d'induction pour conclure.
- (C) : P est telle que $\dfrac{T\vdash w_1\quad T\vdash w_2}{T\vdash \{w_1\}_{w_2}}$. Par induction pour $i=1,2$ atoms$(w_i)\subseteq$ atoms$(S_c(T,w_i))$, or $w_i\in S_c(\{w_1\}_{w_2})$ donc atoms$(w_i)\subseteq$ atoms$(S_c(w))$, ce qui nous permet de conclure.
- (GX) : alors la preuve P est telle que $(GX)\dfrac{(R_1)\dfrac{T\vdash B_1}{T\vdash B_1'}\quad\ldots\quad (R_n)\dfrac{T\vdash B_n}{T\vdash B_n'}}{T\vdash w}$. Nous prouvons que atoms(B_i') sont dans atoms$(S_c(T,w))$, en regardant les différents cas possibles pour

(R_i) :
- (A) : par définition $\text{atoms}(B'_i) \in \text{atoms}(S_c(T))$,
- (D) : par le lemme 29 page 104 nous obtenons le résultat.
- (GX) : impossible, car P est aplatie.
- (C_K) : Si $\text{atoms}(B'_i) \subseteq \text{atoms}(S_c(T))$ le résultat est vrai, sinon $\text{atoms}(B'_i) \notin \text{atoms}(S_c(T))$. Le terme B'_i peut être éliminé partiellement dans une somme. Nous avons donc deux possibilités soit les atomes de B'_i sont dans w, dans ce cas $\text{atoms}(B'_i) \subseteq \text{atoms}(S_c(w))$, soit il est éliminé par un autre terme B'_j de la somme, dans ce cas $\text{atoms}(B'_i) \subseteq \text{atoms}(S_c(B'_j))$. Nous prouvons dans le dernier cas que $\text{atoms}(S_c(B'_j)) \subseteq \text{atoms}(S_c(T))$ en analysant les différents cas possibles. Comme B'_i est chiffré par l'ensemble de clefs K, B'_j n'est pas issu d'une règle $(C_{K'})$ avec $K' \neq K$, ni d'une règle (C'_K) avec $K' \cap K \neq \emptyset$ car P est $\oplus$-*eager*. Par conséquent ce terme provient d'une règle (A) ou (D) et grâce au lemme 29 page 104 $\text{atoms}(B'_j) \subseteq \text{atoms}(S_c(T))$, ce qui nous permet de conclure que pour tout i nous avons $\text{atoms}(B'_i) \in \text{atoms}(S_\oplus(T, w))$, nous avons donc la localité atomique.

□

5.3.1.3 Résultats.

Nous concluons en présentant les résultats de décidabilité pour le problème de déduction de l'intrus dans ces deux théories équationnelles.

Le cas ACUN$\{.\}_.$ avec chiffrement commutatif. La localité atomique nous permet de conclure directement comme dans le cas ACUNh.

Le cas AG$\{.\}_.$ avec chiffrement commutatif. Grâce au résultat de localité atomique du lemme 30 page précédente, nous obtenons l'ensemble de toutes les clefs, ainsi que l'ensemble de tous les atomes utilisés dans une preuve. Nous montrons la déductibilité en une étape pour la « macro » règle de la même manière que dans la section 5.2.2.2 page 84, ce qui nous permet de conclure pour le cas AG$\{.\}_.$.

5.3.2 Cas binaire.

Nous démontrons que le problème de déduction de l'intrus est EXP-SPACE difficile pour le cas binaire pour la théorie équationnelle ACUN$\{.\}_.$ commutatif asymétrique.

Nous considérons un chiffrement asymétrique, *i.e.* un terme chiffré $\{u\}_k$ peut être décrypté si et seulement si la clef inverse de la clef k, dénotée $Inv(k)$ est connue. Nous devons ajouter le symbole de fonction Inv à la signature de notre modèle et modifié la règle de déchiffrement par :

$$(D_K)\frac{T \vdash r \qquad T \vdash Inv(K)}{T \vdash u\downarrow} \quad \textit{if } r =_E \{u\}_K$$

où K est le multi-ensemble non vide $\{k_1^{\alpha_1}, \ldots, k_n^{\alpha_n}\}$, $Inv(K)$ est une notation pour le multi-ensemble $\{Inv(k_1)^{\alpha_1}, \ldots, Inv(k_n)^{\alpha_n}\}$, nous dénotons comme précédemment $T \vdash Inv(K)$ autant de séquents de chaque clef inverse que de multiplicité associée. Remarquons que si la clef inverse n'est pas connue, il n'y a aucun moyen ni de l'apprendre, ni de la générer. Tous les résultats de la section précédente sont encore valables.

Nous appelons le cas binaire pour le « ou exclusif », la situation où l'ensemble des hypothèses de T et le but u de la preuve P de $T \vdash u$ ne contiennent pas de termes avec plus de deux applications consécutives du symbole $\oplus$.

Nous montrons dans le cas d'un chiffrement commutatif que le problème de déduction est EXP-SPACE difficile par réduction au problème du mot uniforme dans les semi-groupes commutatifs (*uniform word problem in commutative semigroups*), dénoté ici *CSG* [MM82b].

Une instance du problème *CSG* est définie par :

$$\alpha_1 = \beta_1, \ldots, \alpha_n = \beta_n \models \alpha = \beta$$

où α, β, α_i et β_i sont des mots sur un alphabet. Il est essentiel pour la complexité du problème que l' alphabet soit infini (bien entendu, toute instance C du problème *CSG* n'utilise qu'une partie fini de $\Sigma(C)$ de cet alphabet). Une telle instance de *CSG* possède une solution si et seulement si $\alpha = \beta$ dans chaque semi-groupe *commutatif* satisfaisant les axiomes $\alpha_i = \beta_i$. Nous dénotons par $x =_C y$ l'égalité entre les mots x et y modulo commutativité, ce qui est équivalent à l'assertion suivante :

Soit $\alpha =_C \beta$, soit il existe une suite de paires $(\gamma_1, \delta_1), \ldots, (\gamma_l, \delta_l)$ telle que chaque paire (γ_j, δ_j) est soit $\alpha_i = \beta_i$ soit $\beta_i = \alpha_i$
et une suite de mots $c_1, \ldots, c_l$ avec $c_j \in \Sigma(C)^*$ telle que

$$\alpha =_C \gamma_1 c_1 \quad , \quad \delta_1 c_1 =_c \gamma_2 c_2, \ldots, \delta_{l-1} c_{l-1} =_C \gamma_l c_l \quad , \quad \delta_l c_l =_C \beta$$

Nous montrons donc le théorème suivant.

Théorème 5 *Dans le cas binaire pour la théorie équationnelle* $\mathsf{ACUN}\{.\}_.$ *avec chiffrement commutatif et asymétrique le problème de déduction de l'intrus est EXP-SPACE difficile.*

Preuve :
Étant donné une instance $C = (\alpha_1 = \beta_1, \ldots, \alpha_n = \beta_n \models \alpha = \beta)$ du problème *CSG*, soit

$$\begin{array}{rcl} T & = & \{\{\circledast\}_{\alpha_i} \oplus \{\circledast\}_{\beta_i} \mid 1 \leq i \leq n\} \cup \Sigma(C) \\ u & = & \{\circledast\}_\alpha \oplus \{\circledast\}_\beta \end{array}$$

où $\circledast$ est une constante, tous les symboles de $\Sigma(C)$ sont alors considérés comme des constantes, et α, β, α_i et β_i sont des clefs.

Remarque : nous nous restreignons au cas où T, u sont binaires et T, u ne contiennent aucune clef de déchiffrement (*i.e.* ne contiennent pas le symbole Inv) et aucun terme n'est en tête avec le symbole de paire.

En utilisant le résultat de localité syntaxique nous pouvons regarder uniquement les preuves utilisant les règles (D), (C) et (GX). Par le lemme de localité atomique 30 page 105, qui est encore valable pour le chiffrement asymétrique, l'existence d'une preuve de $T \vdash u$ est équivalent à l'existence d'une preuve dans laquelle tous les atomes des nœuds de la preuve sont dans les atomes de T, u pour une preuve de $T \vdash u$. Par la remarque précédente, la règle (D) n'est pas utilisée dans la preuve P de $T \vdash u$. Par conséquent, la preuve P contient uniquement des combinaisons des règles (A), (C) et (GX).

Nous appliquons donc la transformation de la figure 5.9 page 101 (fusion de deux règles (GX)) et la transformation de la figure 5.13 page suivante (permutation des règles (GX) et (C)), pour obtenir une preuve de la forme suivante :

$$\dfrac{\dfrac{\dfrac{\dfrac{\{\circledast\}_{\gamma_1} \oplus \{\circledast\}_{\delta_1} \in T}{T \vdash \{\circledast\}_{\gamma_1} \oplus \{\circledast\}_{\delta_1}}\text{(A)}}{\vdots}\text{(C)}}{T \vdash \{\circledast\}_{\gamma_1 c_1} \oplus \{\circledast\}_{\delta_1 c_1}}\text{(C)} \quad \ldots \quad \dfrac{\dfrac{\dfrac{\{\circledast\}_{\gamma_l} \oplus \{\circledast\}_{\delta_l} \in T}{T \vdash \{\circledast\}_{\gamma_l} \oplus \{\circledast\}_{\delta_l}}\text{(A)}}{\vdots}\text{(C)}}{T \vdash \{\circledast\}_{\gamma_l c_l} \oplus \{\circledast\}_{\delta_l c_l}}\text{(C)}}{T \vdash \{\circledast\}_\alpha \oplus \{\circledast\}_\beta}(GX)$$

$$
(C_K)\dfrac{(GX)\dfrac{T \vdash x_1 \quad \dots \quad T \vdash x_n}{T \vdash x_1 \oplus \ldots \oplus x_n} \qquad T \vdash K}{T \vdash \{x_1\}_K \oplus \ldots \oplus \{x_n\}_K}
$$
$$
\Downarrow
$$
$$
(GX)\dfrac{(C_K)\dfrac{T \vdash x_1 \quad T \vdash K}{T \vdash \{x_1\}_K} \quad \dots \quad (C_K)\dfrac{T \vdash x_n \quad T \vdash K}{T \vdash \{x_n\}_K}}{T \vdash \{x_1\}_K \oplus \ldots \oplus \{x_n\}_K}
$$

Fig. 5.13 – Permutation des règles (GX)-(C) en (C)-(GX).

où nous supposons, sans perdre de généralité, que les hypothèses de la règle (GX) ne valent pas 0. Il existe une telle preuve si soit $\{\boxast\}_\alpha = \{\boxast\}_\beta$, ou s'il existe une suite de termes $\{\boxast\}_{\gamma_1} \oplus \{\boxast\}_{\delta_1}, \ldots, \{\boxast\}_{\gamma_l} \oplus \{\boxast\}_{\delta_l}$ telle que chaque terme est soit de la forme : $\{\boxast\}_{\alpha_i} \oplus \{\boxast\}_{\beta_i}$ soit de la forme : $\{\boxast\}_{\beta_i} \oplus \{\boxast\}_{\alpha_i}$, et une suite $c_1, \ldots, c_l$ telle que

$$
\{\boxast\}_\alpha = \{\boxast\}_{\gamma_1 c_1}, \{\boxast\}_{\delta_1 c_1} = \{\boxast\}_{\gamma_2 c_2}, \ldots, \{\boxast\}_{\delta_{l-1} c_{l-1}} = \{\boxast\}_{\gamma_l c_l}, \{\boxast\}_{\delta_l c_l} = \{\boxast\}_\beta
$$

dans l'algèbre de terme, ce qui est équivalent à l'existence d'une solution de C. Nous concluons en utilisant le résultat de Mayr *et alii* [MM82b] qui montre la EXP-SPACE difficulté du problème *CSG*. $\Box$

5.4 Résumé des résultats pour l'intrus passif.

Dans la figure 5.14, nous rappelons les principaux résultats existants et obtenus dans cette thèse pour le problème de déduction de l'intrus en présence d'un opérateur possédant la propriété d'homomorphisme.

	Problème de déduction de l'intrus	
	ACUN	**AG**
Homomorphisme	*P-TIME [Del06a]*	*P-TIME [Del06a]*
Chiffrement distributif	*EXP-TIME*	*EXP-TIME*
Chiffrement distributif et commutatif	*2-EXP-TIME*	*2-EXP-TIME*

Fig. 5.14 – Résumé des résultats obtenus pour le problème de déduction de l'intrus.

Chapitre 6

Indépendance entre unification et problème de déduction de l'intrus.

« Everything should be made as simple as possible, but not simpler. »

Albert Einstein.

Sommaire

Remarquons que le problème de déduction de l'intrus en présence d'une théorie équationnelle vide est décidable et que l'unification pour la théorie vide est également décidable. Dans ce chapitre, nous faisons le lien entre ces deux problèmes, en exhibant des exemples de théories équationnelles. Nous montrons ainsi que les problèmes d'unification et de déduction de l'intrus modulo une théorie équationnelle sont indépendants pour un intrus passif, alors que dans le cas actif l'unification est nécessaire pour prouver le problème de sécurité (cf chapitre 8 page 125).

Nous présentons d'abord le cas où le problème de déduction de l'intrus est décidable et l'unification peut alors être soit indécidable soit décidable pour certaines théories équationnelles. Ensuite nous construisons un problème indécidable, en codant l'arrêt d'une machine de Turing. À partir de ce problème nous exhibons deux théories équationnelles où le problème de déduction de l'intrus est indécidable, et l'unification est soit indécidable, soit décidable.

6.1 Problème de déduction de l'intrus décidable modulo une théorie équationnelle.

Nous montrons grâce à deux exemples que même si le problème de déduction de l'intrus est décidable modulo une théorie équationnelle, le problème d'unification modulo cette théorie n'est pas forcément décidable ou indécidable.

6.1.1 L'unification modulo est décidable.

Dans la théorie vide par exemple, l'unification est décidable [Her30, Rob65, CB83, MM82a, PW78] et le problème de déduction de l'intrus l'est aussi, comme nous l'avons redémontré dans le chapitre 3 page 45.

6.1.2 L'unification modulo est indécidable.

Nous considérons la théorie équationnelle d'un opérateur associatif et commutatif $\oplus$ plus un symbole homomorphique h sur le symbole $\oplus$. P. Narendran montre en codant le dixième problème de Hilbert que l'unification modulo cette théorie équationnelle est indécidable [Nar96]. Dans le chapitre 5 page 69, nous avons montré que le problème de déduction de l'intrus est décidable modulo cette même théorie équationnelle [LLT05].

6.2 Problème de l'intrus indécidable modulo une théorie équationnelle.

Nous exhibons deux exemples de théories équationnelles dans lesquels l'unification est soit décidable soit indécidable et le problème de l'intrus lui reste indécidable modulo la théorie équationnelle utilisée.

Nous construisons d'abord une théorie équationnelle E dans laquelle le problème de mot clos (« word problem ») est indécidable avec cette théorie équationnelle *i.e.* savoir si deux termes clos sont égaux modulo cette théorie est indécidable. Le problème de mot en présence d'une théorie équationnelle est un cas particulier de l'unification modulo cette théorie. Ainsi, si le problème de mot clos est indécidable alors le problème d'unification l'est aussi.

6.2.1 Un problème indécidable.

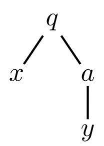
Nous construisons la théorie équationnelle particulière E, telle que le problème de mot est indécidable modulo E. Une instance du problème de mot est constituée de deux termes clos t_1 et t_2 et d'un ensemble d'équations E. Nous cherchons à savoir si $t_1 =_E t_2$?

Pour montrer que le problème de mot pour la théorie équationnelle E est indécidable, nous réduisons l'arrêt d'une machine de Turing qui est un problème indécidable. Nous prouvons que la machine de Turing avec en entrée la configuration associée au mot t_1 s'arrête si et seulement si $t_1 =_E t_2$, où t_2 est le terme associé à la configuration finale de la machine de Turing.

Preuve : Nous considérons une machine déterministe de Turing $M = (Q, \Sigma, \delta, \Box, q_0, q_F)$ où :
- Q est un ensemble fini d'états.
- Σ est un alphabet fini pour le ruban.

– $\delta : Q \times \Sigma \to Q \times \Sigma \times \{L, R, 0\}$ est une fonction partielle appelée fonction de transition, où L correspond à un décalage gauche du ruban, R à un décalage droit du ruban et 0 pas de décalage du ruban.
– $\Box \in \Sigma$ est le symbole « blanc » (le seul symbole qui peut apparaître sur le ruban infiniment souvent à n'importe quel moment).
– $q_0 \in Q$ est l'état initial.
– $q_f \in Q$ est l'état acceptant.

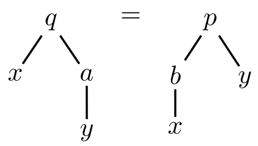
Une configuration de la machine de Turing est représentée par un terme : si la machine se trouve dans l'état $q \in Q$, que le ruban à gauche est représenté par la variable $x \in Vars$ et le ruban à droite par le symbole $a \in \Sigma$ suivi de la variable $y \in Vars$, alors le terme associé qui représente cette configuration est :



Nous noterons cette configuration par $q(x, a(y))$.

Nous supposons qu'il n'y a pas de transition qui parte d'une configuration finale, c'est à dire une configuration qui contient l'état final q_f. La machine de Turing s'arrête si la configuration finale est atteinte. Chaque transition de δ est représentée par une équation de E telle que si $a, b, \epsilon \in \Sigma$, $p, q \in Q$ et $x, y \in Vars$:

– la transition $(q, a) \to (p, b, R)$ donne l'équation suivante :



Nous utilisons la notation plus concise : $q(x, a(y)) = p(b(x), y)$.
– la transition $(q, \Box) \to (q, b, R)$ donne l'équation suivante : $q(x, \epsilon) = p(b(x), \epsilon)$.
– la transition $(q, a) \to (q, b, L)$ donne deux équations :
  – $\forall f \in \Sigma, q(f(x), a(y)) = p(x, f(b(y)))$
  – $q(\epsilon, a(y)) = p(\epsilon, \Box(b(y)))$
– la transition $(q, a) \to (q, b, 0)$ donne l'équation : $q(x, a(y)) = p(x, b(y))$
– la transition $(q, \Box) \to (q, b, 0)$ donne l'équation : $q(x, \epsilon) = p(x, b(\epsilon))$

Nous ajoutons l'équation $q_f(x, y) = t_2$ une fois l'état final atteint. Le terme t_1 représente la configuration initiale suivante $q_0 = (\epsilon, w_1(\dots(w_n(\epsilon)))))$, où $w_i \in \Sigma$. Le terme clos t_2 est associé à la configuration finale $q_f(x, y)$ de la machine de Turing.

Nous résumons les différentes équations de la théorie équationnelle E ainsi construite, où $f, a, b, \in \Sigma, p, q \in Q, x, y \in Vars$:

$$E = \left\{ \begin{array}{rcl} q(x, a(y)) & = & p(b(x), y) \\ q(x, \epsilon) & = & p(b(x), \epsilon) \\ q(f(x), a(y)) & = & p(x, f(b(y))) \\ q(\epsilon, a(y)) & = & p(\epsilon, \Box(b(y))) \\ q(x, a(y)) & = & p(x, b(y)) \\ q(x, \epsilon) & = & p(x, b(\epsilon)) \\ q_f(x, y) & = & t_2 \end{array} \right.$$

Nous montrons maintenant que le problème du mot $t_1 =_E t_2$ est équivalent à l'arrêt de la machine de Turing avec en entrée la configuration associée à t_1.

$\Rightarrow$ Par construction, si la machine de Turing prend en entrée la configuration associée au terme t_1 et s'arrête sur un état acceptant, alors $t_1 =_E t_2$.

$\Leftarrow$ Étant donné une équation entre deux termes t_1 et t_2 dans E, nous considérons toutes les équations de gauche à droite appliquée pour prouver cette égalité (nous faisons cette distinction car les transitions de la machine de Turing sont définies de la gauche vers la droite par rapport aux équations de E). Nous choisissons la plus petite séquence d'équations appliquées de gauche à droite pour démontrer $T_1 =_E t_2$. Nous raisonnons pas induction sur la taille de cette suite d'équations.

– Dans le cas de base, il n'y a pas d'équation entre t_1 et t_2 car t_1 et t_2 sont syntaxiquement les mêmes. Nous avons directement que la configuration est dans la configuration finale donc la machine de Turing s'arrête bien.
– Induction : Supposons une séquence de n équations, nous remarquons que la configuration finale $q_f(x, y)$ représente le terme t_2. Par construction de la machine de Turing il n'existe pas de transition qui parte de cette configuration, il n'existe que des transitions qui arrivent dans cette configuration. Nous regardons plus précisément une des configurations c qui précède la configuration finale. Or par hypothèse, il y a une suite de n équations qui relie t_1 à t_2, nous considérons la dernière équation de E : $t_c = t_2$, appliquée pour arriver à t_2. Par construction de la machine de Turing cette équation correspond à une transition Tr_c, partant d'une configuration c correspondant au terme t_c et arrivant dans la configuration finale. Nous appliquons l'hypothèse d'induction à la suite de $n-1$ équations reliant dans E le termes t_1 au terme t_c, ainsi nous savons que la machine de Turing avec la configuration initiale associée u terme t_1 arrive bien dans la configuration c associée au terme t_c, en appliquant la dernière transition Tr_c nous concluons. que la machine de Turing qui prend en entrée la configuration associée au terme t_1 s'arrête sur un état acceptant correspondant au terme t_2.

□

6.2.2 L'unification modulo est indécidable.

Le problème du mot est indécidable pour la théorie équationnelle E, par réduction au problème d'arrêt d'une machine de Turing, donc le problème d'unification pour cette théorie équationnelle est aussi indécidable.

Nous montrons maintenant que le problème de l'intrus modulo la théorie équationnelle E est aussi indécidable. Nous considérons le système de Dolev-Yao de la figure 6.1 prenant en compte la théorie équationnelle E :

$$(A)\frac{u \in T}{T \vdash_E u} \qquad (UL)\frac{T \vdash_E \langle u, v\rangle}{T \vdash_E u}$$

$$(P)\frac{T \vdash_E u \qquad T \vdash_E v}{T \vdash_E \langle u, v\rangle} \qquad (UR)\frac{T \vdash_E \langle u, v\rangle}{T \vdash_E v}$$

$$(C)\frac{T \vdash_E u \qquad T \vdash_E v}{T \vdash_E \{u\}_v} \qquad (D)\frac{T \vdash_E \{u\}_v \qquad T \vdash_E v}{T \vdash_E u}$$

$$(Eq)\frac{T \vdash_E u \qquad u =_E v}{T \vdash_E v}$$

FIG. 6.1 – Le modèle de preuve de Dolev-Yao étendu à la théorie équationnelle (E).

Pour démontrer que le problème de l'intrus modulo la théorie équationnelle E est indécidable, nous réduisons le problème de mot clos $t_1 =_E t_2$ au problème de déduction de l'intrus. C'est à dire, nous montrons qu'en utilisant le système de déduction de Dolev-Yao de la figure 6.1 page ci-contre déduire à partir du terme t_1 le terme t_2 est équivalent résoudre modulo E le problème du mot clos suivant entre t_1 et t_2.

Nous prouvons donc $t_1 \vdash t_2 \Leftrightarrow t_1 =_E t_2$. *Preuve* :

- ($\Leftarrow$) Si $t_1 =_E t_2$ alors nous construisons en appliquant l'axiome puis la règle (E_q) une preuve de $t_1 \vdash t_2$
- ($\Rightarrow$) Nous considérons la preuve minimale de $t_1 \vdash t_2$ en nombre de nœuds. Regardons la dernière règle de cette preuve.
  - Étant donné que le terme t_2 est en tête ni avec le symbole de paire, ni avec le symbole de chiffrement, il ne peut être directement issu ni de la règle (P), ni de la règle (C).
  - Si la preuve se termine par une règle (UR), (UL) ou (D), comme l'unique terme dans la connaissance initiale est t_1 qui est ni en tête avec le symbole de paire, ni le symbole de chiffrement, et que les équations de E ne comporte pas de symbole ni de paire ni de chiffrement, il y a forcément eu une application à partir de t_1 d'un constructeur de paire ou de chiffrement. Ceci implique que nous pouvons trouver une preuve plus petit de $t_1 \vdash t_2$, ce qui contredit la minimalité de la preuve, ainsi la dernière règle ne peut être ni (UR), ni (UL) et ni (D).
  - Si la preuve se termine par la règle (E_q) nous obtenons bien la preuve minimale constituée de l'application de l'axiome puis de la règle (E_q), ce qui implique que $t_1 =_E t_2$.

□

Nous concluons donc que le problème de déduction de l'intrus pour la théorie équationnelle E est indécidable.

6.2.3 L'unification modulo est décidable.

Nous considérons une nouvelle théorie équationnelle E_s construite à partir de la théorie équationnelle E, pour laquelle l'unification est décidable et le problème de déduction de l'intrus est indécidable.

6.2.3.1 Nouvelle théorie équationnelle : E_s.

Nous modifions la théorie équationnelle E de la section précédente en ajoutant un nouveau symbole de fonction s. Nous obtenons un nouvel ensemble d'équations E_s où $f, a, b, \in \Sigma_0, p, q \in Q, x, y \in Vars$:

$$E_s = \left\{ \begin{array}{rcl} s(q(x,a(y))) & = & p(b(x),y) \\ s(q(x,\epsilon)) & = & p(b(x),\epsilon) \\ s((f(x),a(y))) & = & p(x,f(b(y))) \\ s(q(\epsilon,a(y))) & = & p(\epsilon,\Box(b(y))) \\ s(q(x,a(y))) & = & p(x,b(y)) \\ s(q(x,\epsilon)) & = & p(x,b(\epsilon)) \\ s(q_f(x,y)) & = & t_2 \end{array} \right.$$

6.2.3.2 L'unification modulo E_s est décidable.

Nous transformons le système d'équations E_s en un système de réécriture R_{E_S} où $f, a, b, \in \Sigma_0, p, q \in Q, x, y \in Vars$:

$$R_{E_s} = \left\{ \begin{array}{rcl} s(q(x,a(y))) & \rightarrow & p(b(x),y) \\ s(q(x,\epsilon)) & \rightarrow & p(b(x),\epsilon) \\ s((f(x),a(y))) & \rightarrow & p(x,f(b(y))) \\ s(q(\epsilon,a(y))) & \rightarrow & p(\epsilon,\square(b(y))) \\ s(q(x,a(y))) & \rightarrow & p(x,b(y)) \\ s(q(x,\epsilon)) & \rightarrow & p(x,b(\epsilon)) \\ s(q_f(x,y)) & \rightarrow & t_2 \end{array} \right.$$

Le nombre d'application de la fonction s décroît, le système de réécriture termine donc. La construction de E_s est basée sur une machine de Turing déterministe, donc il n'y a pas de superposition entre les termes, par conséquent il n'existe pas de paire critique. Nous concluons donc que le système de réécriture R_{E_S} est confluent. Nous en déduisons que le système de réécriture associé à E_s est convergent. Nous rappelons maintenant la définition de sur-réduction.

Définition 29 (Sur-réduction) *Un terme t se sur-réduit (« is narrowed ») en t', à la position non variable $p \in Dom(t)$, utilisant la règle de réécriture $l \rightarrow r$ et la substitution σ, où σ est l'unificateur le plus général de $t|_p$ et l, et $t' = \sigma(t[r]_p)$, dénoté par $t \rightsquigarrow_{[p,l\rightarrow r,\sigma]} t'$ et nous supposons toujours qu'il n'y a pas de conflit de variables entre les règles et les termes, i.e. $Vars(l,r) \cap Var(t) = \emptyset$.*

D'après le résultat de Hullot [Hul80], la sur-réduction (« narrowing ») est complète pour une théorie équationnelle représentée par un système de réécriture convergent (une présentation de ce résultat est donnée dans le papier de F. Baader et W. Snyder [BS01] et dans le papier de A. Middeldorp [Mid94]). Une conséquence de ce résultat est que si la sur-réduction termine pour une théorie représentée par un système de réécriture convergent, alors l'unification est décidable.

Pour montrer la décidabilité de l'unification modulo la théorie équationnelle E_s nous prouvons donc que la sur-réduction (« narrowing ») termine pour E_S.

Remarquons d'abord qu'il y a un nombre fini de règles de réécriture dans R_{E_s}, par conséquent la sur-réduction a un nombre fini de possibilités dans la règle applicable. De plus, tous les membres gauches des règles débutent par le symbole de fonction s et toutes les règles préservent l'ensemble des variables, ($vars(l) = vars(r)$). Pour prouver la terminaison de la sur-réduction (« narrowing »), nous considérons la mesure n_s qui compte le nombre de symboles s dans un terme. Selon la forme particulière de nos égalités la mesure décroît, car les variables sont préservées et le symbole de fonction s lui disparaît. Par construction de la sur-réduction (« narrowing »), σ est l'unificateur le plus général de $t|_p$ et l. Nous pouvons donc conclure que la sur-réduction termine et donc l'unification modulo E_s est décidable.

6.2.3.3 Le problème de l'intrus est indécidable modulo E_s.

Nous utilisons le système de déduction de Dolev-Yao suivant :

Le système de réécriture R_{E_S} termine car le nombre d'applications de s décroît et il est confluent car il n'y a pas de paire critique. Le système de réécriture R_{E_S} est donc convergent, d'après les résultats de la section 3.2.2 page 49, nous éliminons donc la règle (E_q) dans le système de Dolev-Yao de la figure 6.2 page ci-contre et travaillons alors avec des formes normales.

Pour plus de clarté, nous ne présentons pas les démonstrations formelles des résultats suivants car elles sont semblables à celles effectuées dans les chapitres précédents de cette première partie pour les résultats de localité.

$$
\begin{array}{ll}
(A)\dfrac{u \in T}{T \vdash_{E_s} u} & (UL)\dfrac{T \vdash_{E_s} \langle u, v\rangle}{T \vdash_{E_s} u} \\
(P)\dfrac{T \vdash_{E_s} u \qquad T \vdash_{E_s} v}{T \vdash_{E_s} \langle u, v\rangle} & (UR)\dfrac{T \vdash_{E_s} \langle u, v\rangle}{T \vdash_{E_s} v} \\
(C)\dfrac{T \vdash_{E_s} u \qquad T \vdash_{E_s} v}{T \vdash_{E_s} \{u\}_v} & (D)\dfrac{T \vdash_{E_s} \{u\}_v \qquad T \vdash_{E_s} v}{T \vdash_{E_s} u} \\
(E_q)\dfrac{T \vdash_{E_s} u \qquad u =_{E_s} v}{T \vdash_{E_s} v} & (S)\dfrac{T \vdash_{E_s} u}{T \vdash_{E_s} s(u)}
\end{array}
$$

FIG. 6.2 – Le modèle de preuve de Dolev-Yao étendu à la théorie équationnelle (E_s).

Nous pouvons prouver que les règles classiques de Dolev-Yao *i.e.* (P), (C), (D), (UL) et (UR), ne sont pas utilisées pour obtenir à partir du terme t_1 le terme t_2. La preuve est alors uniquement constituée d'applications de la règle (S). Ainsi savoir si à partir du terme t_1 nous pouvons déduire le terme t_2 en appliquant que des règles (S) revient à déterminer le nombre d'application de la règle (S) c'est à dire décider l'arrêt de la machine de Turing construite précédemment. Par conséquent, le problème de déduction de l'intrus pour la théorie équationnelle E_s reste indécidable.

Récapitulatif Nous présentons dans la figure 6.3 les différentes théories équationnelles exhibées dans ce chapitre montrant que le problème de déduction de l'intrus et le problème d'unification sont deux problèmes indépendants.

		Problème de déduction de l'intrus	
		Décidable	Indécidable
Unification	Décidable	$\emptyset$	E_s
	Indécidable	ACh	E

FIG. 6.3 – Récapitulatif des théories équationnelles prouvant l'indépendance du problème de déduction de l'intrus et du problème d'unification.

Troisième partie

Problème de sécurité (intrus actif).

Chapitre 7

Caractéristiques des protocoles.

« All truths are easy to understand once they are discovered ; the point is to discover them. »
Galileo Galilei.

Sommaire

Quelle classe de protocoles est-il réaliste d'étudier ? Pour répondre à cette question, nous définissons d'abord une classe de protocoles qui nous semble pertinente : la classe des protocoles « déterministes ». Nous montrons qu'il est décidable de savoir si un protocole est déterministe en présence de théories équationnelles. Ensuite nous proposons d'analyser ces protocoles déterministes en les transformant en systèmes de contraintes. Nous définissons plusieurs types de systèmes de contraintes et démontrons que tous protocoles déterministes peuvent se traduire en un système de contraintes « bien défini », notion introduite par J. Millen et V. Shmatikov [MS05]. Dans les chapitres suivants, nous construisons une procédure de décision pour le problème de sécurité en présence de la théorie équationnelle ACUNh en résolvant ces systèmes de contraintes bien définis.

7.1 Protocoles déterministes.

Nous considérons le modèle introduit par N. Durgin *et alii* [DLMS99] où un protocole est en un ensemble de séquences de messages échangés entre différents agents. Un échange de messages est noté $u \to v$, où $u, v \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ et représente la suite d'instruction « *reçu*(u) ; *envoie*(v) ». Un agent reçoit un message u, si ce message filtre (« match ») le motif (« pattern ») attendu lors de l'exécution normale du protocole, l'agent continue donc le déroulement du protocole et envoie le message v. Un protocole $\mathcal{P}$ est donc une séquence d'échanges de messages *i.e.* une liste de la forme $(u_i \to v_i)_{1 \leq i \leq k}$ contenant des variables.

$$\mathcal{P} \quad := \quad \begin{cases} u_1 & \rightarrow & v_1 \\ & \vdots & \\ u_n & \rightarrow & v_n \end{cases}$$

Nous cherchons à savoir si une donnée s reste secrète, en considérant que l'intrus contrôle le réseau. Ainsi l'identité des agents, qui jouent le protocole, n'est pas une information pertinente pour nous, elle n'apparaît pas dans notre notation. Un protocole est *déterministe* si une séquence de messages reçus détermine de façon unique les messages envoyés, plus formellement :

Définition 30 (Protocole deterministe) *Un protocole est* déterministe *si pour toutes substitutions closes τ, σ et pour tout $i \in [1, \ldots, n]$, $u_1\tau = u_1\sigma, \ldots, u_i\tau = u_i\sigma$ alors $v_i\tau = v_i\sigma$.*

Cette notion est totalement indépendante de la théorie équationnelle puisque le filtrage (« matching ») s'effectue modulo la théorie équationnelle, de plus la substitution par filtrage (« matching substitution ») n'est pas unique. Dans le corollaire 1, nous prouvons qu'il est décidable de savoir si un protocole n'est pas déterministe pour la théorie équationnelle ACUNh.

Corollaire 1 *Soit $t_1, \ldots, t_n, t$ des termes dans $\mathcal{T}(\mathcal{F}, \mathcal{X})$, savoir s'il existe deux substitutions closes σ, τ telles que*

$$t_1\tau = t_1\sigma \wedge \ldots \wedge t_n\tau = t_n\sigma \wedge t\tau \neq t\sigma$$

est décidable pour la théorie équationnelle ACUNh.

Preuve : Soit $X = vars(t_1, \ldots, t_n, t)$ et ρ une variable différente de X. Soit $t'_i = t_i\rho(i = 1, \ldots, n)$, et $t' = t\rho$, le lemme est équivalent à l'existence d'une substitution close τ telle que $t_1\tau = t'_1\tau \wedge \ldots \wedge t_n\tau = t'_n\tau \wedge t\tau \neq t'\tau$. Cette propriété est décidable grâce au lemme 36 page 136. $\square$

Exemple 20 *Le protocole construit par l'envoi de messages suivants : $x + y \quad \rightarrow \quad x$ n'est pas déterministe (pour la théorie équationnelle ACUNh), considérons $\sigma = \{x \mapsto 0, y \mapsto 0\}$ et $\tau = \{x \mapsto \langle 0, 0\rangle, y \mapsto \langle 0, 0\rangle\}$. Nous obtenons : $(x + y)\sigma = 0 = (x + y)\tau$ et $x\sigma = 0 \neq \langle 0, 0\rangle = x\tau$.*

Par contre le protocole suivant est déterministe : la valeur de $h(x+y)$ est uniquement déterminée une fois la valeur de $x + y$ fixée (même si la matching substitution n'est pas unique), la valeur de $h(h(y))$ est uniquement déterminée une fois les valeurs de $x + y$ et x connues.

$$\begin{cases} x + y & \rightarrow & h(x + y) \\ x & \rightarrow & h(h(y)) \end{cases}$$

Si nous considérons une autre théorie équationnelle où il existe un résultat de disunification alors cette caractérisation est décidable dans cette théorie.

7.2 Systèmes de contraintes.

Nous définissons d'abord formellement ce qu'est un système de contraintes et expliquons comment à partir d'un protocole sous la forme d'une séquence de messages reçus et envoyés nous construisons un système de contraintes.

7.2.1 Définitions.

Nous construisons à partir d'un protocole $\mathcal{P}$, décrit sous la forme d'échange de messages envoyés et reçus, un système de contraintes $\mathcal{C}$.

$$\mathcal{P} \quad := \quad \begin{cases} u_1 & \to & v_1 \\ & \vdots & \\ u_n & \to & v_n \end{cases}$$

Ce système de contraintes modélise le point de vue de l'attaquant. L'intrus prend en quelque sorte possession du réseau, il voit passer tous les messages et il est capable d'envoyer des messages aux différents participants. La première contrainte contient sur la partie gauche la connaissance initiale de l'intrus T_0 et sur la partie droite le premier message reçu u_1 pendant une session normale du protocole par le premier agent. La seconde contrainte contient sur sa partie gauche, la connaissance initiale de l'intrus plus le message v_1 envoyé comme réponse au message u_1, et sur la partie droite le prochain message échangé. Ainsi la connaissance de l'intrus est croissante. Le n-ième et dernier message reçu est u_n et la réponse émise est v_n.

$$\mathcal{C} \quad := \quad \begin{cases} T_0 & \Vdash & u_1 \\ T_0, v_1 & \Vdash & u_2 \\ & \vdots & \\ T_0, v_1, \ldots, v_{n-1} & \Vdash & u_n \\ T_0, v_1, \ldots, v_{n-1}, v_n & \Vdash & s \end{cases}$$

Nous nous intéressons à la propriété de secret *i.e.* , savoir si un intrus peut déduire un secret s. Pour modéliser cette propriété l'idée est de rajouter une dernière contrainte au système dans laquelle l'intrus doit déduire le secret s à partir des messages échangés précédemment. S'il existe une solution à ce système de contraintes alors il existe une attaque permettant à l'attaquant de connaître s. Nous devons considérer tous les systèmes de contraintes à une contrainte puis deux contraintes etc ..., auxquels nous ajoutons à chaque fois la contrainte qui permet à un intrus de déduire le secret s à partir du dernier message échangé, comme nous l'avons fait sur le système complet $\mathcal{C}$. Il faut ensuite étudier chaque système de contraintes. Si un des systèmes possède une solution alors il existe une attaque permettant à l'intrus de déduire le secret s. Si aucun système ne possède de solution, alors il n'existe pas d'attaque sur le protocole sous les hypothèses considérées. Étudier tous ces systèmes est nécessaire, car il se peut qu'un système possède une solution pour les cinq premières contraintes, qui permettent de déduire le secret alors qu'avec toutes les contraintes le système ne possède pas de solution. Dans ce cas là l'intrus peut bien sûr obtenir le secret mais plus tard l'exécution du protocole s'arrêtera et le protocole ne peut être entièrement exécuté. Dans la suite nous considérerons un seul système de contraintes, car tous ces systèmes seront résolus par la même technique.

Définition 31 (Système de contraintes) *Une* contrainte *(resp. une* contrainte une étape*, une* contrainte $\mathsf{M_E}$*) est un séquent de la forme* $T \Vdash u$ *(resp.* $T \Vdash_1 u$*,* $T \Vdash_{\mathsf{M_E}} u$*) où* T *est sous-ensemble fini de* $\mathcal{T}(\mathcal{F}, \mathcal{X})$ *et* $u \in \mathcal{T}(\mathcal{F}, \mathcal{X})$*. Nous appelons* T l'ensemble des hypothèses *d'une contrainte. Un* système de contraintes *est une séquence de contraintes.*

Définition 32 (Solutions d'un système de contraintes) *Une solution d'un système de contraintes* $\mathcal{C}$ *(resp. une étape,* $\mathsf{M_E}$*) est une substitution* σ *telle que*

- *pour chaque* $T \Vdash u \in \mathcal{C}$ *il existe une preuve de* $T\sigma \vdash u\sigma$ *;*

– *pour chaque* $T \Vdash_1 u \in \mathcal{C}$ *le terme* $u\sigma$ *est déductible en une étape de* $T\sigma$;
– *pour chaque* $T \Vdash_{\mathsf{M_E}} u \in \mathcal{C}$ *le terme* $u\sigma$ *est déductible en une étape* $\mathsf{M_E}$ *de* $T\sigma$.

Exemple 21 *Considérons le protocole naïf suivant : l'agent A envoie le « ou exclusif » de son identité A et d'un secret s crypté avec une clef secrète symétrique k connue de A et de B, grâce au chiffrement de Vernam. L'agent B connaissant la clef k peut donc renvoyer le message $s \oplus A$ à l'agent A.*

$$\begin{array}{lllll} 1 & A & \rightarrow & B: & A \oplus k \oplus s \\ 2 & B & \rightarrow & A: & A \oplus s \end{array}$$

Nous supposons que l'intrus connaisse les constantes A et B qui représentent les identités de l'agent A et de l'agent B. Nous construisons le système de contraintes correspondant à ce protocole :

$$\left\{ \begin{array}{lll} A, B & \Vdash & x \oplus k \oplus s \\ A, B, x \oplus s & \Vdash & k \end{array} \right.$$

Il existe une substitution $\sigma = \{x \rightarrow A \oplus k \oplus s\}$, nous obtenons donc :

$$\left\{ \begin{array}{lll} A, B & \Vdash & A \\ A, B, A \oplus k & \Vdash & k \end{array} \right.$$

L'intrus peut donc connaître la clef k en faisant le « ou exclusif » du premier et du second message, nous obtenons k la clef secrète entre les agents A et B.

7.3 Systèmes de contraintes bien définis.

Nous définissons maintenant les notions de solution non effondrante, F'-solution, système de contraintes bien défini. Toutes ces notions seront utilisées dans le chapitre 10 page 145.

Définition 33 (Solution non effondrante) *Une solution σ d'un système de contrainte $\mathcal{C}$ est* non effondrante *si pour tout $u, v \in St_{\mathsf{E}}(\mathcal{C}) \setminus \mathcal{X}$ tels que $u\sigma =_{\mathsf{E}} v\sigma$ alors $u =_{\mathsf{E}} v$.*

Définition 34 ($\mathcal{F}'$-solution) *Si $\mathcal{F}'$ est une sous-signature de $\mathcal{F}$ alors une solution σ d'un système de contraintes est appelée* $\mathcal{F}'$-solution *si $x\sigma \in \mathcal{T}(\mathcal{F}', \mathcal{X})$ pour tout $x \in dom(\sigma)$.*

Remarque 1 *Si σ est une solution d'une contrainte $T \Vdash u$ (resp. une étape contraintes, $\mathsf{M_E}$ contrainte), alors $\sigma\theta$ est aussi une solution de $T \Vdash u$ pour toute substitution θ.*

Définition 35 (Système de contraintes bien définis) *Un système de contraintes $\mathcal{C} = \{T_i \Vdash u_i\}_{1 \leq i \leq k}$ est* bien défini *si :*

1. (monotonie) *pour tout $i < k$: $T_i \subseteq T_{i+1}$,*
2. (origination) *pour toute substitution θ : $\mathcal{C}\theta$ satisfait la propriété suivante :*
$$\forall i \leq k,\ \forall x \in vars(T_i\theta),\ \exists j < i \text{ tel que } x \in vars(u_j\theta).$$

Exemple 22 *Ce système de contraintes n'est pas bien défini.*

$$\mathcal{C} \quad := \quad \left\{ \begin{array}{lll} T_0 & \Vdash & X \oplus Y \\ T_0, X & \Vdash & Z \end{array} \right.$$

Car la substitution $\theta = \{Y \to X\}$ nous donne un système de contraintes $\mathcal{C}\theta$ qui n'est pas bien formé ce qui contredit la propriété d'origination.

$$\mathcal{C}\theta \quad := \quad \begin{cases} T_0 & \Vdash \quad 0 \\ T_0, X & \Vdash \quad Z \end{cases}$$

Cette notion de système de contraintes bien défini est introduite par J. Millen et V. Shmatikov [MS05]. Elle est définie de manière similaire pour les systèmes de contraintes une étape (resp. M_E). Dans ce papier les auteurs étudient une classe « raisonnable » de protocoles, les protocoles « bien définis ». Nous montrons que tous protocoles déterministes peuvent être représentés par un système de contraintes bien défini.

7.4 D'un protocole déterministe vers un système de contraintes bien défini.

Par construction le système de contraintes construit à partir d'un protocole déterministe est monotone. Il ne reste plus qu'à montrer la propriété d'origination.

Lemme 31 *Si P est un protocole déterministe alors le système de contraintes associé est bien défini (« well-defined »).*

Preuve : Supposons qu'un système de contraintes ne soit pas bien défini, par conséquent il existe une substitution μ et une variable x tels que $x \in vars(v_i\mu)$ et $x \notin \bigcup_{j<i} vars(u_j\mu)$.

Soit a, b deux constantes qui ne sont pas utilisées dans le protocole telles que $\mu_a = \{x \mapsto a\}$ et $\mu_b = \{x \mapsto b\}$. Puisque $x \notin \bigcup_{j<i} vars(u_j\mu)$ nous avons $v_j\mu\mu_a = v_j\mu\mu_b$ pour tout $j < i$, mais $u_i\mu\mu_a \neq u_i\mu\mu_b$. Ceci contredit le fait que le protocole soit déterministe. $\square$

Remarquons qu'un système de contraintes bien défini n'est pas forcément issu d'un protocole déterministe comme le montre l'exemple 23.

Exemple 23 *Supposons que $f(a, a) = c$ et $f(b, b) = c$ le protocole suivant n'est pas déterministe.*

$$\begin{array}{llcll} 1 & A & \to & B: & f(x, x) \\ 2 & B & \to & A: & x \end{array}$$

Le système de contraintes associé est pourtant bien défini :

$$\begin{cases} T_0 & \Vdash \quad f(x, x) \\ T_0, x & \Vdash \ldots \end{cases}$$

Nous nous intéressons au cas actif pour la théorie équationnelle ACUNh, partant d'un protocole déterministe nous pouvons associer un système de contraintes, nous proposons dans la suite une procédure de résolution des ces systèmes de contraintes. Dans le chapitre 8 page 125, nous montrons un résultat sur l'unification modulo ACUNh nécessaire pour notre procédure de décision. Puis dans le chapitre 9 page 137, nous proposons une méthode de résolution d'équations particulières qui nous permet de terminer notre algorithme de résolution de contrainte dans le chapitre 10 page 145.

Chapitre 8

Unification.

« Necessity, who is the mother of invention. »
Platon.

Sommaire

Une condition nécessaire pour que le problème de sécurité soit décidable modulo une théorie équationnelle est que l'unification modulo cette théorie soit décidable. Nous montrons d'abord les relations qui existent entre ces deux problèmes. Ensuite, nous développons un algorithme d'unification pour la théorie équationnelle ACUNh en utilisant une approche basée sur les automates.

8.1 Relations entre unification et le problème de sécurité.

Nous montrons que la décidabilité de l'unification est un paramètre nécessaire mais pas suffisant pour que le problème de sécurité soit décidable. Si l'unification est indécidable le problème de sécurité est lui aussi indécidable, par contre si l'unification est décidable nous ne pouvons pas conclure pour le problème de sécurité.

8.1.1 L'unification est nécessaire.

Nous prouvons que si l'unification est indécidable le problème de sécurité est indécidable. Nous réduisons le problème d'unification de deux termes $u[x_1, \ldots, x_n]$ et $v[x_1, \ldots, x_n]$ avec les variables $x_1, \ldots, x_n$ au problème d'insécurité de l'intrus pour une session grâce au protocole suivant. Ce protocole cherche à protéger le secret s entre deux participants A et B :

A :		$\rightarrow$	$M_1, \ldots, M_n$	(A envoie une séquence de n messages.)
B :	$x_1, \ldots, x_n$	$\rightarrow$	$\{u[x_1, \ldots, x_n], v[x_1, \ldots, x_n]\}_k$	(B répond en instanciant les variables $x_1, \ldots, x_n$ par les messages envoyés par A.)
A :	$\{x, x\}_k$	$\rightarrow$	s	(k est une clef fraîche.)

8.1.2 L'unification n'est pas suffisante.

La théorie équationnelle des groupes abéliens et de l'homomorphisme AGh nous offre un exemple où l'unification est décidable, comme l'ont montré P. Narendran [Nar96] et F. Baader [Baa93].

Comme l'a montré S. Delaune [Del06b], en même temps que nous élaborions notre résultat pour le cas actif dans la théorie équationnelle ACUNh, le problème de sécurité pour un intrus actif est indécidable. Ce résultat est prouvé par une réduction du dixième problème d'Hilbert [Dav73, Mat93].

8.2 Unification ACUNh.

Nous rappelons d'abord les appellations courantes utilisées en unification et présentées par exemple par F. Baader et W. Snyder [BS01]. Les différentes hiérarchies des problèmes d'E-unification, où E désigne une théorie équationnelle, sont :

- le *problème élémentaire d'E-unification* correspond à des systèmes d'équations entre des termes construits à partir des fonctions de symboles de E et des variables.
- le *problème d'E-unification avec constantes* correspond à des systèmes d'équations entre des termes construits à partir des fonctions de symboles de E, des constantes libres et des variables.
- le *problème d'E-unification général* correspond à des systèmes d'équations entre des termes construits à partir des fonctions de symboles de E, des symboles de fonctions libres (les fonctions de symboles d'arité zéro sont des constantes) et des variables.

Nous nous intéressons au calcul de l'ensemble minimal et complet des unificateurs. Cet ensemble peut être de plusieurs types (unitaire 1, finitaire ω, infinitaire ∞, ou de type zéro 0), ceci en fonction de sa cardinalité.

- type zéro : s'il n'existe pas d'ensemble minimal complet d'E-unificateurs.
- unitaire : si l'ensemble minimal complet d'E-unificateurs est de cardinal un.
- finitaire : si l'ensemble minimal complet d'E-unificateurs est de cardinal fini.
- infinitaire : si l'ensemble minimal complet d'E-unificateurs est de cardinal infini.

Nous rappelons dans la Figure 8.1 page suivante les principaux résultats réalisés pour le problème d'E-unification général pour les théories équationnelles : AC, ACUN, AG, ACh, ACUNh et AGh.

Nous prouvons dans cette section le théorème 7 page 154. Ce théorème assure que l'unification modulo ACUNh est finitaire et prouve le lemme technique 34 page 135. Ce lemme est un élément important dans les preuves du chapitre 10 page 145.

Théorie équationnelle	Décidabilité
AC	Décidable [Sti75, Fag84, Kir89]
ACUN	NP-complet [GNW00]
AG	Décidable [LBB84]
ACh	Indécidable [Nar96]
ACUNh	NP-complet [GNW00]
AGh	Décidable [Baa93]

FIG. 8.1 – Récapitulatif des résultats de décidabilité pour l'unification.

Notons d'abord que F. Baader et K. U. Schulz [BS96] proposent un algorithme NP de combinaison pour le problème d'unification. Considérons maintenant la signature Σ composée de constantes, d'un symbole de fonction $\oplus$ associatif, commutatif, unitaire et avec un élément neutre ACUNet un symbole d'homomorphisme h avec la propriété suivante $h(x \oplus y) = h(x) \oplus h(y)$. P. Narendran prouve que le problème d'unification avec constantes modulo ACUNhest décidable [GNW00] en temps polynomial, et affirme que le problème d'unification général est NP-complet. De plus F. Baader*et alii* donne un algorithme de décidabilité pour l'unification pour plusieurs théories équationnelles qui contiennent un symbole d'homomorphisme comme les groupes abéliens [Baa93]. Cet algorithme utilise les bases de Gröbner, il est donc difficile de connaître la complexité exacte de ce résultat.

Nous proposons ici un algorithme basé sur des automates pour la théorie ACUNh. Cet algorithme calcule l'ensemble complet des unificateurs sur la signature Σ. Cela nous permet une analyse plus précise de la complexité de notre algorithme. Ensuite nous montrons que l'unification avec restriction sur les constantes (consulter [BS96] pour plus de détails) est finitaire. Nous appliquons alors le résultat de combinaison de F. Baader and Schimidt-Schauss [BS96] pour obtenir un algorithme qui calcule l'ensemble complet des unificateurs sur la signature Σ augmentée avec des symboles libres.

8.2.1 Lien entre unification et équations diophantiennes.

Nous rappelons en détail le lien qui existe entre les équations linéaires diophantiennes sur l'anneau $\mathbb{Z}/2\mathbb{Z}[h]$ et le problème d'unification modulo ACUNh. Ceci est un cas particulier de la théorie plus générale, développée par W. Nutt [Nut90] pour l'unification en présence de théories monoïdales. Notre présentation se focalise uniquement sur la théorie ACUNh, ainsi elle est plus simple et plus compréhensible que celle donnée par W. Nutt pour le cas général.

Nous présentons les notations utilisées dans le reste de ce chapitre pour manipuler les polynômes de $\mathbb{Z}/2\mathbb{Z}[h]$. Un m-tuple de polynômes de $\mathbb{Z}/2\mathbb{Z}[h]$, dénoté par $(p_1, \ldots, p_m)$ représente un ensemble de m polynômes de $\mathbb{Z}/2\mathbb{Z}[h]$. Nous dénotons $p.q$ la multiplication entre deux polynômes de $\mathbb{Z}/2\mathbb{Z}[h]$. Nous décomposons alors un polynôme $p(h) \in \mathbb{Z}/2\mathbb{Z}[h]$ de la manière suivante :

$$p(h) = \sum_{i=0}^{n} b_i h^i \quad \text{où} \quad b_i \in \mathbb{Z}/2\mathbb{Z}$$

Nous notons par le symbole $\odot$ le produit d'un terme et d'un polynôme. Ce produit est défini par :

$$(\sum_{i=0}^{n} b_i h^i) \odot t = \sum_{i=0 \ | \ b_i \neq 0}^{n} h^i(t)$$

Ainsi le produit du polynôme (h^2+1) par le terme $(X \oplus h(a))$ s'écrit :

$$(h^2+1) \odot (X \oplus h(a)) = h^2(X) \oplus X \oplus h^3(a) \oplus h(a)$$

D'autre part nous pouvons décomposer un terme suivant les variables qui le composent. Considérons un terme t tel que les variables de t soient dénotées par $\mathcal{V}(t) = \{X_1, \ldots, X_p\}$, alors t peut se décomposer de la manière suivante : $t = t^{X_1} \odot X_1 \oplus \ldots \oplus t^{X_p} \odot X_p \oplus t_0$ où $t^{X_1}, \ldots, t^{X_p} \in \mathbb{Z}/2\mathbb{Z}[h]$, et t_0 un terme clos. Enfin nous utilisons le symbole binaire $\oplus$ pour l'addition dans $\mathbb{Z}/2\mathbb{Z}[h]$.

Définition 36 (Degré d'un polynôme de $\mathbb{Z}/2\mathbb{Z}[h]$) *Soit p un polynôme de $\mathbb{Z}/2\mathbb{Z}[h]$ tel que $p = \sum_{i=0}^{n} b_i h^i$. Le* degré *de p, dénoté par $deg(p)$, vaut n pour $b_i \neq 0$. Nous étendons naturellement cette notion aux tuples de polynômes par $deg(p_1, \ldots, p_n) = (deg(p_1), \ldots, deg(p_n))$.*

Nous présentons d'abord une méthode de résolution pour des systèmes linéaires d'équations diophantiennes.

8.2.1.1 Solution d'équations linéaires diophantiennes dans $\mathbb{Z}/2\mathbb{Z}[h]$.

Dans un premier temps, nous considérons la signature Σ composée de $\{\oplus, h, c_1, \ldots, c_m\}$. Soit (E) un système d'équations de la forme suivante :

$$(E) \quad \left\{ \begin{array}{lcl} A_{1,1}.X_1 \oplus \ldots \oplus A_{1,n}.X_n & = & B_1 \\ & \vdots & \\ A_{m,1}.X_1 \oplus \ldots \oplus A_{m,n}.X_n & = & B_m \end{array} \right.$$

où les $A_{i,j}$, B_j sont des polynômes de $\mathbb{Z}/2\mathbb{Z}[h]$, et les inconnues les X_i sont aussi des polynômes de $\mathbb{Z}/2\mathbb{Z}[h]$. Nous dénotons par (H) le système homogène associé à (E) *i.e.* le système où tous les B_j valent 0.

$$(H) \quad \left\{ \begin{array}{lcl} A_{1,1}.X_1 \oplus \ldots \oplus A_{1,n}.X_n & = & 0 \\ & \vdots & \\ A_{m,1}.X_1 \oplus \ldots \oplus A_{m,n}.X_n & = & 0 \end{array} \right.$$

Nous rappelons la définition d'ordre, ordre total et ordre partiel. Ensuite nous définissons un ordre partiel entre les tuples de polynômes de $\mathbb{Z}/2\mathbb{Z}[h]$ et la relation d'équivalence associée. Nous pourrons ainsi parler des solutions minimales de (E) pour cette relation d'ordre.

Définition 37 (Relation d'ordre) *Une relation d'ordre $\mathcal{R}$ sur un ensemble E est une relation binaire sur E réflexive, transitive et antisymétrique.*
réflexive : $\forall x \in E, x\mathcal{R}x$
antisymétrique : $\forall x \in E, \forall y \in E, (x\mathcal{R}y) \wedge (y\mathcal{R}x) \Rightarrow x = y$
transitive $\forall x \in E, \forall y \in E, \forall z \in E, (x\mathcal{R}y) \wedge (y\mathcal{R}z) \Rightarrow x\mathcal{R}z$

Définition 38 (Relation d'ordre totale et partielle) *Une relation d'ordre est* totale *si pour tous x, y dans E, nous avons $x\mathcal{R}y$ ou $y\mathcal{R}x$. Une relation d'ordre est* partielle *si elle n'est pas totale i.e. $\exists x \in E, \exists \in E, \neg(x\mathcal{R}y) \wedge \neg(y\mathcal{R}x)$*

Dans l'exemple 24 page suivante, nous donnons un exemple de tuples de polynômes de $\mathbb{Z}/2\mathbb{Z}[h]$ incomparables pour l'ordre partiel introduit dans la définition 39 page ci-contre.

Définition 39 (Quasi-ordre) *Soit deux m-tuples de polynômes $(p_1, \ldots, p_m)$ et $(q_1, \ldots, q_m)$, nous définissons un quasi-ordre sur les tuples de polynômes tel que $(p_1, \ldots, p_m) \leq (q_1, \ldots, q_m)$ si et seulement si $\forall 1 \leq i \leq m deg(p(i)) \leq deg(q_i)$. De plus $(p_1, \ldots, p_m) < (q_1, \ldots, q_m)$ si et seulement si $(p_1, \ldots, p_m) \leq (q_1, \ldots, q_m)$ et $\neg((q_1, \ldots, q_m) \leq (p_1, \ldots, p_m))$*

Exemple 24 *Le tuples $(x^2, x^3) \leq (x^2, 1)$ alors que (x^2, x^3) et (x, x^4) sont incomparables.*

Nous associons à ce quasi-ordre de manière naturelle une relation d'équivalence.

Définition 40 (Relation d'équivalence) *Soit deux m-tuples de polynômes $(p_1, \ldots, p_m)$ et $(q_1, \ldots, q_m)$, nous définissons la relation d'équivalence $\sim$ entre deux tuples de polynômes tel que $(p_1, \ldots, p_m) \sim (q_1, \ldots, q_m)$ si et seulement si $(p_1, \ldots, p_m) \leq (q_1, \ldots, q_m)$ et $(p_1, \ldots, p_m) \leq (q_1, \ldots, q_m)$.*

Exemple 25 *Le tuples $(4x^2, x^3)$ et $(x^2, 2x^3)$ sont équivalents car $(4x^2, x^3) \leq (x^2, 2x^3)$ et $\neg((x^2, 2x^3) \leq (4x^2, x^3))$.*

Remarquons que deux tuples de polynômes $(p_1, \ldots, p_m)$ et $(q_1, \ldots, q + m)$ sont équivalents si et seulement si pour tout i, $deg(p_i) = deg(q_i)$. De plus comme $\mathbb{Z}/2\mathbb{Z}[h]$ est un corps fini (tous les coefficients sont dans $\mathbb{Z}/2\mathbb{Z}$), le nombre de classes d'équivalence est fini.

Nous rappelons d'abord le lemme de Dickson [Dic13]

Lemme 32 (Lemme de Dickson) *Toute séquence finie de tuples distincts d'entiers positifs contient au moins deux (réellement une infinité) tuples comparables.*

Fait 1 *Le nombre de solutions minimales non nulles d'un système d'équation (E) est fini.*

Preuve : Nous montrons par l'absurde qu'il existe une nombre fini de classes d'équivalence de solution de (E) grâce au lemme de Dickson [Dic13]. Supposons que le nombre de classes d'équivalence de solutions minimales non nulles de (E) soit infini, il existe alors une séquence infinie de tuples incomparables d'entiers positifs correspondant au degré des polynômes ce qui implique une contradiction. Comme nous sommes dans un corps fini $\mathbb{Z}/2\mathbb{Z}[h]$ nous savons qu'il existe une nombre fini de tuples de polynômes plus petit qu'un tuple de polynômes fixé et donc que chaque classe d'équivalence à un nombre fini d'éléments. □

Fait 2 *Une solution de (E) est la somme d'une solution de (E) et d'une solution de (H).*

Preuve : La différence de deux solutions de (E) est une solution de (H), or nous considérons la théorie ACUNh et donc la somme de deux solutions de (E) est une solution de (H). □

Notons que la somme de deux solutions de (H) est une solution de (H).

Fait 3 *Toutes solutions σ de (H) est une combinaison linéaire de solutions minimales de (H).*

Preuve : Soit σ une solution de (E) un système de n équations à m inconnues, σ est un m-tuple de polynômes de $\mathbb{Z}/2\mathbb{Z}[h]$. Nous prouvons le résultat par induction sur la taille des solutions.

Cas de base : Si σ une solution minimale de (H) un système homogène de n équations, le résultat est immédiat.

Induction : soit σ une solution non minimale de (H) un système de m équations. L'hypothèse de récurrence nous dit que pour toutes solutions plus petites que σ, suivant l'ordre défini précédemment, ce sont des combinaisons linéaires de solutions minimales de (H).

Soit τ une solution minimale de (H) telle que $\tau < \sigma$.

Soit $d = min(deg(\sigma_i) - deg(\tau_i) | 1 \leq i \leq m)$, nous définissons σ' tel que :

$$\sigma'(X_i) = \sigma(X_i) - h^d.\tau X_i \forall 1 \leq i \leq m$$

σ' est bien une solution de (H) car nous que si σ' est une solution et q est un polynôme de $\mathbb{Z}/2\mathbb{Z}[h]$ alors $q\sigma'$ est toujours une solution de (E) et que la différence de deux solutions de (H) est encore une solution de (H).
Nous montrons maintenant que $\sigma' < \sigma$. Comme $\tau < \sigma$ nous savons qu'il existe au moins un polynôme de τ qui a un degré strictement plus petit que le polynôme de σ. Par définition de d le degré d'un polynôme de σ va décroître et les autres degrés des polynômes de σ n'augmenterons pas, donc $\sigma' < \sigma$. Remarquons que même dans le cas ou $d = 0$ la différence entre τ et σ décroît. Par hypothèse d'induction nous savons alors que τ est une combinaison linéaire de solutions minimales de (H). De plus nous pouvons écrire σ ainsi : $\sigma = \sigma' \oplus h^d.\tau \forall 1 \leq i \leq m$. Donc σ est bien une combinaison linéaire de solutions minimales de (H).
□

Fait 4 *Une solution de (E) est la somme d'une solution minimale σ_0 de (E) et d'une solution de (H).*

Preuve : C'est une conséquence immédiate du fait 2 page précédente et du fait 3 page précédente
□

8.2.1.2 Des équations diophantiennes à l'unification ACUNh.

Dans le reste de cette section nous prouvons que le problème d'unification ACUNh avec constantes est unitaire. Ceci est une conséquence de la construction de l'unificateur le plus général grâce à la résolution d'équations diophantiennes linéaires.

Nous considérons un ensemble fini de symboles de constantes dénoté par $\Sigma_C = \{c_1, \ldots, c_k\}$. Nous nous intéressons au problème d'unification constitué d'un ensemble d'équations de la forme $s_j = t_j$ pour $j = 1, \ldots, m$ où s_j, t_j sont des termes construits à partir des éléments de Σ_C, le symbole d'homomorphisme h, l'opérateur binaire $\oplus$, et la constante 0. Soit $x_1, \ldots, x_n$ l'ensemble des variables présentes dans ce problème d'unification. Le problème d'unification est équivalent à la résolution du système d'équation (U) suivant :

$$\sum_{i=1}^{i=n} A_{i,j} \odot x_i = b_j \quad \text{pou} \quad j = 1, \ldots, m \tag{U}$$

où les $A_{i,j}$ sont des polynômes de $\mathbb{Z}/2\mathbb{Z}[h]$, les b_j des termes clos, et les variables x_i. Nous considérons le système homogène (HU) associé au système (U) en remplaçant les b_i par 0.

$$\sum_{i=1}^{i=n} A_{i,j} \odot x_i = 0 \quad \text{for} \quad j = 1, \ldots, m \tag{HU}$$

Exemple 26 *Considérons la signature suivante $\Sigma = \{h, \oplus, c_1, c_2\}$, $s = h(X_1) \oplus X_2 \oplus h(c_1)$ et $t = X_1 \oplus h^2(X_2) \oplus c_2$ deux termes, alors l'unification de s et t revient à résoudre :*
$h(X_1) \oplus X_2 \oplus h(c_1) = X_1 \oplus h^2(X_2) \oplus c_2 \Leftrightarrow h(X_1) \oplus X_1 \oplus h^2(X_2) \oplus X_2 = h(c_1) \oplus c_2$

Nous notons $Sol(U)$ (resp. $Sol(HU)$) l'ensemble des substitutions closes qui sont des solutions de (U) (resp. (HU)). Une solution close de (U) est obtenue de la manière suivante :

Nous décomposons chaque terme du membre droit de (U) comme suit :

$$b_j = \sum_{i=1}^{i=k} B_j^i \odot c_j$$

Pour tout $i = 1, \ldots, k$, nous définissons le système d'équation (E_i) suivant :

$$\sum_{i=1}^{i=n} A_{i,j}.X_i = B_j^i \quad \text{pour} \quad j = 1, \ldots, m \tag{E_i}$$

où les variables X_i sont des polynômes de $\mathbb{Z}/2\mathbb{Z}[h]$. Si σ_i est une solution de (E_i) pour chaque $i = 1, \ldots, k$, alors nous obtenons une solution σ de (U) par le calcul suivant :

$$\sigma(x_j) = \sum_{i=1}^{i=k} \sigma_i(X_j) \odot c_j$$

L'unificateur le plus général de (HU) est obtenu en calculant l'ensemble fini des solution minimales du système (HE), ceci grâce au Fait 1 page 129.

$$\sum_{i=1}^{i=n} A_{i,j}.X_i = 0 \quad \text{pour} \quad j = 1, \ldots, m \tag{HE}$$

où les variables X_i sont des polynômes de $\mathbb{Z}/2\mathbb{Z}[h]$. Nous notons l'ensemble fini des solutions de (HE) par $(\sigma_k)_{k \in I_\mu}$.

Fait 5 *Le problème (HU) a un unificateur le plus général σ_H défini par $x_i\sigma_H = \Sigma_{k \in I_\mu} P_{i,k} \odot y_k$ où $\sigma_k = \{X_1 \leftarrow P_{1,k}, \ldots, X_n \leftarrow P_{n,k}\}$ avec $P_{i,k} \in \mathbb{Z}/2\mathbb{Z}[h]$, et où les y_k sont des variables fraîches.*

Preuve :

– Premièrement nous prouvons que σ_H est une solution de (HU). Pour $j = 1, \ldots, m$, nous avons :

$$\begin{array}{rcl}
\Sigma_{i=1}^{i=n} A_{i,j} x_i \sigma_H & = & \Sigma_{i=1}^{i=n} A_{i,j}(\Sigma_{k \in I_\mu}(P_{i,k} \odot y_k)) \\
& = & \Sigma_{i=1}^{i=n} \Sigma_{k \in I_\mu}(A_{i,j}(P_{i,k} \odot y_k)) \\
& = & \Sigma_{i=1}^{i=n} \Sigma_{k \in I_\mu}(A_{i,j} P_{i,k}) \odot y_k) \\
& = & \Sigma_{k \in I_\mu}((\Sigma_{i=1}^{i=n} A_{i,j} P_{i,k}) \odot y_k) \\
& = & 0
\end{array}$$

– Deuxièmement nous prouvons que toute solution σ de (HU) est une instance de σ_H. Soit $\mathcal{Z}$ l'ensemble des variables de $x_i\sigma$. Pour $i = 1, \ldots, n$ nous avons $x_i\sigma = \Sigma_{c \in \Sigma_C} X_i^c \odot c + \Sigma_{z \in \mathcal{Z}} Z_i^z \odot z$ et $(x_1\sigma, \ldots, x_n\sigma)$ est une solution de (HU) si et seulement si pour chaque $c \in \Sigma_C$, pour chaque $z \in \mathcal{Z}$ $(X_1^c, \ldots, X_n^c)$ et $(Z_1^z, \ldots, Z_n^z)$ sont solutions de (HE).
Par conséquent chaque $c \in \Sigma_C$, $(X_1^c, \ldots, X_n^c)$ est une combinaison linéaire d'une solution minimale de (HE), *i.e.* $X_i^c = \Sigma_{k \in I_\mu} Q_k^c P_{i,k}$ pour $i = 1, \ldots, n$ où les Q_k^c sont des coefficients de la combinaison linéaire.
Pour chaque $z \in \mathcal{Z}$, nous avons pour $(Z_1^z, \ldots, Z_n^z)$ que $Z_i^z = \Sigma_{k \in I_\mu} R_k^z P_{i,k}$. Nous en déduisons que pour $i = 1, \ldots, n$,

$$\begin{array}{rcl}
x_i\sigma & = & \Sigma_{c \in \Sigma_C}(\Sigma_{k \in I_\mu} Q_k^c P_{i,k}) \odot c + \Sigma_{z \in \mathcal{Z}}(\Sigma_{k \in I_\mu} R_k^z P_{i,k}) \odot z \\
& = & \Sigma_{k \in I_\mu}((\Sigma_{c \in \Sigma_C} P_{i,k} \odot (Q_k^c \odot c)) + (\Sigma_{z \in \mathcal{Z}} P_{i,k} \odot (R_k^z \odot z))) \\
& = & \Sigma_{k \in I_\mu} P_{i,k} \odot (\Sigma_{c \in \Sigma_C} Q_k^c \odot c + \Sigma_{z \in \Sigma_C} R_k^z \odot z) \\
& = & \Sigma_{k \in I_\mu} P_{i,k} \odot (\Sigma_{c \in \Sigma_C} Q_k^c \odot c + \Sigma_{z \in \mathcal{Z}} R_k^z \odot z)
\end{array}$$

ce qui termine la preuve en choisissant $y_k = \Sigma_{c \in \Sigma_C} Q_k^c \odot c \oplus \Sigma_{z \in \mathcal{Z}} R_k^z \odot z$.

□

Fait 6 *Soit σ une solution close de (U) et σ_H l'unificateur le plus général de (HU). La substitution $\sigma \oplus \sigma_H$ est un unificateur le plus général de (U).*

Preuve : C'est une conséquence immédiate du Fait 2 page 129 et du Fait 5 page précédente. □

Fait 7 *Le problème d'unification avec constantes pour la théorie équationnelle* ACUNh *est unitaire.*

Preuve : Le Fait 6 page précédente implique immédiatement le résultat. □

8.2.2 Un algorithme d'unification élémentaire.

Pour construire un tel algorithme, comme nous l'avons vu dans la section précédente il suffit de calculer l'ensemble des solutions minimales du système (E). Nous construisons donc une méthode de calcul pour l'ensemble des solutions minimales du système homogène d'équations diophantiennes (HE) et une solution minimale particulière du système (E). Une approche possible pour cela serait d'utiliser une technique similaire à celle utilisée pour l'unification AC. Nous utilisons ici une approche basée sur les automates afin d'obtenir un résultat plus général. Nous dénotons par $\langle \mathbb{Z}/2\mathbb{Z}[h], \lesssim, \oplus, 0, h \rangle$ la structure composée de la théorie universelle sur $\mathbb{Z}/2\mathbb{Z}[h]$ avec la relation d'ordre $\lesssim$ et les symboles de constantes, $\oplus$, 0, et h. Nous montrons que la théorie du premier ordre pour cette structure est décidable, car c'est une *structure automatique* [BG00].

Pour cela, nous montrons comment construire un automate qui accepte l'ensemble minimal des solutions associées au problème d'unification.

Lemme 33 *La théorie du premier ordre pour la structure* $\langle \mathbb{Z}/2\mathbb{Z}[h], \lesssim, \oplus, 0, h \rangle$ *est décidable.*

Preuve : Nous montrons que la structure $\langle \mathbb{Z}/2\mathbb{Z}[h], \lesssim, \oplus, 0, h \rangle$ est automatique, et grâce au résultat sur les structures automatiques [BG00] nous concluons la décidabilité.

Nous représentons les polynômes de $\mathbb{Z}/2\mathbb{Z}[h]$ par un mot de $\{0,1\}^*$. Soit un polynôme $p \in \mathbb{Z}/2\mathbb{Z}[h]$ tel que $p(h) = \sum_{i=1}^{i=n} b_i h \in \mathbb{Z}/2\mathbb{Z}[h]$, où les $b_i = 1$, nous désignons par $b_0 \cdots b_n$ le mot associé à p, en ordonnant les bits de poids faible en premier. Le polynôme 0 est représenté par le mot 0. L'image par ν d'un polynôme est reconnaissable par une expression de $0 \cup \{0,1\}^*$.

Nous donnons les automates qui acceptent les représentations des tuples des polynômes de $\mathbb{Z}/2\mathbb{Z}[h]$ en fonction des relations de la structure $\langle \mathbb{Z}/2\mathbb{Z}[h], \lesssim, \oplus, 0 \rangle$. La constante 0 est remplacée par la relation un-aire $X_1 = 0$, et la fonction h par la relation suivante $X_1 = h(X_2)$. Nous complétons la fin des mots par des 0 si nécessaire, comme il est d'usage dans les automates. Nous présentons donc les automates qui décrivent les relations de la structure :

- L'automate pour $X_1 = 0$ est simple et n'est pas écrit ici.
- L'automate pour $X_1 \lesssim X_2$ est décrit dans la Figure 8.2.

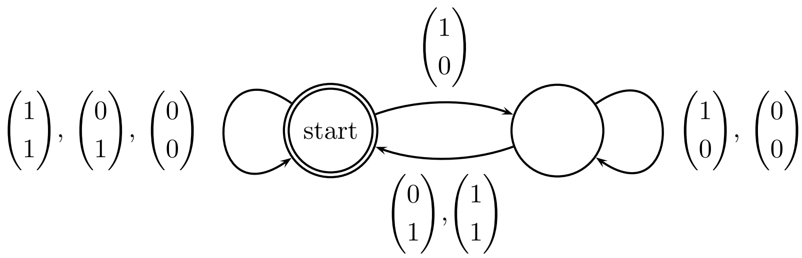


Fig. 8.2 – Automate pour $X_1 \lesssim X_2$.

- L'automate pour $X_1 = X_2 \oplus X_3$ est donné dans la Figure 8.3 page ci-contre. Cet automate est plus simple que celui de l'addition dans l'arithmétique de Presburger car nous n'avons pas à nous soucier de la retenue.

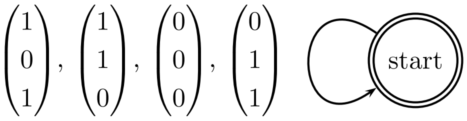


FIG. 8.3 – Automate pour $X_1 = X_2 \oplus X_3$.

– L'automate qui accepte les paires (X_1, X_2) telle que $X_1 = h(X_2)$ est un peu plus complexe. Nous devons nous souvenir de l'ancienne valeur de X_2 pour faire la comparaison. Cet automate est donné par la Figure 8.4.

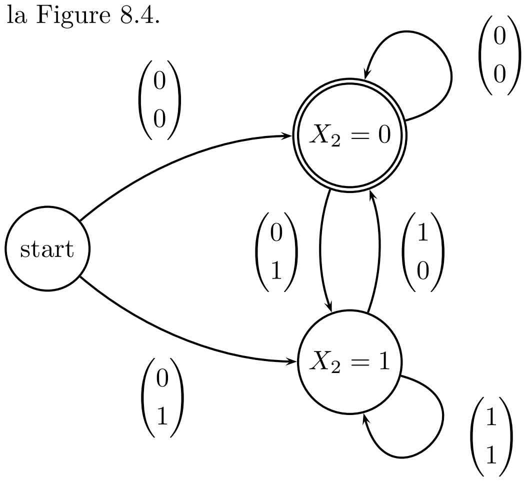


FIG. 8.4 – Automate pour $X_1 = h(X_2)$.

$\square$

Nous pouvons donc calculer les solutions pour un système linéaire en construisant l'automate associé au système et de calculer la solution minimale grâce au fait 8.

Fait 8 *L'ensemble des solutions minimales du système homogène d'équations diophantiennes est calculable.*

Preuve : Un vecteur $\overline{X}$ est une solution minimale d'un système d'équations diophantiennes $\phi(\overline{X})$ si et seulement si $\overline{X}$ vérifie la formule suivante :

$$\phi(\overline{X}) \wedge \forall \overline{Y} \left(\overline{Y} \lesssim \overline{X} \wedge \phi(\overline{Y}) \implies \overline{X} \lesssim \overline{Y} \right)$$

Il existe bien un automate calculable qui accepte l'ensemble des éléments de $\overline{X}$ satisfaisant cette formule. Car il n'y a qu'un nombre fini de solutions et que le langage de cet automate est fini. Nous générons donc l'ensemble de tous les termes acceptés par cet automate pour obtenir l'ensemble des solutions minimales. $\square$

8.2.3 Général unification ACUNh.

Nous regardons maintenant le problème d'unification ACUNh où Σ est augmentée par des symboles libres. Nous utilisons l'algorithme proposé par F. Baader *et alii* [BS96].

8.2.3.1 Un algorithme d'unification.

Pour appliquer l'algorithme de combinaison de F. Baader *et alii* [BS96], nous devons montrer que l'unification avec restriction linéaire constante est finitaire. Soit un problème d'unification (*i.e.* un ensemble fini d'équations de la forme $s_i = t_i$), nous associons à chaque constante c apparaissant dans le problème un ensemble V_c de variables qui sont les variables dans lesquelles c ne doit pas apparaître.

Supposons que nous avons un ordre linéaire $<$ sur l'ensemble de constantes Σ_C et de variables $\mathcal{X}$, alors nous définissons $V_c = \{x \in \mathcal{X} \mid x < c\}$. Un problème d'unification avec une restriction linéaire sur les constantes est un problème d'unification avec l'ajout de la contrainte de restriction qui correspond à donner un ordre $<$. Ceci pour garantir que chaque variable X du problème peut être instancié juste par un terme constant c tel que $X \notin V_c$. Cet ensemble est calculable et fini, et nous pouvons écrire $X = \Sigma_{\{X \notin V_c\}} X_{i,c} \odot c$ pour un polynôme $X_{i,c}$ de $\mathbb{Z}/2\mathbb{Z}[h]$. Ainsi les problèmes d'unification avec restriction linéaire de constantes sont résolus de la même manière que les problèmes d'unification.

En appliquant l'algorithme de combinaison de F. Baader [BS96] nous obtenons un algorithme d'unification pour la théorie ACUNh dans Σ étendu avec les symboles libres.

8.2.3.2 Un résultat technique d'unification.

Pour prouver le lemme 34 page suivante, nous reprenons les notations et algorithmes introduits par F. Baader [BS96] pour son algorithme de combinaison.

Nous considérons maintenant $\mathcal{F} = \Sigma \uplus \Sigma'$ où Σ' est un ensemble de symboles qui contient au moins un symbole d'arité supérieur ou égal à 2. Les notations sont étendus comme suit : $t = C[t_1, \ldots, t_n]$ si C est un contexte de symbole de Σ simplement et les t_i sont standards, ou si C est un contexte de symboles de Σ' et les t_i ne sont pas standards.

Définition 41 (Terme pur) *Si un terme t contient que des symboles de Σ et des variables, ou s'il ne contient que des symboles de Σ' et des variables, il est dit* pur*.*

Définition 42 ($\#(t)$) *Le nombre d'alternation de théorie d'un terme est défini par induction :*
- $\#(t) = 0$ *si t est pur*
- *sinon* $\#(C[t_1, \ldots, t_n]) = 1 \oplus max\{\#(t_i) \mid i = 1, \ldots, n\}$

Définition 43 (Termes étrangers) *L'ensemble $AF(t)$ de facteur étrangers (« alien ») de t est défini :*
- $AF(t) = \{t\}$ *si t est pur*
- $AF(t = C[t_1, \ldots, t_n]) = \{t\} \cup AF(t_1) \cup \ldots \cup AF(t_n)$

Nous rappelons la définition de sous-termes qui sera utilisée dans le chapitre 10 page 145 dans la procédure de décision pour le problème de sécurité pour la théorie équationnelle ACUNh.

Définition 44 (Sous-termes) *L'ensemble $St_{\mathsf{E}}(t)$ de* sous-termes *de t est le plus petit ensemble de termes tel que :*
- $t \in St_{\mathsf{E}}(t)$,
- *Si $f(t_1, \ldots, t_n) \in St_{\mathsf{E}}(t)$ et $f(t_1, \ldots, t_n)$ est un terme standard alors $t_1, \ldots t_n \in St_{\mathsf{E}}(t)$,*
- *Si $s \in St_{\mathsf{E}}(t)$ et s n'est pas un terme standard alors $Fact_{\mathsf{E}}(s) \subseteq St_{\mathsf{E}}(t)$.*

Exemple 27 *Soit $t_1 = h^2(a) \oplus b \oplus x$ et $t_2 = h(\langle a, b\rangle) \oplus x$, alors $St_{\mathsf{E}}(t_1) = \{t_1, a, b, x\}$ et $St_{\mathsf{E}}(t_2) = \{t_2, \langle a, b\rangle, a, b, x\}$.*

Lemme 34 *Soit P un problème d'unification dans la théorie $E =$ ACUNh (avec des fonctions libres de symboles) et θ un mgu_{E} de P. Alors pour tout $x \in dom(\theta)$ et $v \in St_{\mathsf{E}}(x\theta) \setminus vars(x\theta)$ il existe $t \in St_{\mathsf{E}}(P)$ tel que $v =_{\mathsf{E}} t\theta$.*

Nous prouvons le résultat pour l'ensemble d'unificateurs calculé par l'algorithme de combinaison de F. Baader et K. U. Schulz [BS96].

Preuve :

Premièrement, nous remarquons que le lemme est vrai pour une unification avec restriction linéaire de constantes pures. C'est vrai pour la théorie vide, et pour ACUNh c'est une conséquence de nos résultats sur l'unification : une solution d'un système d'équations $\bigoplus_{i\in I} P_i(h) \odot X_i \oplus \bigoplus_{j\in J} Q_j(h) \odot c_j = 0$ avec restriction linéaire de constantes est une combinaison linéaire avec des variables et des constantes c_i.

Pour généraliser l'union de théorie, nous analysons l'algorithme de combinaison. Nous rappelons brièvement les principes de cet algorithme (non-déterministe) :

(1) Remplacer chaque terme non pur $t = C[t_1, \ldots, t_n]$ par $C[X_{t_1}, \ldots, X_{t_1}]$ et ajouter les équations $X_{t_i} = t_i$ où les X_{t_i} sont des nouvelles variables.

(2) Remplacer chaque équation $s = t$ telle que s, t sont des termes purs mais pas dans la même théorie par $X_{s,t} = t \wedge X_{s,t} = s$ où $X_{s,t}$ est une nouvelle variable.

(3) Choisir une partition de l'ensemble de variables $\mathcal{X}_1, \ldots, \mathcal{X}_p$, pour chaque $\mathcal{X}_i$ choisir un représentant X_i et remplacer toutes les variables $X \in \mathcal{X}_i$ par X_i (ceci par l'ajout des équations $X_i = X$ pour tout $X \in \mathcal{X}_i$).

(4) Étiqueter chaque variable par Σ ou Σ' de manière non-déterministe, et choisir un ordre linéaire $X_1 < \ldots < X_n$.

(5) Le problème se décompose en un problème d'unification pure avec une restriction linéaire des constantes (sinon retourner faux).Chaque problème est résolu en considérant les variables de l'autre théorie comme des constantes. L'unificateur est donné par la combinaison de solutions des deux problèmes d'unification

Nous utilisons les propriétés suivantes de l'algorithme. Supposons que l'algorithme retourne une substitution θ.

– Pour chaque paire de variables X, Y dans la même classe d'équivalence $X\theta = Y\theta$.
– Pour chaque facteur étranger $t = C[t_1, \ldots, t_n]$ de P, il existe des variables $X_t, X_{t_1}, \ldots, X_{t_n}$ telles que $X_t\theta = t\theta = C[X_{t_1}\theta, \ldots X_{t_n}\theta]$.
– Pour les variables $X_{s,t}$, nous avons $X_{s,t}\theta = s\theta = t\theta$.
– Pour chaque terme $C[X_1, \ldots, X_l]$ apparaissant dans le problème final du problème d'unification pur, il existe Y_{t_1} dans la même classe d'équivalence que X_1,..., Y_{t_l} dans la même classe d'équivalence que X_l tel que $C[t_1, \ldots, t_l]$ est un facteur étranger de P.

La solution du problème d'unification pur a la forme suivante : $X = C'[X_1, \ldots, X_n]$ ou X est une combinaison de variables nouvelles et les variables X_i et les constantes de Σ. Dans tous les cas les facteurs de $X\theta$ pour une variable X du problème initial sont, soit $X\theta$, soit un $X_t\theta$ pour une variable X_t par conséquent il y a un $t\theta$ pour un facteur t de P ou d'une nouvelle variable. $\square$

L' algorithme combinaison calcule l'ensemble fini et complet des unificateurs $CS(P)$. Pour trouver l'ensemble des *mgu*, nous devons détecter les unificateurs qui seront subsumés par d'autres pour obtenir $CS(P)$. Cette étape ne change pas le résultat.

8.3 Disunification ACUNh.

Nous nous intéressons maintenant au problème de disunification, problème simple dans notre cas qui a déjà été étudié par F. Baader et que nous redémontrons ici.

Nous dénotons par $=_{\mathsf{AC}}$ l'égalité modulo les axiomes d'associativité et de commutativité et par $=$ l'égalité modulo les axiomes de ACUNh.

Lemme 35 *Soit* $t_1, s_1, \ldots, t_n, s_n \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ *des termes en forme normale alors les deux assertions suivantes sont équivalentes :*

1. *Il n'existe pas de substitution close* σ *telle que* $t_1\sigma \neq s_1\sigma \wedge \ldots \wedge t_n\sigma \neq s_n\sigma$
2. $t_i =_{\mathsf{AC}} s_i$ *pour un* $i \in \{1, \ldots, n\}$

Preuve :
(*2* $\Rightarrow$ *1*) Si $t_i =_{\mathsf{AC}} s_i$ pour un i alors $t_i = s_i$ et $t_i\sigma = s_i\sigma$ pour toutes substitutions. Par conséquent il n'existe pas de substitution σ telle que $t_1\sigma \neq s_1\sigma \wedge \ldots \wedge t_n\sigma \neq s_n\sigma$.
(*1* $\Rightarrow$ *2*) Soit $T = \{s_1, \ldots, s_n, t_1, \ldots, t_n\}$, si $t_i \neq_{\mathsf{AC}} s_i$ pour tout $i, 1 \leq i \leq n$, alors nous résonnons par induction sur le nombre de variables de $vars(T)$.

- Si $n = 0$ alors les s_i et les t_i sont clos. Par conséquent, la substitution vide ϵ satisfait $t_1\epsilon \neq s_1\epsilon \wedge \ldots \wedge t_n\epsilon \neq s_n\epsilon$
- Si $n > 0$ alors soit $x \in vars(T)$, et soit g un terme clos tel que :
  1. est différent de 0,
  2. ne contient pas de symbole $\oplus$, et
  3. n'est pas un sous-terme syntaxique de terme de T.

  Soit $t'_i = t_i[x \mapsto g]$ et $s'_i = s_i[x \mapsto g]$ pour $1 \leq i \leq n$. Nous avons que $t'_i \neq_{AC} s'_i$ pour tout i, et tous les t'_i et s'_i sont termes en forme normale. Comme $\{t'_1, \ldots, t'_n, s'_1, \ldots, s'_n\}$ contient $n-1$ variables il existe par hypothèse d'induction une substitution σ' tel que $t'_1\sigma' \neq s'_1\sigma' \wedge \ldots \wedge t'_n\sigma' \neq s'_n\sigma'$. Par conséquent, $\sigma = \sigma' \circ [x \mapsto g]$ nous obtenons $t_1\sigma \neq s_1\sigma \wedge \ldots \wedge t_n\sigma \neq s_n\sigma$.

□

Lemme 36 *Le fragment existentiel* Σ_1 *de la théorie du premier ordre de* $\mathcal{T}(\mathcal{F})$ *modulo ACUNh est décidable.*

Preuve : Soit une formule existentielle close $\phi = \exists\bar{x}\psi$, où ψ est formule sans quantifier, soit $c_1 \vee \ldots \vee c_n$ une formule disjonctive en forme normale de ψ. La validité de la formule ϕ est équivalent à la validité de $\exists\bar{x}c_i$. Soit

$$c_i = (r_1 = u_1 \wedge \ldots \wedge r_m = u_m \wedge s_1 \neq t_1 \wedge \ldots \wedge s_n \neq t_n)$$

Cette formule est satisfiable s'il existe un unificateur plus général μ de $r_1 = u_1 \wedge \ldots \wedge r_m = u_m$ tel que la formule suivante est satisfiable :

$$s_1\mu \neq t_1\mu \wedge \ldots \wedge s_n\mu \neq t_n\mu$$

Il existe un ensemble fini d'unificateur plus général μ calculable et la satisfiablilité de la disunification est décidable grâce au lemme 35. □

Chapitre 9

Résolution de systèmes «dépendants» d'équations.

« Mathematics is the queen of the sciences. »
Carl Friedrich Gauss.

Sommaire

Il existe de nombreux travaux en mathématiques et en informatique pour résoudre des systèmes d'équations. Dans ce chapitre, nous présentons une nouvelle méthode de résolution pour des systèmes d'équations particulieres quadratiques. Résoudre de tels systèmes est en général indécidable. Nous rappelons d'abord quelques notions mathématiques nécessaires. Nous définissons ensuite les « systèmes dépendants d'équations », ces systèmes quadratiques caractérisés par une dépendance linéaire entre certaines variables proviennent du problème de sécurité pour un intrus actif. Nous montrons ensuite qu'il est possible de résoudre des systèmes dépendants d'équations. Nous appliquerons cette technique de résolution à la vérification de protocoles cryptographiques dans le chapitre 10 page 145.

Nous choisissons de nous intéresser plus particulièrement à $\mathbb{Z}/2\mathbb{Z}[h]$. Nous pouvons obtenir les mêmes résultats pour $\mathbb{Z}$, $\mathbb{Z}/p\mathbb{Z}$ ou $\mathbb{Z}/p\mathbb{Z}[h]$. Malheureusement, la proposition 12 page suivante n'est pas vraie pour $\mathbb{Z}[h]$, mais reste vraie pour $\mathcal{Z}$, $\mathbb{Z}/p\mathbb{Z}$ et $\mathbb{Z}/p\mathbb{Z}[h]$. Cette proposition joue un rôle crucial dans notre procédure de résolution des systèmes « dépendants » d'équations, par conséquent notre procédure ne traite pas ce cas là. Notons juste que pour $\mathbb{Z}[h]$ S. Delaune [Del06b] a montré que le problème de sécurité pour un intrus actif est indécidable dans le cas $\mathbb{Z}[h]$.

9.1 Rappels mathématiques.

Nous utiliserons dans la suite, la division euclidienne et les déterminants sur les matrices, notions usuelles que nous ne rappelons pas ici.

9.1.1 Notations et définitions.

Soit $\sum_{i=0}^{n} b_i h^i$ où $b_i \in \mathbb{Z}/2\mathbb{Z}$ un polynôme de $\mathbb{Z}/2\mathbb{Z}[h]$, le produit $\odot$ d'un polynôme par un terme t est défini comme suit :

$$(\sum_{i=0}^{n} b_i h^i) \odot t = \sum_{i=0 \ | \ b_i \neq 0}^{n} h^i(t)$$

Exemple 28 *Le produit du polynôme $h^2 + 1$ par le terme $x + a$ donne :*

$$(h^2 + 1) \odot (x + a) = h^2(x) + x + h^2(a) + a$$

Définition 45 (Vecteurs indépendants) *Soit $\mathcal{V} = \{v_1, \ldots, v_m\}$ un ensemble de vecteurs. $\mathcal{V}$ est un ensemble de vecteurs* indépendants *s'il existe α_i tel que si $\alpha_1 v_1 + \ldots + \alpha_m v_m = 0$ alors $\alpha_i = 0$ pour tout $1 \leq i \leq m$. Un ensemble de vecteurs non indépendants sont dits* dépendants *ou* liés*.*

Définition 46 (Degré) *Soit $p = \sum_{i=0}^{n} b_i h^i$ où $b_i \in \mathbb{Z}/2\mathbb{Z}$ un polynôme de ZZ, le degré de p est défini comme $deg(p) = max\{i | b_i \neq 0\}$.*

9.1.2 Résultats mathématiques.

Nous rappelons ici les deux résultats mathématiques que nous utiliserons par la suite, pour résoudre les systèmes dépendants d'équations.

Proposition 12 *Soit P un polynôme de $\mathbb{Z}/2\mathbb{Z}[h]$, il n'existe qu'un nombre fini de polynômes $Q \in \mathbb{Z}/2\mathbb{Z}[h]$ tels que $P < Q \leq 0$.*

Preuve : Le degré sur les polynômes constitue un ordre sur les polynômes, *i.e.* soit p et q deux polynômes de $\mathbb{Z}/2\mathbb{Z}[h]$, $p = q$ si $deg(p) = deg(q)$ et $p < q$ si $deg(p) < deg(q)$. Nous utilisons cet ordre pour prouver le résultat par induction sur le degré des polynômes :

Cas de Base : pour le polynôme nul, le résultat est évidemment vrai.

Induction : Supposons que pour tous polynômes P de degré d, il n'existe qu'un nombre fini de polynômes $Q \in \mathbb{Z}/2\mathbb{Z}[h]$ tels que $P < Q \leq 0$. Considérons un polynôme P' de degré $d+1$, alors $P' = b_d h^d + P$ où $b_d \in \mathbb{Z}/2\mathbb{Z}$ et P un polynôme de degré d. Comme b_d vaut 0 ou 1, il n'existe qu'un nombre fini de polynômes $Q \in \mathbb{Z}/2\mathbb{Z}[h]$ tels que $P' < Q \leq 0$, plus précisement deux fois le nombre de polynômes présents entre P et 0.

□

Proposition 13 *Soit A une matrice $n \times m$ sur $\mathbb{Z}/2\mathbb{Z}[h]$ telle que les n vecteurs lignes soient indépendants ($n \leq m$) alors :*

$$\exists Q \in \mathbb{Z}/2\mathbb{Z}[h], \forall b \in \mathbb{Z}/2\mathbb{Z}[h]^n, \exists X \in \mathbb{Z}/2\mathbb{Z}[h]^m \quad A \cdot X = Q \cdot b \tag{9.1}$$

De plus Q est le déterminant d'une sous-matrice de A.

Preuve : Soit $\tilde{A}$ une matrice $m \times m$ construite à partir de la matrice A en ajoutant des lignes telles que toutes les lignes de $\tilde{A}$ restent indépendantes. Soit Q le déterminant de $\tilde{A}$, $Q = det(\tilde{A})$, comme les lignes de $\tilde{A}$ sont indépendantes le déterminant est non nul *i.e.* , $Q \neq 0$. Il est donc possible de calculer l'inverse de $\tilde{A}$, noté $\tilde{A}^{-1}$. Par construction de l'inverse d'une matrice $\tilde{A}^{-1} = \frac{1}{Q}.A'$, où A' est une matrice à coefficients dans $\mathbb{Z}/2\mathbb{Z}[h]$ (plus précisément A' est la transposée de la co-matrice de A). Soit $b \in \mathbb{Z}/2\mathbb{Z}[h]^n$ et $\tilde{b} \in \mathbb{Z}/2\mathbb{Z}[h]^m$ le vecteur b complété par des coefficients arbitraires pour obtenir un vecteur de taille m. Soit $\tilde{X} = \tilde{A}^{-1}.Q.\tilde{b}$ une solution particulière du système suivant $\tilde{A} \cdot \tilde{X} = Q.\tilde{b}$. Or X est obtenu à partir de $\tilde{X}$ en ne gardant que les n premières lignes, donc X est aussi une solution de $A \cdot X = Q.b$. $\square$

9.2 Systèmes « dépendants » d'équations.

Nous notons $\mathcal{X} = \{x_1, \ldots, x_n\}$ l'ensemble des variables utilisées dans un système d'équations.

Définition 47 (Système monotone quasi-quadratique d'équations) *Un système d'équations est un* système monotone quasi-quadratique d'équations *s'il est de la forme suivante :*

$$\begin{aligned}
&z[1,1] \odot t_1 + \ldots + z[1,n] \odot t_n = u_1 \\
&z[2,1] \odot t_1 + \ldots + z[2,n] \odot t_n + z[2,n+1] \odot t_{n+1} = u_2 \\
&\vdots \\
&z[k,1] \odot t_1 + \ldots + z[k,n] \odot t_n + \ldots + z[k,n+k-1] \odot t_{n+k-1} = u_k
\end{aligned}$$

Où les $t_1, \ldots, t_n$ sont de termes constants, les $t_{n+1}, \ldots, t_{n+k-1}$, $z[i,j]$, et les u_i, pour $1 \leq i \leq k$ et $1 \leq j \leq n+i-1$, peuvent s'écrire : $t_i^{x_1} \odot x_1 + \ldots t_i^{x_p} \odot x_p + t_i^0$ où les x_i sont des variables de $\mathcal{X}$, les $t_i^{x_v}$ sont des éléments de $\mathbb{Z}/2\mathbb{Z}[h]$ et t_i^0 est un terme constant. Les variables $z[i,j]$ sont appelées variables de contexte *et nous dénotons l'ensemble de toutes les variables de contexte par $\mathcal{Z} = \{z[i,j] / i = 1, \ldots, k, j = 1, \ldots, n+k-1\}$.*

- *le système d'équations est dit* quasi-quadratique *car la première équation n'est pas une équation quadratique mais linéaire.*
- *le système est dit* monotone *car chaque membre gauche est croissant d'équation en équation, il reprend à chaque fois les mêmes termes constants qu'à l'équation précédente avec de nouvelles variables de contexte et seule est ajouté le produit d'une nouvelle variable t_i à la $i$ème équation avec une nouvelle variable de contexte $z[i,j]$.*

Nous dénotons par $\overline{t}_i$ le vecteur $(t_i^{x_1}, \ldots, t_i^{x_p})$, remarquons que $\overline{t}_i$ ne contient pas les termes constants.

Nous définissons pour un système monotone quasi-quadratique d'équations l'ensemble L_i tel que :

- $L_0 = \emptyset$
- $L_{i+1} = L_i \cup \{\overline{u}_{i+1}\}$ si $\{\overline{u}_{i+1}\} \cup \{\overline{u}_j | j \in L_i\}$ sont des vecteurs indépendants et $L_{i+1} = L_i$ sinon.

Nous supposons que les variables de contexte de $\mathcal{Z}$ sont totalement ordonnées par un ordre lexicographique sur les indices tels que $z[i,j] \prec z[i',j']$ si et seulement si $i < i'$, si $i = i'$ alors $j < j'$.

Définition 48 (Système dépendant d'équations) *Un système d'équations est* dépendant *s'il est monotone quasi-quadratique et si tous les $\overline{t}_{n+i-1}$ pour $i \leq k$ sont tels que $\{\overline{t}_{n+i-1}\} \cup \{\overline{u}_j | j \in L_i\}$ forment un ensemble de vecteurs dépendants.*

Exemple 29 *Nous considérons ce système d'équations que nous étudions au fil de ce chapitre. Nous notons* $t_1 = h(a) \oplus a$ *et* $t_2 = b \oplus h^2(a)$.

$$\begin{aligned}
&z[1,1] \odot t_1 \oplus z[1,2] \odot t_2 = h(X_1) \oplus h^2(X_2)\\
&z[2,1] \odot t_1 \oplus z[2,2] \odot t_2 \oplus z[2,3] \odot (X_1 \oplus h(X_2)) = X_1 \oplus a\\
&z[3,1] \odot t_1 \oplus z[3,2] \odot t_2 \oplus z[3,3] \odot (X_1 \oplus h(X_2)) \oplus z[3,4] \odot (h(X_1) \oplus h(a))\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad = h(X_1) \oplus h^2(X_2) \oplus X_1 \oplus a
\end{aligned}$$

Ce système d'équations est bien un système dépendant d'équations car :
- *il est monotone quasi-quadratique de par sa structure.*
- $t_3 = X_1 \oplus h(X_2)$, $\overline{u}_1 = h(X_1) \oplus h^2(X_2)$ *et* $\overline{t}_3$ *est lié à* $\{\overline{u}_1\}$.
- $t_4 = (h(X_1) \oplus h(a))$, $u_2 = X_1 \oplus a$ *et* $\overline{t}_4$ *est lié à* $\{\overline{u}_1, \overline{u}_2\}$.

Remarque 2 *D'après la notion de dépendance introduite entre les* $\overline{t_i}$ *et les* $\overline{u_i}$ *dans la définition 48 page précédente, si* σ *et* σ' *sont deux substitutions telles que* $u_i\sigma = u_i\sigma'$ *pour tout* $1 \leq i \leq N$ *avec* $i \in L$ *alors* $t_j\sigma = t_j\sigma'$ *pour tout* $1 \leq j < n + N$.

Cette remarque sera souvent utilisée implicitement dans la preuve du théorème 6.

9.3 Méthode de résolution.

L'idée principale de la résolution des systèmes dépendants d'équations (définition 48 page précédente) consiste à borner chaque variable de contexte par un terme constant calculable à priori (théorème 6). Comme nous travaillons sur $\mathbb{Z}/2\mathbb{Z}[h]$ d'après la proposition 12 page 138, il existe un nombre fini d'éléments dans un intervalle. Il ne reste plus qu'à énumérer l'ensemble des solutions pour les variables de contexte, ainsi nous obtenons un système linéaire. Nous résolvons alors ce système par les méthodes habituelles de résolution.

Nous séparons les variables de contexte $\mathcal{Z}$ en deux parties, celles qui proviennent de l'ensemble L et les autres. Nous définissons donc l'ensemble $\mathcal{Z}_L = \{z[i,j] \mid i \in L \text{ et } 1 \leq j < n + i\}$.

Théorème 6 *Soit* $\mathcal{S}(\mathcal{C})$ *un système monotone dépendant d'équations quadratiques, si* $\mathcal{S}(\mathcal{C})$ *a une solution alors il existe une solution* σ *de* $\mathcal{S}(\mathcal{C})$ *telle que* $0 \leq z\sigma < Q_{max}$ *pour tout* z *dans* $\mathcal{Z}_L$.

Preuve :

Tout d'abord remarquons que pour tout $\mathcal{Z}'$ construit ligne par ligne comme Z_L *i.e.* $\mathcal{Z}' = \{z[i,j] \mid i \in L \text{ et } 1 \leq j < n + i\}$ et tel que $\mathcal{Z}' \subseteq \mathcal{Z}$, si $z_1 \prec z_2$ et $z_2 \in \mathcal{Z}'$ alors $z_1 \in Z'$. Nous montrons que pour tout z dans $\mathcal{Z}'$ si $\mathcal{S}(\mathcal{C})$ a une solution alors il existe une solution σ de $\mathcal{S}(\mathcal{C})$ telle que $0 \leq z\sigma < Q_{max}$. Nous obtenons le résultat du théorème 6 en choisissant $\mathcal{Z}' = \mathcal{Z}_L$. Nous raisonnons par induction sur le nombre d'éléments de $\mathcal{Z}'$

Cas de base : $\mathcal{Z}' = \emptyset$, le résultat est immédiat.

Induction : Soit $z = max_{\prec}(\mathcal{Z}')$. Puisque $\mathcal{Z}' \subseteq \mathcal{Z}_L$, nous posons $z = z[N, M]$ pour N, M tels que $N \in L$ et $1 \leq M < n + N$. Par hypothèse d'induction il existe une solution σ telle que $0 \leq z'\sigma < Q_{max}$ pour tout $z' \in \mathcal{Z}'$ où $z' \neq z$.

Maintenant construisons une solution σ' de $\mathcal{S}$ telle que $0 \leq z\sigma' < Q_{max}$ pour tout $\forall z \in \mathcal{Z}'$. La construction de σ' s'effectue en quatre étapes. Dans une cinquième et dernière étape nous prouvons que σ' est aussi une solution de $\mathcal{S}$.

1. Définition de σ' pour les z' tels que $z' \prec z$

Nous définissons :

$$z'\sigma' = z\sigma \qquad \text{pour } z' \in \mathcal{Z}' \setminus \{z\}$$

2. Définition de σ' pour $z = z[N, M]$

Si z est plus grand que Q_{max} nous définissions $K, r \in \mathcal{E}$ tel que $0 \leq r < Q_{max}$ et $z\sigma = r + K \cdot Q_{max}$ et nous posons :

$$z[N, M]\sigma' = r$$

3. Définition de σ' pour $x \in vars(\mathcal{S})$

Nous cherchons à définir σ' tel que :

– pour chaque $i \in L \setminus \{N\}$, $u_i\sigma - u_i\sigma' = 0$

– $u_N\sigma - u_N\sigma' = K \cdot Q_{max} t_M\sigma$

Nous devons donc résoudre les équations suivantes où la variable X'_i correspond à $X_i\sigma - X_i\sigma'$:

$$\begin{pmatrix} u_1^{X_1} & u_1^{X_2} & \dots & u_1^{X_p} \\ \vdots & & & \\ u_N^{X_1} & u_N^{X_2} & \dots & u_N^{X_p} \\ \vdots & & & \\ u_\ell^{X_1} & u_\ell^{X_2} & \dots & u_\ell^{X_p} \end{pmatrix} \odot \begin{pmatrix} X'_1 \\ \vdots \\ X'_p \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ K \cdot Q_{max} \odot t_M\sigma \\ \vdots \\ 0 \end{pmatrix} \tag{9.2}$$

Nous pouvons résoudre le système d'équations suivant où les variables Y_i sont des éléments de $\mathbb{Z}/2\mathbb{Z}[h]$:

$$\begin{pmatrix} u_1^{X_1} & u_1^{X_2} & \dots & u_1^{X_p} \\ \vdots & & & \\ u_N^{X_1} & u_N^{X_2} & \dots & u_N^{X_p} \\ \vdots & & & \\ u_\ell^{X_1} & u_\ell^{X_2} & \dots & u_\ell^{X_p} \end{pmatrix} \cdot \begin{pmatrix} Y_1 \\ \vdots \\ Y_p \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ K \cdot Q_{max} \\ \vdots \\ 0 \end{pmatrix} \tag{9.3}$$

Grâce à la proposition 13 page 138, l'équation (9.3) a une solution $(c_1, \dots, c_p)$. Par conséquent $(c_1 \odot t_M\sigma, \dots, c_p \odot t_M\sigma)$ est une solution de (9.2). Nous pouvons donc définir σ' sur $vars(\mathcal{C})$ tel que :

$$X_i\sigma' = X_i\sigma - c_i \odot t_M\sigma \qquad i = 1, \dots, p$$

Remarquons que ces modifications n'affectent pas les variables de contexte de $\mathcal{Z}$, par conséquent nous pouvons contraindre les variables de contexte de $\mathcal{Z}$ pour arriver à borner les variables restantes.

4. Définition de $z'\sigma'$ pour $z \prec z'$

Rappelons que $z \prec z'$ si et seulement si z' est une variable contexte $z[i, q]$ où $i = N$ et $q > M$, ou $i > N$. Par conséquent, si $z \prec z[i, q]$ alors soit $q = M$ et $i > N$, soit $q \neq M$.

$$z[i,q]\sigma' = \begin{cases} z[i,q]\sigma + \sum_{j=n+N}^{n+i-1} (\sum_{l=1}^{p} t_j^{X_l} \cdot c_l) \cdot z[i,j]\sigma & \text{si } q = M, i > N \\ z[i,q] & \text{si } q \neq M \end{cases}$$

5. Vérification que σ' est une solution de $\mathcal{S}$

Remarquons d'abord :

$$t_j\sigma = t_j\sigma' \qquad \text{for } 1 \leq j < n + N \tag{9.4}$$

Ceci est une conséquence de la remarque 2 page précédente et du fait que $u_i\sigma = u_i\sigma'$ pour $1 \leq i < N$.

Premier cas $i < N$: nous concluons directement grâce à (9.4) et au fait que $u_i\sigma = u_i\sigma'$ pour $1 \leq i < N$.

Deuxième cas $i = N$: nous rappelons que

$$r = z[N,M]\sigma - K \cdot Q_{max} \tag{9.5}$$

Par conséquent :

$$\begin{aligned}
& \sum_{j=1}^{n+N-1} z[N,j]\sigma' \odot t_j\sigma' \\
= \; & \sum_{j=1}^{M-1} z[N,j]\sigma' \odot t_j\sigma' + z[N,M]\sigma' \odot t_M\sigma' + \sum_{j=M+1}^{n+N-1} z[N,j]\sigma' \odot t_j\sigma' \\
= \; & \sum_{j=1}^{M-1} z[N,j]\sigma \odot t_j\sigma + r \odot t_M\sigma + \sum_{j=M+1}^{n+N-1} z[N,j]\sigma \odot t_j\sigma \\
& \quad \text{(en utilisant (9.4 page précédente) et } z[N,j]\sigma = z[N,j]\sigma' \text{ pour } j \neq M) \\
= \; & \sum_{j=1}^{M-1} z[N,j]\sigma \odot t_j\sigma + (z[N,M]\sigma - K \cdot Q_{max}) \odot t_M\sigma + \sum_{j=M+1}^{n+N-1} z[N,j]\sigma \odot t_j\sigma \\
& \quad \text{(grâce à (9.5))} \\
= \; & \sum_{j=1}^{n+N-1} z[N,j]\sigma \odot t_j\sigma - K.Q_{max} \odot t_M\sigma \\
= \; & u_N\sigma - K.Q_{max} \odot t_M\sigma \\
& \quad \text{(puisque } \sigma \text{ est une solution de } \mathcal{S}) \\
= \; & u_N\sigma' \\
& \quad \text{(par définition de } \sigma')
\end{aligned}$$

Troisième cas $i > N$: nous considérons la $i$ème équation de $\mathcal{S}$ *i.e.* : $\sum_{1 \leq j < n+i} z[i,j] \odot t_j = u_i$ En utilisant $X_i\sigma' = X_i\sigma - c_i \odot t_M\sigma$, nous obtenons :

$$\begin{aligned}
t_j\sigma' \;&=\; \sum_{v \in Fact_{\mathsf{E}}(\mathcal{C}) \setminus vars(\mathcal{C})} (t_j^v \odot v)\sigma' + \sum_{v \in vars(\mathcal{C})} (t_j^v \odot v)\sigma' \\
&=\; \sum_{v \in Fact_{\mathsf{E}}(\mathcal{C}) \setminus vars(\mathcal{C})} (t_j^v \odot v) + \sum_{l=1}^{p} (t_j^{X_l} \odot X_l\sigma) - \sum_{l=1}^{p} (t_j^{X_l}.c_l \odot t_M\sigma)
\end{aligned} \tag{9.6}$$

Par conséquent nous obtenons :

$$
\begin{array}{rl}
& \displaystyle\sum_{j=1}^{n+i-1} z[i,j]\sigma' \odot t_j\sigma' \\
= & \displaystyle\sum_{j=1}^{M-1} z[i,j]\sigma' \odot t_j\sigma' + z[i,M]\sigma' \odot t_M\sigma' + \sum_{j=M+1}^{n+N-1} z[i,j]\sigma' \odot t_j\sigma' + \sum_{j=n+N}^{n+i-1} z[i,j]\sigma' \odot t_j\sigma' \\
= & \displaystyle\sum_{j=1}^{M-1} z[i,j]\sigma \odot t_j\sigma + z[i,M]\sigma' \odot t_M\sigma' + \sum_{j=M+1}^{n+N-1} z[i,j]\sigma \odot t_j\sigma + \sum_{j=n+N}^{n+i-1} z[i,j]\sigma \odot t_j\sigma' \\
& \qquad \text{(par (9.4 page 141) et } z[N,j]\sigma = z[N,j]\sigma' \text{ pour } j \neq M) \\
= & \displaystyle\sum_{j=1}^{M-1} z[i,j]\sigma \odot t_j\sigma + (z[i,M]\sigma + \sum_{j=n+N}^{n+i-1} (\sum_{l=1}^{p} t_j^{X_l} \cdot c_l) \cdot z[i,j]\sigma) \odot t_M\sigma \\
& \displaystyle + \sum_{j=M+1}^{n+N-1} z[i,j]\sigma \odot t_j\sigma \\
& \displaystyle + \sum_{j=n+N}^{n+i-1} z[i,j]\sigma \cdot (\sum_{v \in Fact_{\mathsf{E}}(\mathcal{C}) \setminus vars(\mathcal{C})} (t_j^v \odot v) \\
& \displaystyle \qquad\qquad + \sum_{l=1}^{p} (t_j^{X_l} \odot X_l\sigma) - \sum_{l=1}^{p} (t_j^{X_l} \cdot c_l \odot t_M\sigma)) \\
& \qquad \text{(par définition de } \sigma' \text{ et (9.6 page ci-contre))} \\
= & \displaystyle\sum_{j=1}^{n+i-1} z[i,j]\sigma \odot t_j\sigma \\
= & u_i\sigma \\
& \qquad \text{(puisque } \sigma \text{ est une solution de } \mathcal{S}) \\
= & u_i\sigma' \\
& \qquad \text{(comme } u_i\sigma = u_i\sigma' \text{ pour } i > N)
\end{array}
$$

□

Chapitre **10**

Procédure de décision.

« C'est avec la logique que nous prouvons et avec l'intuition que nous trouvons. »

Henri Poincaré.

Sommaire

Se servant de tous les résultats obtenus dans les chapitres précédents, nous montrons maintenant dans le théorème 8 page 164 que le problème d'insécurité avec la théorie équationnelle du « ou exclusif » avec un symbole d'homomorphisme est décidable. Nous considérons alors les protocoles représentés par des systèmes de contraintes bien définis (définition 31 page 121).

Nous définissons d'abord les notions importantes de termes, sous-termes, preuves et solutions. Ensuite nous rappelons le résultat de localité pour un intrus passif avec la théorie équationnelle du « ou exclusif » et de l'homomorphisme. Nous prouvons alors qu'il existe toujours une solution qui conserve la « structure » des termes présents dans la contrainte, nous appelons une telle solution une solution conservatrice (définition 61 page 149). Nous montrons ensuite qu'il existe toujours une solution conservatrice, cela nous permet d'étendre le résultat de localité obtenu dans le cas clos (cas passif) au « cas actif ». Nous pouvons alors transformer par le lemme 40 page 153, un système de contraintes bien défini vers un système de contraintes une étape bien défini. À partir d'un tel système nous construisons alors un système $\mathsf{M_E}$ de contraintes (lemme 41 page 155). Nous détaillons comment résoudre un système $\mathsf{M_E}$ en le transformant en un système d'équations dépendantes. Nous savons résoudre grâce au chapitre 9 page 137 ce type d'équations. Tous ces résultats peuvent aussi être obtenus pour les théories équationnelles $\mathbb{Z}$, $\mathbb{Z}/p\mathbb{Z}$, $\mathbb{Z}/p\mathbb{Z}[h]$ et $\mathbb{Z}/2\mathbb{Z}[h]$. Dans ce chapitre, nous présentons en détails la procédure de décision pour $\mathbb{Z}/2\mathbb{Z}[h]$, le cas le plus complexe à notre avis.

Nous donnons d'abord dans la figure 10.1 le système de Dolev-Yao avec lequel nous travaillons dans ce chapitre, nous regroupons les applications des règles (GX) et (h) dans une seule « macro » règle ($\mathsf{M_E}$), où $\mathcal{F}$ est l'ensemble des symboles de fonctions que nous considérons.

$$(\mathsf{UL})\ \frac{T \vdash \langle u, v\rangle}{T \vdash u} \qquad \text{Composition } (\mathsf{C})\ \frac{T \vdash u_1\ \ldots\ T \vdash u_n}{T \vdash f(u_1, \ldots, u_n)} \text{ où } f \in \mathcal{F} \smallsetminus \{+, h, 0\}$$

$$(\mathsf{UR})\ \frac{T \vdash \langle u, v\rangle}{T \vdash v} \qquad \text{Contexte } (\mathsf{M_E})\ \frac{T \vdash u_1\ \ldots\ T \vdash u_n}{T \vdash C[u_1, \ldots, u_n]\downarrow} \text{ avec } C \text{ un } \mathsf{E}\text{-context}$$

$$\text{Déchiffrement } (\mathsf{D})\ \frac{T \vdash \{u\}_v \quad T \vdash v}{T \vdash u}$$

FIG. 10.1 – Dolev-Yao étendu pour la théorie équationnelle ACUNh, où C est l'application d'un polynôme de $\mathbb{Z}/2\mathbb{Z}[h]$.

Ce modèle correspond à celui de la figure 5.2 page 75, où la fonction de chiffrement est vue comme une règle de composition, car elle appartient aux symboles de fonction de $\mathcal{F} \smallsetminus \{+, h, 0\}$.

10.1 Définitions.

Nous rappelons d'abord quelques définitions mathématiques usuelles.

Définition 49 (Ordre bien fondé) *Soit E un ensemble non vide, une relation R sur E est* bien fondée ou *noethérien si et seulement s'il vérifie l'une des deux conditions suivantes (équivalentes d'après l'axiome du choix) :*

- *Il n'existe pas de suite infinie (x_n) d'éléments de E telle que $x_{n+1} R x_n$ pour tout n.*
- *Pour toute partie X de E, il existe un élément x de X n'ayant aucun élément y de X tel que yRx.*

Définition 50 (Ordre total) *Un ordre total est un ordre dans lequel tous les éléments sont deux à deux comparables.*

Définition 51 (Domaine et substitution) *Une substitution σ est une application d'un sous-ensemble fini de $\mathcal{X}$ vers $\mathcal{T}(\mathcal{F}, \mathcal{X})$. Ce sous-ensemble fini est appelé domaine de σ et noté $dom(\sigma)$.*

Définition 52 (Image) *L'image d'une substitution σ est l'ensemble $img(\sigma) = \{x\sigma | x \in dom(\sigma)\}$.*

10.1.1 Termes, sous-termes, facteurs.

Nous introduisons les notions sur les termes, sous-termes et facteurs utilisées par la suite.

Définition 53 (Terme standard) *Un terme t est* standard *si et seulement si t n'est pas de la forme $f(t_1, \ldots, t_n)$ où $t_1, \ldots, t_n$ sont des termes et f est un symbole de fonction de la signature $\{0, h, +\}$. En particulier toute variable est un terme standard.*

Exemple 30 *Le terme $x \oplus h(\{y\}_k) \oplus \langle x, b\rangle$ n'est pas un terme standard alors que les termes $\langle h(x) \oplus a, b\rangle$ et $\{x \oplus a\}_k$ sont des termes standards.*

Définition 54 (Terme décomposable) *Soit P une preuve de $T \vdash u$. Nous disons qu'un terme standard v est décomposable dans P si :*
- *soit $v = \langle u_1, u_2\rangle$ et P contient une instance de* (UL) *ou* (UR) *dont l'hypothèse est étiquetée par $T \vdash \langle u_1, u_2\rangle$.*
- *soit $v = \{u_1\}_{u_2}$ et P contient une instance de* (D) *dont les hypothèses sont étiquetées par $T \vdash \{u_1\}_{u_2}$ et $T \vdash u_2$.*

Définition 55 (Facteurs) *Soit t un terme en forme normale, nous définissons pour $t = C[t_1, \ldots, t_n]$ où $t_1, \ldots, t_n$ sont des termes et C est un* E*-contexte, l'ensemble $Fact_{\mathsf{E}}(t)$ des* facteurs *de t par $Fact_{\mathsf{E}}(t) = \{t_1, \ldots, t_n\}$.*

Exemple 31 *Considérons les trois termes suivants t_1, t_2 et t_3. Calculons leurs ensembles facteurs.*

$$\begin{array}{ll} t_1 = h^2(a) \oplus b \oplus x & Fact_{\mathsf{E}}(t_1) = \{a, b, x\} \\ t_2 = h(\langle a, b\rangle) \oplus x & Fact_{\mathsf{E}}(t_2) = \{\langle a, b\rangle, x\} \\ t_3 = \langle a \oplus b \oplus x, d\rangle & Fact_{\mathsf{E}}(t_3) = \{t_3\} \end{array}$$

Définition 56 (Sous-termes) *L'ensemble $St_{\mathsf{E}}(t)$ de* sous-termes *de t est le plus petit ensemble de termes tel que :*
- *$t \in St_{\mathsf{E}}(t)$,*
- *Si $f(t_1, \ldots, t_n) \in St_{\mathsf{E}}(t)$ et $f(t_1, \ldots, t_n)$ est un terme standard alors $t_1, \ldots t_n \in St_{\mathsf{E}}(t)$,*
- *Si $s \in St_{\mathsf{E}}(t)$ et s n'est pas un terme standard alors $Fact_{\mathsf{E}}(s) \subseteq St_{\mathsf{E}}(t)$.*

Nous étendons ces notions de manière naturelle aux ensembles de termes *i.e.* soit T un ensemble de termes nous posons :

$$St_{\mathsf{E}}(T) = \bigcup_{t \in T} St_{\mathsf{E}}(t)$$

Exemple 32 *Reprenons les termes de l'exemple 31 et calculons les ensembles de sous-termes correspondants :*

$$\begin{array}{lll} t_1 = h^2(a) \oplus b \oplus x & Fact_{\mathsf{E}}(t_1) = \{a, b, x\} & St_{\mathsf{E}}(t_1) = \{t_1, a, b, x\} \\ t_2 = h(\langle a, b\rangle) \oplus x & Fact_{\mathsf{E}}(t_2) = \{\langle a, b\rangle, x\} & St_{\mathsf{E}}(t_2) = \{t_2, \langle a, b\rangle, a, b, x\} \\ t_3 = \langle a \oplus b \oplus x, d\rangle & Fact_{\mathsf{E}}(t_3) = \{t_3\} & St_{\mathsf{E}}(t_3) = \{t_3, a \oplus b \oplus x, d, a, b, x\} \end{array}$$

Remarquons que la constante 0 est une conséquence de la règle ($\mathsf{M_E}$). De plus 0 n'est pas un terme standard par définition et les facteurs de tout terme sont nécessairement des termes standards.

Proposition 14 *Soit t un terme et σ une substitution, nous avons :*

$$St_{\mathsf{E}}(t\sigma) \subseteq St_{\mathsf{E}}(t)\sigma \cup \bigcup_{x \in vars(t)} St_{\mathsf{E}}(x\sigma)$$

Preuve : Nous prouvons cette proposition par induction structurelle sur t.
- Si t est une constante ou une variable, alors $St_{\mathsf{E}}(t\sigma) = St_{\mathsf{E}}(t)\sigma$.
- Si t est un terme standard, *i.e.* $t = f(t_1, \ldots, t_n)$ avec $f \in \mathcal{F} \setminus sig(\mathsf{E})$, nous avons :

$$\begin{array}{rcl} St_{\mathsf{E}}(t\sigma) & = & \{t\sigma\} \cup \bigcup_{i=1}^{n} St_{\mathsf{E}}(t_i\sigma) \text{ par définition} \\ & \subseteq & \{t\sigma\} \cup \bigcup_{i=1}^{n} (St_{\mathsf{E}}(t_i)\sigma \cup \bigcup_{x \in vars(t_i)} St_{\mathsf{E}}(x\sigma)) \text{ par hypothèse d'induction} \\ & \subseteq & St_{\mathsf{E}}(f(t_1, \ldots, t_n))\sigma \cup \bigcup_{x \in vars(\{t_1, \ldots, t_n\}} St_{\mathsf{E}}(x\sigma)) \\ & \subseteq & St_{\mathsf{E}}(t)\sigma \cup \bigcup_{x \in vars(t)} St_{\mathsf{E}}(x\sigma) \end{array}$$

- Si t n'est pas un terme standard, alors nous avons $t = C[t_1, \ldots, t_n]$ où $t_1, \ldots, t_n$ sont des termes standards et C est un E-contexte. Nous avons donc $St_{\mathsf{E}}(t\sigma) = \{t\sigma\} \cup St_{\mathsf{E}}(Fact_{\mathsf{E}}(t\sigma))$ par définition, or $Fact_{\mathsf{E}}(t\sigma) \subseteq \bigcup_{i=1}^{n} t_i\sigma$. Nous pouvons alors effectuer le même raisonnement que précédemment.

□

Nous pouvons évidemment étendre la proposition 14 page précédente aux ensembles de sous-termes. Remarquons, toutefois, que l'inclusion doit être stricte comme le montre l'exemple 33.

Exemple 33 *Soit* $t = x \oplus y$ *et* $\sigma = \{x \mapsto a; y \mapsto a\}$. *Nous avons* $St_{\mathsf{E}}(t\sigma) = \{0\}$ *tandis que* $St_{\mathsf{E}}(t)\sigma \cup St_{\mathsf{E}}(\{x\sigma, y\sigma\}) = \{0, a\}$.

10.1.2 Preuves.

Nous introduisons la notion de terme déductible en une étape dans une preuve par une instance d'une règle précise, dans le but de distinguer les règles de « composition » et les règles de « décomposition ».

Définition 57 (R-déductible en une étape) *Un terme u est* R-déductible en une étape *à partir d'un ensemble de termes T si :*
- *$T \vdash u$ est une preuve de $T \vdash u$ (i.e. $u \in T$ ou $u = 0$),*
- *il existe $u_1, \ldots, u_n$ tels que* (R) *est une preuve de $T \vdash u$ de la forme suivante :*

$$\frac{T \vdash u_1 \quad \ldots \quad T \vdash u_n}{T \vdash u}\ (\mathsf{R})$$

Le terme u est déductible en une étape *à partir de T si u est* R-*déductible à partir de T grâce à une seule règle R du système d'inférence.*

Lorsqu'il n'y a pas d'ambiguïté, nous utiliserons déductible en une étape au lieu de R-déductible en une étape.

Définition 58 (DY-*déductible en une étape*) *Un terme u est* DY-déductible en une étape *si et seulement si la règle d'inférence utilisée est une des règles du modèle de Dolev-Yao usuel i.e. une des règles parmi* $\{\mathsf{C}, \mathsf{UL}, \mathsf{UR}, \mathsf{D}\}$.

Nous notons $T \vdash_{\mathsf{DY}} u$ si u est déductible en une (R) étape à partir de T où R est une des règles suivantes (D, UL, UR, C). Il est facile de décider si u est DY-déductible à partir de T ou non.

La proposition 15 a été prouvée par M. Rusinovitch *et alii* [RT03] pour le modèle de Dolev-Yao standard. Nous redémontrons ce résultat pour notre modèle d'intrus avec la nouvelle règle (M_{E}). Cette proposition sera utilisée dans la preuve du lemme 38 page 150.

Proposition 15 *Soit P une preuve de $T \vdash u$ et P' une preuve minimale de $T \vdash \gamma$. Si P' termine avec une instance de la règle* (C) *de composition, alors il existe une preuve de $T \vdash u$ dans laquelle le terme γ n'est jamais décomposable.*

Preuve : Nous effectuons la preuve par induction sur le nombre de règles d'inférence de la preuve.

Cas de base : S'il n'y a pas d'instance de règle alors le terme γ n'est jamais décomposable.

Induction : Hypothèse d'induction : si nous avons une preuve P de $T \vdash u$ contenant n règles d'inférence et P' une preuve minimale de $T \vdash \gamma$ et si P' termine avec une instance de la règle *(C)* de composition, alors il existe une preuve de $T \vdash u$ dans laquelle le terme γ n'est jamais décomposable. Supposons qu'il existe $n+1$ instances de règle dans la preuve P de $T \vdash u$ et que P décompose γ. Nous nous ramenons à une preuve de $T \vdash u$ avec moins de $n+1$ instances de règle d'inférence et appliquer l'hypothèse d'induction. Nous distinguons deux cas : si γ est une paire (*i.e.* $\langle \gamma_1, \gamma_2 \rangle$) ou si γ est un message chiffré (*i.e.* $\{\gamma_1\}_{\gamma_2}$). Dans le premier cas, cela signifie qu'il existe une instance de (UL) (ou (UR)) dont l'hypothèse est $\langle \gamma_1, \gamma_2 \rangle$ et la conclusion est γ_1 (ou γ_2). À partir de P', nous extrayons facilement une preuve P_1 de $T \vdash \gamma_1$ (resp. P_2 de $T \vdash \gamma_2$). Notons que P_1 (resp. P_2) ne décompose pas γ par minimalité de P'. Par conséquent, une telle preuve peut être mise à la place de la sous-preuve de $T \vdash \gamma_1$ (resp. $T \vdash \gamma_2$) dans P, cela diminue le nombre de règles d'inférence de la preuve de $T \vdash u$ et nous pouvons donc appliquer l'hypothèse d'induction. La démonstration pour le second cas, où $\gamma = \{\gamma_1\}_{\gamma_2}$, est similaire au premier cas.

□

Définition 59 (Remplacement) *Soit deux termes u et v, le* remplacement *de u par v, dénoté $[u \mapsto v]$, fait correspondre à tout terme t le terme $t[u \mapsto v]$*

obtenu en remplaçant toutes les occurrences de u dans t par v (« top-down »).

Exemple 34 *Soit $t = \langle u, \{a\}_u \rangle$ un terme et le remplacement $[u \mapsto v]$. Alors $t[u \mapsto v] = \langle v, \{a\}_v \rangle$*

Remarquons que le résultat d'un remplacement est uniquement déterminé.

Définition 60 (Preuve de décomposition) *Une preuve P de $T \vdash u$ est une* preuve de décomposition *si soit :*

- *P est réduit à une feuille.*
- *P termine par une instance d'une règle de décomposition (i.e. (UL, UR, D)),*
- *P termine par une instance de la règle ($\mathsf{M_E}$) et u est un terme standard.*

Exemple 35 *La preuve suivante est une preuve de décomposition car elle se termine par un terme standard c.*

$$\frac{T \vdash a \oplus h(b) \oplus c \quad T \vdash h^2(b) \oplus h(a)}{T \vdash c}\ (\mathsf{M_E})$$

10.1.3 Solution conservatrice.

Nous introduisons la notion de *solution conservatrice* d'un système de contraintes. Intuitivement une solution conservatrice est une solution dont les termes ne contiennent pas de « nouvelle structure » par rapport au système de contraintes *i.e.* par exemple s'il n'y a pas de symbole de paire dans un système de contraintes la solution conservatrice ne contiendra pas de symbole de paire.

Définition 61 (Solution conservatrice) *Soit $\mathcal{C}$ un système de contraintes et σ une solution de $\mathcal{C}$ (une substitution), σ est une* solution conservatrice *de $\mathcal{C}$ si et seulement si pour tout $x \in vars(\mathcal{C})$, $Fact_{\mathsf{E}}(x\sigma) \subseteq (St_{\mathsf{E}}(\mathcal{C}) \setminus vars(\mathcal{C}))\sigma$.*

Exemple 36 *Considérons le système $\mathcal{C}$ de contraintes bien défini (définition 31 page 121) suivant :*

$$\left\{\begin{array}{lcl} a, h(b) & \Vdash & h(x) \\ a, h(b), x & \Vdash & \langle a, b\rangle \end{array}\right.$$

Une solution de $\mathcal{C}$ est : $\sigma = \{x \mapsto \langle a, a\rangle \oplus b\}$. Cette solution n'est pas une solution conservatrice de $\mathcal{C}$, car $Fact_{\mathsf{E}}(\langle a, a\rangle \oplus b) = \{\langle a, a\rangle, b\}$, et $\langle a, a\rangle$ n'appartient pas à $(St_{\mathsf{E}}(\mathcal{C}) \setminus \{x\})\sigma = \{0, h(b), b, h(\langle a, a\rangle \oplus b), \langle a, b\rangle, a\}$. Par contre $\sigma' = \{x \mapsto b\}$ est une solution conservatrice de $\mathcal{C}$.

Rappelons que nous considérons implicitement que les termes sont en forme normale par conséquent nous écrirons $u\sigma$ au lieu de $u\sigma\downarrow$.

10.2 Rappel du résultat de localité dans le cas passif.

Le lemme 37 montré par S. Delaune [Del06a] améliore le résultat de complexité pour l'intrus passif en présence de la théorie équationnelle ACUNh [LLT05], la procédure proposée est P-TIME. Ce résultat, rappelé dans le lemme 37, est le premier pas essentiel pour décider l'intrus actif, il est utilisé dans la suite du chapitre et en particulier dans la preuve du lemme 39 page 152.

Lemme 37 *Une preuve minimale P de $T \vdash u$ contient uniquement des termes de $St_{\mathsf{E}}(T \cup \{u\})$. De plus si P est une preuve de décomposition alors P contient uniquement des termes de $St_{\mathsf{E}}(T)$.*

Preuve : Nous effectuons la preuve par induction sur le nombre d'instances de règle de P qui décomposent γ.

- Cas de base : S'il n'y a pas de telle instance, alors P est la preuve attendue.
- Supposons qu'il y ait $n+1$ instances de règles de P qui décomposent γ. Nous distinguons deux cas cases, suivant si γ est une paire (*i.e.* $\langle \gamma_1, \gamma_2\rangle$) ou un chiffré (*i.e.* $\{\gamma_1\}_{\gamma_2}$). Dans le premier cas, cela signifie qu'il existe une instance de (UL) (ou (UR)) dont l'hypothèse est $\langle \gamma_1, \gamma_2\rangle$ et la conclusion est γ_1 (ou γ_2). À partir de P', nous extrayons une preuve P_1 de $T \vdash \gamma_1$ (resp. P_2 de $T \vdash \gamma_2$). Notons que P_1 (resp. P_2) ne décompose pas γ par minimalité de P'. En conséquence, nous remplaçons cette preuve par la sous-preuve de $T \vdash \gamma_1$ (resp. $T \vdash \gamma_2$) dans P qui décompose γ. Le second cas où $\gamma = \{\gamma_1\}_{\gamma_2}$ est semblable. Nous obtenons une preuve de $T \vdash u$ qui contient moins d'instances de règles qui décomposent γ que P. Par conséquent, nous appliquons l'hypothèse d'induction sur cette nouvelle preuve pour conclure.

$\square$

10.3 Existence d'une solution conservatrice.

Le lemme 38 montre que dans un système de contraintes bien défini s'il existe une solution alors il existe une solution conservatrice. La preuve utilise le lemme de localité 37.

Lemme 38 *Soit $\mathcal{C}$ un système de contraintes bien défini, si $\mathcal{C}$ a une solution σ alors il existe une solution conservatrice de $\mathcal{C}$.*

Preuve : Nous considérons un ordre bien fondé (définition 49 page 146), noté $\prec$, sur les termes standards de $\mathcal{T}(\mathcal{F}, \mathcal{X})$, tel que la constante 0 soit minimale. Nous utiliserons l'extension $\ll$ de l'ordre $\prec$ aux multi-ensembles de termes clos standards. Cette extension $\ll$ est définie de la manière suivante. Soit σ_1 et σ_2 deux solutions d'un système de contraintes, nous dénotons $\sigma_1 \ll \sigma_2$ si et seulement si $Fact_{\mathsf{E}}(img(\sigma_1)) \ll Fact_{\mathsf{E}}(img(\sigma_2))$.

Soit σ une solution minimale, selon l'ordre $\ll$, du système de contraintes $\mathcal{C} = \{C_1, \dots, C_k\}$ tel que pour chaque $i \leq k, C_i = T_i \Vdash u_i$. Nous notons $C_i\sigma$ la contrainte obtenue à partir de C_i en instanciant et normalisant tous les termes par σ. Nous montrons par contradiction que la substitution σ est une solution conservatrice de $\mathcal{C}$. Supposons que la substitution σ ne soit pas une solution conservatrice du système de contraintes $\mathcal{C}$. Il existe donc $x \in vars(\mathcal{C})$ et $v_x \in Fact_{\mathsf{E}}(x\sigma)$ tel que $v_x \notin (St_{\mathsf{E}}(\mathcal{C}) \setminus vars(\mathcal{C}))\sigma$ *i.e.* pour tout $t \in \mathcal{T}(\mathcal{F}, \mathcal{X}) \setminus \mathcal{X}$ tel que $t\sigma =_E v_x$, nous avons $t \notin St_{\mathsf{E}}(\mathcal{C})$. Nous contredisons la minimalité de σ en montrant que sous cette condition il existe une solution σ' plus petite de $\mathcal{C}$. Nous montrons d'abord le fait 9.

Fait 9 *Si $v_x \in St_{\mathsf{E}}(s\sigma)$ pour $i \leq k$ et s tel que $s \in T_i$, alors il existe $j < i$ tel que $v_x \in St_{\mathsf{E}}(u_j\sigma)$.*

Preuve : Nous raisonnons par contradiction. Supposons que $v_x \in St_{\mathsf{E}}(s\sigma)$ pour $s \in T_i$ $(i \leq k)$, et $v_x \notin St_{\mathsf{E}}(u_j\sigma)$ pour tout $j < i$. Soit z une nouvelle variable, ρ un remplacement $\{v_x \mapsto z\}$ et $\theta := \sigma\rho$, nous montrons que $\mathcal{C}\theta$ ne satisfait pas la propriété d'origination ce qui contredit le fait que $\mathcal{C}$ est bien défini. Premièrement, puisque $v_x \notin (St_{\mathsf{E}}(\mathcal{C}) \setminus vars(\mathcal{C}))\sigma$, nous pouvons faire le remplacement ρ en même temps que la substitution θ, car ils ne partagent pas de variables, et donc $(\mathcal{C}\sigma)\rho = \mathcal{C}(\sigma\rho)$, ce qui est égal à $\mathcal{C}\theta$ par définition de θ. Par hypothèse $v_x \in St_{\mathsf{E}}(T_i\sigma)$, donc $z \in vars(T_i\theta)$. De plus pour tout $j < i$, nous avons $z \notin vars(u_j\theta)$, car $v_x \notin St_{\mathsf{E}}(u_j\sigma)$. $\square$

Ceci nous permet de définir l'indice m tel que : $m = \min\{j \mid v_x \in St_{\mathsf{E}}(u_j\sigma)\}$. Prouvons maintenant le fait 10.

Fait 10 *Il existe P' une preuve de $T_m\sigma \vdash v_x$ finissant par une instance de la règle de composition* (C).

Preuve : Par hypothèse il existe une preuve minimale P de $T_m\sigma \vdash u_m\sigma$. Premièrement, nous montrons qu'il existe dans la preuve P un nœud étiqueté par $T_m\sigma \vdash v_x$. Car si la preuve P ne contient pas de nœud étiqueté par $T_m\sigma \vdash v_x$, nous pouvons trouver récursivement une branche dans la preuve P, de la racine vers une feuille, où un nœud est étiqueté par $T_m\sigma \vdash u$ tel que $v_x \in St_{\mathsf{E}}(u)$. Grâce au fait 9, l'existence d'une telle branche contredit la minimalité de m. Donc $T_m\sigma \vdash v_x$ est bien un nœud de P. Ainsi par définition de m et grâce au lemme 37 page ci-contre, la sous-preuve P' de P étiqueté par $T_m\sigma \vdash v_x$ n'est pas une preuve de décomposition (sinon $v_x \in St_{\mathsf{E}}(T_m\sigma)$). Puisque v_x est nécessairement un terme standard (car c'est un facteur), cela implique que P' termine par une instance de la règle (C). $\square$

Maintenant, nous considérons δ un remplacement $\{v_x \mapsto 0\}$. Nous montrons que $\sigma' := \sigma\delta$ est aussi une solution de $\mathcal{C}$, cela va contredire le fait que $\sigma' \ll \sigma$. Or v_x est un facteur donc un terme standard, par conséquent $0 \prec v_x$. Nous construisons une preuve pour chaque $C_i\sigma'$, $i \leq l$. Nous distinguons deux cas pour l'indice de la contrainte considérée :

1. $i < m$: Par définition de m, $v_x \notin St_{\mathsf{E}}(C_i\sigma)$. Dans ce cas, $(C_i\sigma)\delta = C_i\sigma = C_i\sigma'$, *i.e.* σ' est une solution de C_i.
2. $i \geq m$: Dans le reste de la démonstration, nous montrons que $\sigma' = \sigma\delta$ est aussi une solution de $C_i = T_i \Vdash u_i$. Premièrement, nous notons que $C_i(\sigma\delta) = (C_i\sigma)\delta$ car par hypothèse $v_x \notin (St_{\mathsf{E}}(\mathcal{C}) \setminus vars(C))\sigma$. De plus par hypothèse σ est une solution de C_i, nous avons donc une preuve P de $T_i\sigma \vdash u_i\sigma$. Le fait 10 assure l'existence d'une preuve de $T_i\sigma \vdash v_x$ qui termine par une instance d'une règle de composition (C) dans la preuve P. La substitution σ' est une solution de $\mathcal{C}_i$, c'est clair pour $i = m$ et nous étendons le résultat pour $i > m$ car $\mathcal{C}$ est bien défini (nous utilisons la stabilité par substitution de la « bonne définition » de $\mathcal{C}$). Nous pouvons alors appliquer la proposition 15 page 148 pour obtenir une preuve P_i de $T_i\sigma \vdash u_i\sigma$

dans laquelle v_x n'est jamais décomposable. Nous construisons alors à partir de P_i une preuve P'_i de $(T_i\sigma)\delta \vdash (u_i\sigma)\delta$ en remplaçant tous sous-arbres qui se terminent avec :

$$\frac{T_i\sigma \vdash v_1 \ldots T_i\sigma \vdash v_n}{T_i\sigma \vdash v_x}\,(\mathsf{C})$$

par une feuille étiquetée par $T_i\sigma \vdash v_x$ et en appliquant δ à tous les termes de l'arbre obtenu. Nous montrons alors grâce au fait 11 que σ' est une solution de $\mathcal{C}_i$ ce qui termine la démonstration.

Fait 11 *P'_i est une preuve de $(T_i\sigma)\delta \vdash (u_i\sigma)\delta$.*

Nous devons montrer que pour tous nœuds de P'_i étiquetés par $T_i\sigma\delta \vdash v_0$ et possédant n fils étiquetés respectivement par $T_i\sigma\delta \vdash v_1, \ldots T_i\sigma\delta \vdash v_n$, alors l'inférence :

$$\frac{T_i\sigma\delta \vdash v_1 \ldots T_i\sigma\delta \vdash v_n}{T_i\sigma\delta \vdash v_0}$$

est une instance d'une règle des règles d'inférences considérées dans la figure 5.2 page 75.

Nous considérons plusieurs cas :

- Si l'inférence est une feuille ajoutée par le remplacement d'une instance de (C) dans la construction de P'_i donnée ci-dessus, nous avons alors $v_0 = 0$, par conséquent $v_0 \in T_i\sigma\delta$.
- Si l'inférence n'est pas une feuille ajoutée par le remplacement, nous avons alors une inférence « correspondante » dans P_i. Cela signifie qu'il existe :

  $$\frac{T_i\sigma \vdash u_1 \ldots T_i\sigma \vdash u_n}{T_i\sigma \vdash u_0}$$

  une étape d'inférence dans P_i telle que $v_i = u_i\delta$ pour chaque $0 \leq i \leq n$. Or par construction de P'_i, v_x n'est jamais décomposable dans P_i et la conclusion de l'instance de la règle (C) ne peut pas être v_x, alors nous montrons par une analyse de cas sur la règle d'inférence que l'application de δ sur l'inférence ci-dessus, nous donne une autre instance de la même règle d'inférence.

□

Exemple 37 *Reprenons l'exemple 36 page 150, le lemme 38 page 150 assure donc l'existence de la solution conservatrice σ'.*

10.4 Lemme de localité étendu au cas actif.

La notion de solution conservatrice et le résultat sur la localité (lemme 37 page 150) nous permettent de démontrer un lemme de localité sur les variables (lemme 39).

Lemme 39 *Soit σ une solution conservatrice d'un système de contraintes $\mathcal{C} = \{C_1, \ldots, C_k\}$. Pour chaque $i \leq k$, il existe une preuve $C_i\sigma$ qui ne contient que des termes dans $St_{\mathsf{E}}(\mathcal{C})\sigma$.*

Preuve : Grâce au lemme 37 page 150 (lemme de localité), pour chaque i il existe P_i une preuve minimale de $T_i\sigma \vdash u_i\sigma$ qui ne contient que des termes de $St_{\mathsf{E}}(\mathcal{C}\sigma)$. Grâce à la proposition 14 page 147, nous obtenons $St_{\mathsf{E}}(\mathcal{C}\sigma) \subseteq St_{\mathsf{E}}(\mathcal{C})\sigma \cup \bigcup_{x\in vars(\mathcal{C})} St_{\mathsf{E}}(x\sigma)$. Par conséquent :

$$\begin{array}{rcll}
St_{\mathsf{E}}(\mathcal{C}\sigma) & \subseteq & St_{\mathsf{E}}(\mathcal{C})\sigma \cup \bigcup_{x\in vars(\mathcal{C})} St_{\mathsf{E}}(x\sigma) & \\
& \subseteq & (St_{\mathsf{E}}(\mathcal{C}) \setminus vars(\mathcal{C}))\sigma \cup \bigcup_{x\in vars(\mathcal{C})} St_{\mathsf{E}}(x\sigma) & \\
& \subseteq & \bar{S}(\mathcal{C})\sigma & \text{puisque } \sigma \text{ est une solution conservatrice de } \mathcal{C}
\end{array}$$

où $\bar{S}(\mathcal{C}) = \{C[t_1, \ldots, t_n] \mid \forall i.\ t_i \in St_{\mathsf{E}}(\mathcal{C}) \setminus vars(\mathcal{C})$ et C est un E-contexte$\}$

Considérons (R) une règle d'inférence de P_i : $\dfrac{T_i\sigma \vdash u_1 \ldots T_i\sigma \vdash u_n}{T_i\sigma \vdash u_0}$ (R)

Il y a alors deux possibilités :

- (R) est une instance d'une règle autre que ($\mathsf{M_E}$), alors pour tout $j \in \{0, \ldots n\}$, $u_j \in St_{\mathsf{E}}(\mathcal{C})\sigma$.
- (R) est une instance de la règle ($\mathsf{M_E}$). Par minimalité de P_i, une instance de règle ($\mathsf{M_E}$) ne peut pas être suivie par une autre instance de ($\mathsf{M_E}$) (sinon nous les regroupons en une seule instance). Par conséquent, pour chaque hypothèse $T_i\sigma \vdash u$ d'une instance de ($\mathsf{M_E}$),
  - soit $T_i\sigma \vdash u$ est une conclusion d'une instance d'une autre règle d'inférence que ($\mathsf{M_E}$),
  - soit $u \in T_i\sigma$.

  La conclusion $T_i\sigma \vdash u$ d'une instance de ($\mathsf{M_E}$) est :
  - soit une hypothèse d'une instance d'une autre règle que ($\mathsf{M_E}$),
  - soit $u = u_i\sigma$.

Par conséquent nous concluons qu'il existe une preuve P_i de $T_i\sigma \vdash u_i\sigma$ qui ne contient que des termes dans $St_{\mathsf{E}}(\mathcal{C})\sigma$. $\square$

10.5 D'un système de contraintes bien défini vers un système de contraintes une étape bien défini.

Nous considérons un système de contraintes bien défini. L'idée de cette première étape est de décomposer chaque contrainte en un ensemble de contraintes une étape. L'algorithme 2 utilisé dans cette étape est complet pour les solutions conservatrices. Grâce au lemme 38 page 150, l'algorithme est correct car il existe toujours une solution conservatrice.

Algorithme 2 : Algorithme de transformation d'un système de contraintes vers un système de contraintes une étape.

```
début
  Entrées : C = {T_1 ⊩ u_1, ..., T_k ⊩ u_k}
  Deviner S tel que S ⊆ St_E(C) ;
  pour s ∈ S faire  deviner j(s) tel que j(s) ∈ {1, ..., k} ;
  C' = ∅ ;
  pour i = 1 jusqu'à k faire
    S_i := {s | j(s) = i} ;
    Choisir un ordre total sur S_i = {s_i^1, ..., s_i^{k_i}} ;
    pour j = 1 jusqu'à k_i faire
      T := T_i ∪ S_1 ∪ ... ∪ S_{i-1} ∪ {s_i^1, ..., s_i^{j-1}} ;
      C' := C' ∪ {T ⊩_1 s_i^j} ;
    C' := C' ∪ {T ⊩_1 u_i} ;
  retourner C'
fin
```

Lemme 40 *Soit $\mathcal{C}$ un système de contraintes bien défini et $\mathscr{C}'$ le système de contraintes obtenu par l'application de l'algorithme 2 sur $\mathcal{C}$:*

1. *$\mathscr{C}'$ est une système fini de contraintes une étape bien défini.*
2. *Si $\mathscr{C}'$ a une solution alors $\mathcal{C}$ a une solution.*
3. *Si $\mathcal{C}$ a une solution conservatrice alors il existe un $\mathscr{C}'$ qui a une solution conservatrice.*

Preuve :

1. L'algorithme 2 page précédente est non déterministe et à chaque étape nous considérons un nombre fini de possibilités. Par conséquent, $\mathscr{C}'$ est formé d'un nombre fini de contraintes. Par construction, chaque contrainte de $\mathscr{C}'$ est une contrainte en une étape. Soit $\mathcal{C}'$ un système une étape de contraintes de $\mathscr{C}'$; la monotonie de $\mathcal{C}'$ provient de la monotonie de $\mathcal{C}$ et de la construction de $\mathscr{C}'$. Pour terminer la preuve il reste à montrer que $\mathcal{C}'$ est bien défini, nous observons que chaque terme apparaissant dans l'ensemble des hypothèses de chaque contrainte est soit un terme introduit par l'algorithme 2 page précédente (*i.e.* un terme dans S) soit un terme issu d'un ensemble d'hypothèses de la contrainte $\mathcal{C}$. Dans le premier cas, par construction, ce terme apparaît avant dans un terme but de la contrainte. Dans le second cas, nous concluons grâce au fait que $\mathcal{C}$ est bien défini.
2. Pour chaque contrainte $T_i \Vdash u_i \in \mathcal{C}$, il existe une contrainte $T_i \cup S_1 \cup \ldots \cup S_i \ \Vdash_1 \ u_i \in \mathscr{C}'$. Puisque σ est une solution de $\mathscr{C}'$ (par hypothèse), $u_i\sigma$ est déductible en une étape à partir de $T_i\sigma \cup S_1\sigma \cup \ldots \cup S_i\sigma$. Par construction de $\mathscr{C}'$, chaque terme de $S_j\sigma$ est déductible en utilisant uniquement des termes de $T_j\sigma$. Intuitivement, chaque nouvelle contrainte est obtenue en ordonnant des contraintes une étape issues du système de contraintes initial. Ainsi $u_i\sigma$ est déductible de $T_1\sigma \cup \ldots \cup T_i\sigma$, ce qui est équivalent à ce que $u_i\sigma$ soit déductible de $T_i\sigma$ par monotonie de $\mathcal{C}$.
3. Par hypothèse, pour chaque contrainte $T_i \Vdash u_i \in \mathcal{C}$, il existe une preuve P_i de $T_i\sigma \vdash u_i\sigma$. Puisque σ est une solution conservatrice et grâce au lemme 39 page 152 (lemme de localité), nous supposons que P_i contient que des termes dans $St_{\mathsf{E}}(\mathcal{C})\sigma$. Soit $S'_i = \{s \in St_{\mathsf{E}}(\mathcal{C}) \mid T_i\sigma \vdash s\sigma\}$, S'_i contient tous les sous-termes de $\mathcal{C}$ dont l'instance est déductible par σ à l'étape i en utilisant les termes dans T_i. Remarquons que grâce à la monotonie de $\mathcal{C}$, nous avons $S'_i \subseteq S'_{i+1}$ pour tout $1 \leq i < \ell$.
   Maintenant, soit $S_1 = S'_1$ et $S_i = S'_i \setminus (S'_1 \cup \ldots \cup S'_{i-1})$, S_i contient tous les sous-termes de $\mathcal{C}$ dont l'instance par σ est déductible à l'étape i et pas avant. Pour chaque i, nous ordonnons les éléments dans S_i tels que : pour tout $s, s' \in S_i$ si $T_i\sigma \vdash s\sigma$ est la racine d'une sous-preuve d'une preuve minimale de $T_i\sigma \vdash s'\sigma$, alors $s \prec_i s'$. Par conséquent, pour chaque $s \in S_i$, nous avons $s\sigma$ est déductible en une étape à partir $S_1\sigma \cup \ldots \cup S_{i-1}\sigma \cup \{s'\sigma \mid s' \prec_i s \text{ et } s' \in S_i\}$. Nous montrons que $u_i\sigma$ est déductible en une étape à partir de $T_i\sigma \cup S_1\sigma \cup \ldots \cup S_i\sigma$. Par définition de S_j et grâce au fait que $u_i\sigma$ est déductible au moins à l'étape i, nous savons que $u_i \in S_1 \cup \ldots \cup S_i$ et donc $u_i\sigma \in T_i\sigma \cup S_1\sigma \cup \ldots \cup S_i\sigma$. Nous avons donc $St_{\mathsf{E}}(\mathcal{C}') = St_{\mathsf{E}}(\mathcal{C})$. Par conséquent σ est une solution conservatrice de $\mathcal{C}'$.

□

10.6 D'un système une étape bien défini vers un système M_{E} bien défini.

Dans cette section, nous devinons parmi les sous-termes de $\mathcal{C}$ ceux qui vont être déduits par l'intrus. Ainsi le lemme 41 page suivante nous permet de réduire la satisfiabilité d'un système de contraintes une étape en la satisfiabilité d'un système de contraintes M_{E}. Nous devinons d'abord un ensemble d'égalité R entre les sous-termes. Ensuite, nous choisissons un E-unificateur de R parmi l'ensemble fini des possibilités données par le théorème 7 (démontré dans le chapitre 8 page 125).

Théorème 7 *L'unification dans la théorie* ACUNh *est finitaire, et il existe un algorithme pour calculer l'ensemble fini et complet des unificateurs* $mgu_{\mathsf{E}}(R)$ *d'un problème d'unification* R.

Lemme 41 *Soit $\mathcal{C}$ un système de contraintes une étape bien défini, $\mathcal{P} = \{\bigwedge_{(s_1,s_2)\in S'} s_1 = s_2 \mid S' \subseteq St_{\mathsf{E}}(\mathcal{C})^2\}$, $R \in \mathcal{P}$, $\theta \in mgu_{\mathsf{E}}(R)$. Considérons $\mathcal{C}_\theta = \{T\theta \Vdash_{\mathsf{M_E}} u\theta \mid T \Vdash_1 u \in \mathcal{C}$ et $T\theta \nvdash_{\mathsf{DY}} u\theta\}$.*

1. *$\mathcal{C}_\theta$ est un ensemble fini de contraintes $\mathsf{M_E}$ bien défini.*
2. *Si $\mathcal{C}_\theta$ a une solution alors $\mathcal{C}$ a une solution.*
3. *Si $\mathcal{C}$ a une solution conservatrice alors il existe une solution $\mathcal{C}_\theta$ qui possède une solution non effondrante (définition 33 page 122).*

Preuve :

1. $\mathcal{P}$ est un ensemble fini de système d'équations car $St_{\mathsf{E}}(\mathcal{C})$ est un ensemble fini de termes, par construction des sous-termes. Chaque système d'équations représente un problème d'unification qui possède un ensemble fini d'unificateurs grâce au théorème 7 page précédente. Soit θ un tel unificateur et $\mathcal{C}_\theta$ le système de contraintes obtenu en appliquant l'unificateur θ. Le système de contraintes $\mathcal{C}_\theta$ ne contient que des contraintes $\mathsf{M_E}$. Maintenant nous montrons que $\mathcal{C}_\theta$ est bien défini. Soit σ une substitution et notons $\mathcal{C}' = \mathcal{C}\theta\sigma$. Comme $\mathcal{C}$ est bien défini, nous savons que $\mathcal{C}'$ satisfait bien la propriété d'origination. Il nous reste donc à montrer que la suppression de contraintes effectuée n'affecte pas la « bonne définition » de $\mathcal{C}_\theta$. Plus précisément effacer une contrainte $T\theta \Vdash_1 u\theta$ pourrait contredire l'origination *i.e.* il existe $x \in vars(u\theta)$ et $x \notin vars(T\theta)$. Cela n'est pas possible car par hypothèse, une telle contrainte $T\theta \Vdash_1 u\theta$ est telle que $T\theta \vdash_{\mathsf{DY}} u\theta$ et donc $vars(u\theta) \subseteq vars(T\theta)$ ce qui prouve que l'origination est bien préservée.
2. Soit $\mathcal{C}_\theta$ une système de contraintes $\mathsf{M_E}$ construit à partir de $\mathcal{C}$ en appliquant la substitution θ obtenue par la procédure du lemme 41. θ' une solution de $\mathcal{C}_\theta$, nous montrons que $\theta\theta'$ est une solution de $\mathcal{C}$. Soit $T \Vdash_1 u \in \mathcal{C}$, alors $u\theta$ est déductible en une étape à partir de $T\theta$ (sans aucune instance) ou $T\theta \Vdash_{\mathsf{M_E}} u\theta \in \mathcal{C}'$. Dans les deux cas, cela signifie que $u\theta\theta'$ est déductible en une étape à partir de $T\theta\theta'$. Par conséquent $\theta\theta'$ est une solution de $\mathcal{C}$.
3. Soit σ une solution conservatrice de $\mathcal{C}$ et $R = \{(s_1, s_2) \mid s_1, s_2 \in St_{\mathsf{E}}(\mathcal{C})$ et $s_1\sigma =_{\mathsf{E}} s_2\sigma\}$.

   Nous posons $\theta \in mgu_{\mathsf{E}}(R)$ tel que θ est plus général que σ, il existe θ' une substitution telle que $\theta \circ \theta' =_{\mathsf{E}} \sigma$. Rappelons que $\mathcal{C}_\theta = \{T\theta \Vdash_{\mathsf{M_E}} u\theta \mid T \Vdash_1 u \in \mathcal{C}$ et $T\theta \nvdash_{\mathsf{DY}} u\theta\}$. Nous montrons que θ' est une solution de $\mathcal{C}_\theta$, *i.e.* $u\theta'$ est déductible en une étape $\mathsf{M_E}$ à partir de $T\theta'$ pour chaque contrainte $\mathsf{M_E}$ dans $\mathcal{C}_\theta$.

   Considérons $T \Vdash_1 u \in \mathcal{C}$ tel que $u\sigma$ soit DY déductible en une étape à partir de $T\sigma$. Nous montrons que $u\theta$ est DY déductible en une étape à partir de $T\theta$. Par conséquent, les contraintes déductibles en une étape dans $\mathcal{C}_\theta$ sont celles qui n'utilisent que la contrainte ($\mathsf{M_E}$). Si $u\sigma \in T\sigma$, alors il existe $t \in T$ tel que $t\sigma = u\sigma$. Par conséquent, nous avons $t\theta = u\theta$ car $t, u \in St_{\mathsf{E}}(\mathcal{C})$. Donc $u\theta \in T\theta$: $u\theta$ est déductible en une étape à partir de $T\theta$. Sinon, $u\sigma$ est déductible en une étape à partir de $T\sigma$ en utilisant une règle d'inférence parmi (C), (UL), (UR), (D).

   Dans le premier cas (C), nous avons $u\sigma = f(v_1, \ldots, v_n)$ pour $v_i \in T\sigma$ et $f \in \mathcal{F} \setminus \{0, h, +\}$, par conséquent, pour tout $i \leq n$ il existe un $v'_i \in T$ tel que $v_i = v'_i\sigma$. Il y a deux possibilités :
   - Si u n'est pas une variable, alors $u = f(u'_1, \ldots, u'_n)$ et nous avons $u'_i, v'_i \in St_{\mathsf{E}}(\mathcal{C})$ et $u'_i\sigma = v'_i\sigma$ pour chaque $i \leq n$. Nous déduisons que $u'_i\theta = v'_i\theta$, par conséquent $T\theta \vdash_{\mathsf{DY}} u\theta$.
   - Si u est une variable, comme σ est une solution conservatrice de $\mathcal{C}$ alors il existe $t \in St_{\mathsf{E}}(\mathcal{C}) \setminus vars(\mathcal{C})$ tel que $u\sigma =_{\mathsf{E}} t\sigma$. Par conséquent $t = f(t_1, \ldots, t_n)$ pour $t_i \in St_{\mathsf{E}}(\mathcal{C})$, nous en déduisons $t_i = v'_i$ et donc $T\theta \vdash_{\mathsf{DY}} u\theta$.

   Les autres cas (UR), (UL) et (D) sont similaires.

   Finalement nous montrons que θ' est une solution non effondrante de $\mathcal{C}_\theta$. Soit $u, v \in St_{\mathsf{E}}(\mathcal{C}_\theta) \setminus \mathcal{X}$, grâce à la proposition 14 page 147, $u, v \in St_{\mathsf{E}}(\mathcal{C})\theta \cup \bigcup_{x \in vars(\mathcal{C})} St_{\mathsf{E}}(x\theta)$, en conséquence

$u, v \in St_{\mathsf{E}}(\mathcal{C})\theta$. Par le lemme 34 page 135, il existe $u_1, v_1 \in St_{\mathsf{E}}(\mathcal{C})$ tels que $u = u_1\theta$ et $v = v_1\theta$. Supposons que $u\theta' = v\theta'$, nous obtenons $u_1\theta\theta' = v_1\theta\theta'$, donc $u_1\sigma = v_1\sigma$ et $(u_1, v_1) \in R$. Nous déduisons donc $u_1\theta = v_1\theta$ par construction de θ, *i.e.* $u = v$.

□

Nous pouvons alors nous concentrer uniquement sur les contraintes de cette forme $T \Vdash_1 u$ pour lesquelles les contraintes de ce type $T \vdash_{\mathsf{DY}} u$ sont déjà résolues. Nous restreignons donc notre attention uniquement sur les solutions non effondrantes. Considérer des solutions non effondrantes est un point important pour transformer un tel système de contraintes en un système dépendant d'équations, lors de la prochaine étape de notre procédure décrite dans la section suivante.

10.7 D'un système M_{E} bien défini vers un système dépendant d'équations.

Maintenant, nous résolvons un système M_{E} de contraintes où il est suffisant de regarder les solutions non effondrantes. Soit $\mathcal{C}$ un système de contraintes, où $T_0 = \{t_1, \ldots, t_n\}$, les $t_1, \ldots, t_n$ sont de termes constants et les $t_{n+1}, \ldots, t_{n+k-1}$, les u_i, pour $1 \leq i \leq k$ sont des termes non forcément clos. Nous supposons que les hypothèses de la $i + 1$ème contrainte *i.e.* $t_1, \ldots t_{n+i}$ contiennent exactement un terme de plus que l'hypothèse de la $i$ème contrainte. Ceci peut être fait en dupliquant des termes et en ajoutant certaines contraintes.

$$\begin{cases} t_1, \ldots, t_n & \Vdash_{\mathsf{M}_{\mathsf{E}}} & u_1 \\ t_1, \ldots, t_n, t_{n+1} & \Vdash_{\mathsf{M}_{\mathsf{E}}} & u_2 \\ & \vdots & \\ t_1, \ldots, t_n, t_{n+1}, \ldots, t_{n+k-1} & \Vdash_{\mathsf{M}_{\mathsf{E}}} & u_k \end{cases}$$

Soit $\sum_{i=0}^{n} b_i h^i$ où $b_i \in \mathbb{Z}/2\mathbb{Z}$ est un polynôme de $\mathbb{Z}/2\mathbb{Z}[h]$, nous rappelons que le produit $\odot$ d'un polynôme par un terme est un terme défini comme suit :

$$(\sum_{i=0}^{n} b_i h^i) \odot t = \sum_{i=0 \ | \ b_i \neq 0}^{n} h^i(t)$$

Par exemple $(h^2 + 1) \odot (x + a) = h^2(x) + x + h^2(a) + a$.

Résoudre un tel système de contraintes revient à résoudre le système d'équations $\mathcal{E}$ sur les variables de $\mathcal{C}$ et les $z[i, j]$ (variables de contexte).

$$\mathcal{E} \quad := \quad \begin{cases} z[1,1] \odot t_1 + \ldots + z[1,n] \odot t_n = u_1 \\ z[2,1] \odot t_1 + \ldots + z[2,n] \odot t_n + z[2,n+1] \odot t_{n+1} = u_2 \\ \vdots \\ z[k,1] \odot t_1 + \ldots + z[k,n] \odot t_n + \ldots + z[k,n+k-1] \odot t_{n+k-1} = u_k \end{cases}$$

Malheureusement, nous ne savons pas résoudre des équations qui contiennent des termes standards. Dans la suite de cette section, nous abstrayons ces termes par de nouvelles constantes. Cette abstraction doit préserver la « bonne définition » du système initial de contraintes M_{E}. Mais, un système de contraintes bien défini n'est pas forcément bien défini après abstraction des termes standards comme le montre l'exemple 38 page suivante .

Exemple 38 *Considérons le système de contraintes suivant :*

$$\begin{cases} a & \Vdash_{\mathsf{M}_\mathsf{E}} & \{x_1\}_a \\ a, x_1 & \Vdash_{\mathsf{M}_\mathsf{E}} & x_2 \oplus a \end{cases}$$

Après abstraction des termes standards, nous obtenons :

$$\begin{cases} a & \Vdash_{\mathsf{M}_\mathsf{E}} & d \\ a, x_1, & \Vdash_{\mathsf{M}_\mathsf{E}} & x_2 \oplus a \end{cases}$$

Ce système de contraintes n'est pas bien défini.

Nous introduisons donc la notion de système de contraintes facteur préservant, et nous montrons alors que l'abstraction sur les systèmes de contraintes facteur préservant conserve la « bonne définition » du système de contraintes. De plus nous démontrons que nous pouvons toujours obtenir un système facteur préservant. Le lemme 42 page suivante montre que s'il existe une solution non effondrante du système de contraintes alors il existe un système facteur préservant .Or d'après le lemme 41 page 155 nous savons qu'il existe toujours une solution non effondrante. Ensuite, nous montrons qu'un système de contraintes bien défini et facteur préservant satisfait une propriété de « dépendance » entre les termes des hypothèses et les conclusions précédentes sur la signature complète (lemme 43 page 160). Cette propriété est préservée par abstraction, nous obtenons donc un système sur la signature restreinte qui possède cette propriété. Nous montrons dans la section 10.7.3 page 163 que cette propriété introduite précédemment est équivalente sur la signature restreinte à la « bonne définition » des systèmes de contraintes. Ainsi nous pouvons résoudre des systèmes de contraintes bien définis sur la signature réduite $\{0, h, +\}$ ce qui est équivalent à résoudre des systèmes dépendants d'équations (définition 48 page 139). Grâce au chapitre 9.1 page 138, nous savons résoudre ce genre de système d'équations.

10.7.1 Système de contraintes facteur préservant.

Intuitivement un système de contraintes est dit *facteur préservant* si tous ses facteurs apparaissent pour la première fois dans une des hypothèses du système de contraintes.

Définition 62 (Système de contraintes facteur préservant) *Un système de contraintes* M_E *est* facteur préservant *si pour tout* i, $1 \leq i \leq k$, *nous avons que* $Fact_{\mathsf{E}}(u_i) \backslash \mathcal{X} \subseteq \bigcup_{j=1}^{j=n+i-1} Fact_{\mathsf{E}}(t_j)$.

Exemple 39 *Le système suivant est facteur préservant :*

$$\begin{cases} \langle a, b\rangle & \Vdash_{\mathsf{M}_\mathsf{E}} & x_1 \oplus \langle a, b\rangle \\ \langle a, b\rangle, \{x_1\}_b & \Vdash_{\mathsf{M}_\mathsf{E}} & x_2 \oplus \{x_1\}_b \\ \langle a, b\rangle, \{x_1\}_b, x_2 & \Vdash_{\mathsf{M}_\mathsf{E}} & c \end{cases}$$

Le système suivant n'est pas facteur préservant, car les facteurs $\langle a, b\rangle$ *et* $\{x_1\}_b$ *des termes en conclusion ne sont pas présents dans leurs hypothèses respectives :*

$$\begin{cases} a, b & \Vdash_{\mathsf{M}_\mathsf{E}} & x_1 \oplus \langle a, b\rangle \\ a, b, x_1 & \Vdash_{\mathsf{M}_\mathsf{E}} & x_2 \oplus \{x_1\}_b \\ a, b, x_1, x_2 & \Vdash_{\mathsf{M}_\mathsf{E}} & c \end{cases}$$

Cette notion est importante pour assurer que la « bonne définition » d'un système est préservée lorsque nous faisons une abstraction en remplaçant les facteurs par de nouvelles constantes dans le lemme 44 page 162. Nous montrons d'abord que nous travaillons bien avec des systèmes de contraintes facteurs préservants.

Lemme 42 *Si un système* $\mathsf{M_E}$ *de contraintes bien défini* $\mathcal{C}$ *a une solution non effondrante alors il est facteur préservant.*

Preuve : Soit $\mathcal{C} = \{t_1, \ldots, t_{n+k-1} \Vdash_{\mathsf{M_E}} u_i\}_{i=1,\ldots,k}$ un système de contraintes $\mathsf{M_E}$ bien défini et σ une solution non effondrante de $\mathsf{M_E}$. Nous montrons d'abord par induction sur le nombre de contraintes les deux points suivants :

1. $Fact_{\mathsf{E}}(u_i\sigma) \subseteq (Fact_{\mathsf{E}}(t_1, \ldots, t_{n+i-1}) \setminus \mathcal{X})\sigma$.
2. Pour tout x tel que $i = \min\{j \mid x \in vars(u_j)\}$ alors $Fact_{\mathsf{E}}(x\sigma) \subseteq (Fact_{\mathsf{E}}(t_1, \ldots, t_{n+i-1}) \setminus \mathcal{X})\sigma$.

Preuve :

Cas de base : Si $i = 1$ alors $t_1, \ldots, t_n$ sont clos. Nous avons $Fact_{\mathsf{E}}(u_1\sigma) \subseteq Fact_{\mathsf{E}}(t_1, \ldots, t_n)$. Puisque $Fact_{\mathsf{E}}(t_1, \ldots, t_n) \subseteq (Fact_{\mathsf{E}}(t_1, \ldots, t_n) \setminus \mathcal{X})\sigma$, nous avons le premier point. De plus, dans le facteur v de u_1 il n'y a pas de variable x, sinon nous aurions que $v\sigma = t_g$ pour un terme clos $t_g \in St_{\mathsf{E}}(t_1, \ldots, t_n)$ ce qui contredit le fait que σ soit une substitution non effondrante. Puisque, x doit être un facteur de u_1, alors nous obtenons le second point de la même manière que pour le premier point.

Induction : Si $i > 1$ alors nous savons que $Fact_{\mathsf{E}}(u_i\sigma) \subseteq (Fact_{\mathsf{E}}(t_1, \ldots, t_{n+i-1}))\sigma$. Si pour $v \in Fact_{\mathsf{E}}(u_i\sigma)$ nous avons que $v = x\sigma$ pour $x \in Fact_{\mathsf{E}}(t_j)$ avec $1 \leq j < n+i-1$, alors nous concluons par induction que $v \in (Fact_{\mathsf{E}}(t_1, \ldots, t_{n+i-1}))\sigma$, ce qui prouve le premier point. Maintenant soit $x \in vars(u_i\sigma)$, mais $x \notin vars(u_j\sigma)$ pour $j < i$. Par conséquent, comme le système de contraintes est bien défini, $x \notin vars(t_1, \ldots, t_{n+i-1})$. Donc x ne peut pas apparaître à l'intérieur d'un facteur de u_i, car cela contredirait le fait que σ soit une solution non effondrante. En conséquence, $x \in Fact_{\mathsf{E}}(u_i)$, et nous concluons comme précédemment par hypothèse d'induction.

□

Maintenant, soit $T_i = \{t_j \mid 1 \leq j \leq n+i-1\}$ et $v \in Fact_{\mathsf{E}}(u_i) \setminus \mathcal{X}$. Il existe $v' \in Fact_{\mathsf{E}}(T_i) \setminus \mathcal{X}$ tel que $v\sigma = v'\sigma$. Puisque σ est une substitution non effondrante, nous avons $v = v'$, cela nous permet de conclure que $\mathcal{C}$ est facteur préservant. □

La propriété de préservation des facteurs est facilement vérifiable par une analyse syntaxique du système de contraintes. Grâce au lemme 41 page 155 et au lemme 42, si le système de contraintes obtenu n'est pas facteur préservant alors le système initial n'a pas de solution. Nous considérons donc dans la suite un système de contraintes facteur préservant.

10.7.2 Réduction de la signature.

Dans le lemme 44 page 162, nous réduisons la satisfiabilité d'un système de contraintes $\mathsf{M_E}$ facteur préservant à la satifiabilité d'un système de contraintes $\mathsf{M_E}$ bien défini sur la signature réduite à 0, $\oplus$, h, et un ensemble de nouvelles constantes.

Nous définissons une nouvelle notion de sous-termes pour caractériser les systèmes de contraintes bien définis.

Définition 63 (Sous-terme non-standard) *L'ensemble des* sous-termes non-standard *d'un terme* t*, dénoté par* $NSt_{\mathsf{E}}(t)$*, est défini comme suit :*

$$\begin{array}{lcll} NSt(f(t_1, \ldots, t_n)) & = & \bigcup_{i=1}^{n} NSt_{\mathsf{E}}(t_i) & si\ f \notin sig(\mathsf{E}) \\ NSt_{\mathsf{E}}(t) & = & \{t\} \cup \bigcup_{s \in Fact_E(t) \setminus \mathcal{X}} NSt_{\mathsf{E}}(s) & sinon \end{array}$$

Exemple 40 *Soit* $t = h(x_1) + x_2 + \langle x_3, x_4 + x_5 \rangle$*. Nous avons* $NSt_{\mathsf{E}}(t) = \{t, x_3, x_4 + x_5\}$*.*

Considérons le système de contraintes suivant :

$$\begin{cases} a & \Vdash_{\mathsf{M_E}} & x_1 \oplus x_2 \\ a, b & \Vdash_{\mathsf{M_E}} & x_1 \\ a, b, \langle h(x_1), a\rangle \oplus \langle h(x_2), a\rangle & \Vdash_{\mathsf{M_E}} & a \oplus b \end{cases}$$

Ce système est bien défini et facteur préservant.

Si nous omettons la deuxième contrainte, nous obtenons :

$$\begin{cases} a & \Vdash_{\mathsf{M_E}} & x_1 \oplus x_2 \\ a, b, \langle h(x_1), a\rangle \oplus \langle h(x_2), a\rangle & \Vdash_{\mathsf{M_E}} & a \oplus b \end{cases}$$

Ce système de contraintes n'est pas bien défini comme le montre la substitution $\theta\colon x_i \mapsto x_i \oplus W$.

En nous inspirant de la substitution proposée dans l'exemple 40 page ci-contre, la proposition 16 assure que si deux termes sont différents alors ils seront encore différents par une substitution de la forme suivante : $\theta = \{X \to X \oplus (c^X \odot W)\}$.

Proposition 16 *Soit $t_1, t_2 \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ avec $t_1 \neq t_2$. Soit $c^X \in \mathbb{Z}/2\mathbb{Z}[h]$ pour chaque $X \in \mathcal{X}$, soit $W \notin \mathcal{X}$ une nouvelle variable, et $\theta\colon \mathcal{X} \to \mathcal{T}(\mathcal{F}, \mathcal{X} \cup \{W\})$ une substitution telle que $X\theta = X \oplus c^X \odot W$ pour chaque $X \in \mathcal{X}$.Alors $t_1\theta \neq t_2\theta$.*

Preuve : Par induction sur la taille des termes t_1 et t_2. Dans le cas de base le résultat est vrai. Induction :

– Si t_1 et t_2 sont deux termes standards alors nous avons :

$$\begin{array}{rcl} t_1 & = & f_1(t_1^1, \ldots, t_1^n) \\ t_2 & = & f_2(t_2^1, \ldots, t_2^m) \end{array}$$

Si $f_1 \neq f_2$ alors $t_1\theta \neq t_2\theta$.
Si $f_1 = f_2$ alors $t_1^i \neq t_2^i$ pour un i, donc par hypothèse d'induction $t_1^i\theta \neq t_2^i\theta$, et $t_1\theta = f_1(t_1^1\theta, \ldots, t_1^n\theta) \neq f_2(t_2^1\theta, \ldots, t_2^n\theta) = t_2\theta$.

– Si t_1 est standard et t_2 n'est pas standard alors nous avons :

$$\begin{array}{rcl} t_1 & = & f_1(t_1^1, \ldots, t_1^n) \\ t_2 & = & \Sigma_{s \in Fact_{\mathsf{E}}(t_2)}(p^s \odot s) \end{array}$$

pour un polynôme $p^s \in \mathbb{Z}/2\mathbb{Z}[h]$, et $Fact_{\mathsf{E}}(t_2)$ qui contient au moins deux éléments. Par hypothèse d'induction pour tout $s_1, s_2 \in Fact_{\mathsf{E}}(t_2)$ avec $s_1 \neq s_2$, $s_1\theta \neq s_2\theta$. Par conséquent, $t_2\theta$ n'est pas standard, et comme $t_1\theta$ est standard nous concluons que $t_1\theta \neq t_2\theta$.

– Si les deux t_1 et t_2 ne sont pas standards alors nous considérons $F = Fact_{\mathsf{E}}(t_1) \cup Fact_{\mathsf{E}}(t_2)$. Nous décomposons alors t_1 et t_2 tels que :

$$\begin{array}{rcl} t_1 & = & \Sigma_{X \in F \cap \mathcal{X}}(p_1^X \odot X) \oplus \Sigma_{s \in F \setminus \mathcal{X}}(p_1^s \odot s) \\ t_2 & = & \Sigma_{X \in F \cap \mathcal{X}}(p_2^X \odot X) \oplus \Sigma_{s \in F \setminus \mathcal{X}}(p_2^s \odot s) \end{array}$$

Donc, par définition de θ, nous obtenons :

$$\begin{array}{rcl} t_1\theta & = & \Sigma_{X \in F \cap \mathcal{X}}(p_1^X \odot X) \oplus \Sigma_{X \in F \cap \mathcal{X}}((p_1^X.c^X) \odot W) \oplus \Sigma_{s \in F \setminus \mathcal{X}}(p_1^s \odot s\theta) \\ t_2\theta & = & \Sigma_{X \in F \cap \mathcal{X}}(p_2^X \odot X) \oplus \Sigma_{X \in F \cap \mathcal{X}}((p_2^X.c^X) \odot W) \oplus \Sigma_{s \in F \setminus \mathcal{X}}(p_2^s \odot s\theta) \end{array}$$

Puisque $t_1 \neq t_2$ nous avons deux cas :

– Premier cas, $p_1^X \neq p_2^X$ pour un $X \in F \cap \mathcal{X}$. Par conséquence, comme $W \neq X$, nous obtenons que X apparaît dans $t_1\theta$ comme coefficient de p_1^X et dans $t_2\theta$ comme coefficient de p_2^X, donc $t_1\theta \neq t_2\theta$.
– Deuxième cas, $p_1^s \neq p_2^s$ pour $s \in F \setminus \mathcal{X}$. Par hypothèse d'induction, $s\theta \neq s'\theta$ pour chaque $s \in F \setminus \mathcal{X}$ avec $s \neq s'$. Par conséquent, $s\theta$ apparaît dans $t_1\theta$ avec coefficient p_1^s et dans $t_2\theta$ avec coefficient p_2^s, donc $t_1\theta \neq t_2\theta$.

□

Cette proposition est utilisée dans la preuve du lemme 43 (fait 2) pour assurer que θ (celui défini dans la proposition 16 page précédente) n 'annule pas deux termes standards distincts dans t. Plus précisément, si s est une variable non standard sous-terme de t, alors $s\theta$ est aussi une variable sous-terme de $t\theta$.

Nous avons maintenant tous les éléments pour faire le lien dans le lemme 43 entre la notion de système de contraintes facteur préservant bien défini et la dépendance entre vecteurs (définition 45 page 138).

Nous dénotons $vars(\mathcal{C}) = \{x_1, \ldots, x_p\}$ l'ensemble des variables du système de contraintes $\mathcal{C}$. Chaque t_i (ou u_i) s'écrit $t_i^{x_1} \odot x_1 + \ldots t_i^{x_p} \odot x_p + t_i^0$ où les $t_i^{x_v}$ sont des éléments de $\mathbb{Z}/2\mathbb{Z}[h]$ et $Fact_{\mathsf{E}}(t_i^0) \cap \mathcal{X} = \emptyset$. Nous le dénotons par le vecteur $\overline{t_i} = (t_i^{x_1}, \ldots, t_i^{x_p})$. Nous définissons maintenant la base de vecteurs « significatifs » de $\mathcal{C}$, cette définition formalise la construction de l'ensemble L du chapitre 9 page 137.

Définition 64 (Vecteur définissant de $\mathcal{C}$) *Soit $\mathcal{C} = \{t_1, \ldots, t_{n+i-1} \Vdash_{\mathsf{M_E}} u_i\}_{i=1,\ldots,k}$ un système de contraintes, l'ensemble d'indices $L = L_k$ représente les vecteurs définissants de $\mathcal{C}$ et l'ensemble L_k est définit récursivement comme suit :*
- *$L_0 = \emptyset$*
- *$L_{i+1} = L_i \cup \{i+1\}$ si $\{\overline{u}_i\} \cup \{\overline{u}_j \mid j \in L_i\}$ est indépendant*
- *$L_{i+1} = L_i$ sinon*

Nous notons $\mathcal{B}_i = \{\overline{u_j} \mid j \in L, j \leq i\}$ et $\mathcal{B} = \mathcal{B}_k$.

Exemple 41 *Reprenons l'exemple 40 page 158*

$$\begin{cases} a & \Vdash_{\mathsf{M_E}} \quad x_1 \oplus x_2 \\ a, b & \Vdash_{\mathsf{M_E}} \quad x_1 \\ a, b, \langle h(x_1), a\rangle \oplus \langle h(x_2), a\rangle & \Vdash_{\mathsf{M_E}} \quad a \oplus b \end{cases}$$

Nous avons $L = \{1, 2\}$, $\overline{u_1} = (1, 1)$ et $\overline{u_2} = (1, 0)$. Nous calculons $NSt_{\mathsf{E}}(\langle h(x_1), a\rangle + \langle h(x_2), a\rangle) = \{\langle h(x_1), a\rangle + \langle h(x_2), a\rangle; h(x_1); h(x_2)\}$. Les ensembles de vecteurs $\{(1, 1), (1, 0), (0, 0)\}$, $\{(1, 1), (1, 0), (h, 0)\}$ et $\{(1, 1), (1, 0), (0, h)\}$ sont dépendants. D'après le lemme 43 alors ce système est bien défini et facteur préservant.

Si nous omettons la deuxième contrainte, nous obtenons :

$$\begin{cases} a & \Vdash_{\mathsf{M_E}} \quad x_1 \oplus x_2 \\ a, b, \langle h(x_1), a\rangle \oplus \langle h(x_2), a\rangle & \Vdash_{\mathsf{M_E}} \quad a \oplus b \end{cases}$$

Nous obtenons $L = \{1\}$ et l'ensemble de vecteurs sets $\{(1, 1), (h, 0)\}$ est indépendant et donc ce système de contraintes n'est pas bien défini comme le montre la substitution $\theta\colon x_i \mapsto x_i \oplus W$.

Lemme 43 *Soit $\mathcal{C} = \{t_1, \ldots, t_{n+i-1} \Vdash_{\mathsf{M_E}} u_i\}_{i=1,\ldots,k}$ un système de contraintes $\mathsf{M_E}$ bien défini et facteur préservant. Alors pour chaque $i \leq k$ et $s \in NSt_{\mathsf{E}}(t_{n+i-1})$ l'ensemble de vecteurs $\{\overline{s}\} \cup \mathcal{B}_{i-1}$ est dépendant.*

Preuve : Nous effectuons la preuve par induction sur le nombre de contraintes $i = 1, \ldots, k$ du système de contraintes $\mathcal{C}$.

Cas de base : Si $i = 1$ alors $t_1, \ldots, t_n$ clos, et par conséquent $\overline{s} = (0, \ldots, 0)$ pour chaque $s \in NSt_{\mathsf{E}}(t_n)$. Nous concluons donc que l'ensemble $\{(0, \ldots, 0)\}$ est dépendant.

Induction : Soit $1 < i \leq k$, nous raisonnons par contradiction. Supposons que $s \in NSt_{\mathsf{E}}(t_{n+1-1})$ tel que $\{\overline{s}\} \cup \mathcal{B}_{i-1}$ soient indépendants. Nous construisons une substitution θ telle que $\mathcal{C}\theta$ n'est pas bien défini. L'idée de cette construction généralise le point mis en évidence dans l'exemple 40 page 158. Grâce à la proposition 13 page 138, il existe $Q \in \mathbb{Z}/2\mathbb{Z}[h], Q \neq 0$, et un vecteur $(c^{X_1}, \ldots, c^{X_p}) \in (\mathbb{Z}/2\mathbb{Z}[h])^p$ tel que

$$\begin{pmatrix} u_1^{X_1} & \ldots & u_1^{X_p} \\ \vdots & & \vdots \\ u_{i-1}^{X_1} & \ldots & u_{i-1}^{X_p} \\ s^{X_1} & \ldots & s^{X_p} \end{pmatrix} \cdot \begin{pmatrix} c^{X_1} \\ \vdots \\ c^{X_p} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ Q \end{pmatrix}$$

où seuls les vecteurs lignes $\overline{u_j}$ apparaissent dans la matrice pour les $j \in L$. Nous définissons la substitution $\theta \colon \{X_1, \ldots, X_p\} \to \mathcal{T}(F, \{X_1, \ldots, X_p, W\})$ par

$$X\theta = X \oplus c^X \odot W$$

pour tout $X \in \{X_1, \ldots, X_p\}$. Nous prouvons les trois faits suivants qui nous permettront de conclure.

Fait 1 : Soit $j < i$, pour tous sous-termes v de t_{n+j-1} nous avons $W \notin vars(v\theta)$ (*). *Preuve* : Nous distinguons deux cas :

– Si v est un terme standard, tel que $v = f(v_1, \ldots, v_n)$ avec f standard alors $v\theta = f(v_1\theta, \ldots, v_n\theta)$ alors le Fait 1 découle de l'application de l'hypothèse d'induction (*) aux termes v_i.

– Sinon nous décomposons :

$$v = \Sigma_{i=1,\ldots,p}(v^{X_i} \odot X_i) \oplus \Sigma_{f \in Fact_{\mathsf{E}}(v) \setminus \mathcal{X}}(v^f \odot f)$$

Par définition de θ nous obtenons que :

$$\begin{aligned} v\theta &= \Sigma_{i=1,\ldots,p}(v^{X_i} \odot X_i\theta) \oplus \Sigma_{f \in Fact_{\mathsf{E}}(v) \setminus \mathcal{X}}(v^f \odot f\theta) \\ &= \underbrace{\Sigma_{i=1,\ldots,p}(v^{X_i} \odot X_i)}_{\sigma_1} \oplus \underbrace{\Sigma_{i=1,\ldots,p}((v^{X_i}.c^{X_i}) \odot W)}_{\sigma_2} \oplus \underbrace{\Sigma_{f \in Fact_{\mathsf{E}}(v) \setminus \mathcal{X}}(v^f \odot f\theta)}_{\sigma_3} \end{aligned}$$

Puisque $W \neq X_1, \ldots, X_p$ nous avons que $W \notin vars(\sigma_1)$. Par hypothèse d'induction (*), $W \notin vars(\sigma_3)$. Par hypothèse d'induction du lemme 43 page précédente, l'ensemble de vecteurs $\{\overline{v}\} \cup \mathcal{B}_{j-1}$ est dépendant. Par conséquent, il existe des coefficients $\alpha, \alpha_1, \ldots, \alpha_{j-1} \in \mathbb{Z}/2\mathbb{Z}[h]$ tels que $\alpha \neq 0$ et :

$$\alpha.\overline{v} \oplus \alpha_1.\overline{u_1} \oplus \ldots \oplus \alpha_{j-1}.\overline{u_{j-1}} = (0, \ldots, 0)$$

c'est à dire

$$\alpha.\overline{v} = \alpha_1.\overline{u_1} \oplus \ldots \oplus \alpha_{j-1}.\overline{u_{j-1}}$$

Si nous appliquons la multiplication par un scalaire au vecteur $(c^{X_1}, \ldots, c^{X_p})$ des deux côtés de l'équation nous obtenons :

$$\alpha.\Sigma_{i=1}^p(v^{X_i}.c^{X_i}) = \alpha_1.\Sigma_{i=1}^p(u_1^{X_i}.c^{X_i}) \oplus \ldots \oplus \alpha_{j-1}.\Sigma_{i=1}^p(u_{j-1}^{X_i}.c^{X_i})$$

Par définition de θ le membre droit vaut 0 (car chaque partie de la somme vaut 0). Puisque $\alpha \neq 0$ nous concluons que $\Sigma_{i=1}^{p}(v^{X_i}.c^{X_1}) = 0$, *i.e.* $\sigma_2 = 0 \odot W = 0$, et $W \notin vars(\sigma_2)$. Par conséquent, $W \notin vars(t_{n+j-1}\theta)$ pour tout $j < n$.

□

Fait 2 : $W \notin vars(u_j\theta)$ pour tout $j < i$.

Preuve : Supposons le contraire *i.e.* $W \in vars(u_j\theta)$. Si $j \in L$ alors $\Sigma_{i=1}^{p}(u_j^{X_i}.c^{X_i}) = 0$ par construction de θ, et si $j \notin L$ nous avons que $\Sigma_{i=1}^{p}(u_j^{X_i}.c^{X_i}) = 0$ car dans ce cas $\overline{u_j}$ est dépendant de $\mathcal{B}_{j-1}$. Par conséquent, $f \in Fact_{\mathsf{E}}(u_j\theta) \setminus \mathcal{X}$ tel que $W \in vars(f)$. Ceci est possible seulement s'il existe $f' \in Fact(u_j)$ tel que $W \in f = f'\theta$. Comme le système de contraintes $\mathcal{C}$ est facteur préservant il existe un $j' \leq j < i$ tel que $f' \in Fact(t_{n+j'-1})$. Grâce à la proposition 16 page 159, pour chaque $f'' \in Fact_{\mathsf{E}}(t_{n+j'-1})$ où $f' \neq f''$ nous avons $f'\theta \neq f''\theta$, donc $W \in vars(t_{n+j'-1}\theta)$, ce qui contredit le fait 1 prouvé juste avant. □

Fait 3 : $W \in vars(t_{n+i-1}\theta)$.

Preuve : Nous décomposons

$$s = \Sigma_{i=1}^{p}(s^{X_i} \odot X_i) \oplus \Sigma_{f \in Fact(s) \setminus \mathcal{X}}(s^f \odot f)$$

Par définition de θ nous obtenons :

$$s\theta = \Sigma_{i=1}^{p}(s^{X_i} \odot X_i) \oplus \underbrace{\Sigma_{i=1}^{p}((s^{X_i}.c^{X_i}) \odot W)}_{=Q \odot W} \oplus \Sigma_{f \in Fact(s) \setminus \mathcal{X}}(s^f \odot f\sigma)$$

et donc $W \in s\theta$.

Si $s = t_{n+i-1}$ alors nous concluons immédiatement que $W \in vars(t_{n+i-1}\theta)$. Sinon, soit $f \in Fact_{\mathsf{E}}(t_{n+i-1}) \setminus \mathcal{X}$ tel que $s \in NSt_{\mathsf{E}}(f)$. Par la proposition 16 page 159, comme $f \in Fact_{\mathsf{E}}(t_{n+i-1}) \setminus \mathcal{X}$ donc $f\theta$ ne peut être annulé, nous avons alors $W \in vars(f\theta)$, et $W \in vars(t_{n+i-1}\theta)$.

□

Par conséquent, $W \in vars(t_{n+i-1}\theta)$ par le fait 2 et $W \notin vars(u_j\theta)$ pour tout $j < i$ grâce au fait 3, ce qui contredit la propriété d'origination de $\mathcal{C}\theta$, et donc contredit le fait que $\mathcal{C}$ soit bien défini.

□

Nous venons de montrer que si un système de contraintes est bien défini et facteur préservant, alors les vecteurs des sous-termes non standards de ces hypothèses sont dépendants des vecteurs des termes des conclusions précédentes.

Nous rappelons qu'un remplacement $\rho : M \to N$ est une bijection entre deux ensembles de termes M et N. Nous dénotons alors pour tous termes t par t^ρ le terme obtenu en remplaçant dans t toutes les occurrences « top-most » de sous-termes $s \in M$ par $s\rho$. Nous étendons cela au système de contraintes et aux substitutions *i.e.* en posant $x(\sigma^\rho) = (x\sigma)^\rho$ pour toutes les variables $x \in dom(\sigma)$.

En utilisant la nouvelle caractérisation de la section 10.7.3 page ci-contre des protocoles bien définis sur la signature restreinte à $\{0, h, +\}$, nous transformons par abstraction (application d'un remplacement) un système de contraintes bien défini et facteur préservant en un système bien défini sur la signature restreinte à $\{0, h, +\}$.

Lemme 44 *Soit $\mathcal{C}$ un système de contraintes M_{E} bien défini et facteur préservant tel que $F = Fact_{\mathsf{E}}(\mathcal{C}) \setminus \mathcal{X}$. Soit $\mathcal{F}_0$ une ensemble de nouvelles constantes de même cardinalité que F et $\rho : F \to \mathcal{F}_0$ une bijection.*

1. *$\mathcal{C}^\rho$ est bien défini.*
2. *$vars(\mathcal{C}^\rho) = vars(\mathcal{C})$.*
3. *Si $\mathcal{C}$ a une solution non effondrante alors $\mathcal{C}^\rho$ a une $\mathcal{F}_0 \cup \{0, h, +\}$-solution.*
4. *Si $\mathcal{C}^\rho$ a une $\mathcal{F}_0 \cup \{0, h, +\}$-solution alors $\mathcal{C}$ a solution.*

Preuve :

1. Par hypothèse, $\mathcal{C}$ est bien défini et facteur préservant, donc par le lemme 43 page 160, nous savons que pour tout $i \leq k$ et $s \in NSt_{\mathsf{E}}(t_{n+i-1})$, l'ensemble de vecteurs $\{\overline{s}\} \cup \{\overline{u_j} \mid j < i$ et $j \in L\}$ est indépendant. En particulier nous avons que pour tout $i \leq k$, l'ensemble de vecteurs $\{\overline{t_{n+i-1}}\} \cup \{\overline{u_j} \mid j < i$ et $j \in L\}$ est dépendant. Pour tous termes t, nous avons $\overline{t} = \overline{t^\rho}$, nous concluons en appliquant le lemme 45.
2. Nous supposons que $vars(\mathcal{C}^\rho) \subseteq vars(\mathcal{C})$ donc ρ n'introduit aucune nouvelle variable. Réciproquement, si $x \in vars(\mathcal{C})$ alors soit i le plus petit indice tel que $x \in vars(u_i)$. Par le même argument que précédemment, x doit avoir une occurrence dans u_i qui n'est pas dans un facteur, donc $x \in vars(\mathcal{C}^\rho)$.
3. Soit σ une solution non effondrante de $\mathcal{C}$, donc σ, $v_1\sigma = v_2\sigma$ implique que $v_1 = v_2$ et donc $v_1\rho = v_2\rho$ pour tout $v_1, v_2 \in Fact_{\mathsf{E}}(\mathcal{C}) \setminus \mathcal{X}$. Donc, σ^ρ est une solution de $\mathcal{C}^\rho$.
4. Soit σ une solution de $\mathcal{C}\rho$, donc $\sigma^{(\rho^{-1})}$ est une solution de $\mathcal{C}$.

□

Nous avons à résoudre des systèmes de contraintes bien définis sur la signature $\{0, h, +\}$. D'après la section 10.7.3 ces systèmes sont équivalents à des systèmes de contraintes avec une condition de dépendance entre les nouvelles hypothèses et les conclusions précédentes. Ainsi nous pouvons résoudre le système d'équations associé à ce système de contraintes grâce aux résultats du chapitre 9 page 137. Nous obtenons alors directement le corollaire 2.

Corollaire 2 *Soit $\mathcal{C}$ un système de contrainte $\mathsf{M_E}$ bien défini. Nous savons si le système d'équations $\mathcal{S}(\mathcal{C})$ associé à $\mathcal{C}$ a une solution.*

10.7.3 Une autre caractérisation des systèmes bien définis.

Le lemme 45 donne une caractérisation du fait qu'un système soit bien défini sur la signature restreinte $\mathcal{F}_0 \cup \{0, h, \oplus\}$.

Lemme 45 *Soit $\mathcal{C} = \{t_1, \ldots, t_{n+i-1} \Vdash_{\mathsf{M_E}} u_i\}_{i=1,\ldots,k}$ un système de contraintes $\mathsf{M_E}$ facteur préservant sur la signature $\{0, h, +\} \cup \mathcal{F}_0$. $\mathcal{C}$ est bien défini si et seulement si l'ensemble de vecteurs $\{\overline{t_{n+i-1}}\} \cup \{\overline{u}_j \mid j \in L_i\}$ est dépendant pour tout $i \leq k$.*

Preuve : ($\Leftarrow$) Premièrement, la monotonie est clairement satisfaite par abstraction. Deuxièmement, nous devons montrer que pour toutes substitutions θ, $\mathcal{C}\theta$ satisfait la propriété d'origination. Soit θ une substitution et t un terme qui apparaît dans un ensemble d'hypothèses de $\mathcal{C}$ tel que $t\theta$ contient une variable Z. Nous avons $t = t_{n+j}$ pour un j (sinon t serait un terme clos) et nous devons montrer que $Z \in u_i\theta$ pour un $i \leq j$. Par hypothèse, nous savons qu'il existe α et des α_i éléments de $\mathbb{Z}/2\mathbb{Z}[h]$ tels que $\sum \alpha_i \overline{u_i} + \alpha \overline{t_{n+j}} = 0$ et α, α_i sont tous non nuls. Remarquons que $\{\overline{u_i} \mid i \in L$ et $i \leq j\}$ est indépendant car c'est un sous-ensemble de $\mathcal{B}$. Ce qui implique que $\alpha \neq 0$. Nous avons donc :

$$\begin{array}{lrcl} & \alpha.\overline{t_{n+j}} & = & -\sum_{i\in L, i\leq j}\alpha_i.\overline{u_i} \\ \Rightarrow & \alpha.(\sum_{l=1}^{l=p} t_{n+j}^{X_l} + t_{n+j}^0 - t_{n+j}^0) & = & -\sum_{i\in L, i\leq j}\alpha_i.(\sum_{l=1}^{l=p} u_i^{X_l} + u_i^0 - u_i^0) \\ \Rightarrow & \alpha.(t_{n+j} - t_{n+j}^0) & = & -\sum_{i\in L, i\leq j}\alpha_i.(u_i - u_i^0) \\ \Rightarrow & \alpha.(t_{n+j}\theta - t_{n+j}^0) & = & -\sum_{i\in L, i\leq j}\alpha_i.(u_i\theta - u_i^0) \\ \Rightarrow & \alpha.t_{n+j}\theta & = & -\sum_{i\in L, i\leq j}\alpha_i.(u_i\theta - u_i^0) + \alpha t_{n+j}^0 \end{array}$$

Donc, $Z \in vars(t_{n+j}\theta)$ implique que $Z \in vars(u_i\theta)$ pour $i \in L$ et $i \leq j$.
($\Rightarrow$) Supposons qu'il existe $1 \leq j \leq k$ tel que $\{\overline{u_i} \mid i \in L \text{ et } i \leq j\} \cup \{\overline{t_{n+j}}\}$ soit indépendant. Par la proposition 13 page 138, il existe $Q \in \mathbb{Z}/2\mathbb{Z}[h]$ tel que le système suivant d'équations possède un solution sur $\mathbb{Z}/2\mathbb{Z}[h]$.

$$\begin{pmatrix} u_1^{X_1} & u_1^{X_2} & \dots & u_1^{X_p} \\ \vdots & & & \\ u_j^{X_1} & u_j^{X_2} & \dots & u_j^{X_p} \\ t_{n+j}^{X_1} & t_{n+j}^{X_2} & \dots & t_{n+j}^{X_p} \end{pmatrix} \cdot \begin{pmatrix} X_1 \\ \vdots \\ X_p \end{pmatrix} = Q. \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

Soit $(c_1, \dots, c_p)$ une solution de ce système d'équations. Soit Z une nouvelle variable et θ une substitution définie par $X_i \mapsto c_i.Z$ pour $1 \leq i \leq p$ où Z est une nouvelle variable. Par construction de θ, nous avons $u_i\theta = u_i^0$ pour chaque $i \in L$ tel que $i \leq j$ et $t_{n+j}\theta = Q.Z + t_{n+j}^0$. Nous avons trouvé une substitution θ telle que la variable Z apparaît pour la première fois en tant qu'hypothèse de $\mathcal{C}\theta$. ce qui contredit le fait que $\mathcal{C}$ soit bien défini. $\square$

10.8 Procédure de décision.

Nous pouvons maintenant prouver le résultat annoncé au début de cette section, qui reprend les différents éléments de la procédure décrite dans ce chapitre.

Théorème 8 *Savoir si un système de contraintes bien défini a une solution est décidable dans le modèle de Dolev-Yao étendu par la théorie équationnelle ACUNh.*

Preuve : La procédure de décision est le résultat de l'ensemble des sections précédentes :
- **Correction :** Si nous appliquons notre procédure à $\mathcal{C}$ un système de contraintes bien défini, si nous trouvons une solution alors cette solution satisfait bien $\mathcal{C}$. Soit un système de contraintes $\mathcal{C}$ bien défini nous obtenons $\mathcal{C}_1$ un système de $\mathsf{M_E}$ contraintes bien défini et facteur préservant en appliquant les premières étapes de la procédure grâce au lemme 40 page 153 et au lemme 41 page 155. Soit $\mathcal{C}_2$ un système de contraintes obtenu à partir de $\mathcal{C}_1$ en remplaçant les facteurs par de nouvelles constantes. Le système de contraintes $\mathcal{C}_2$ est bien défini grâce au lemme 44 page 162. Supposons que $\mathcal{S}(\mathcal{C}_2)$ (le système d'équations associé à $\mathcal{C}_2$) a une solution. Nous déduisons que le système de contraintes $\mathcal{C}_2$ a une solution, par conséquent le lemme 44 page 162 garantit que le système de contraintes $\mathcal{C}_1$ a une solution. Par le lemme 40 page 153 et le lemme 41 page 155 nous avons aussi une solution pour le système de contraintes $\mathcal{C}$, ce qui prouve la correction de notre procédure.
- **Complétude :** Supposons que σ est une solution du système de contraintes $\mathcal{C}$ alors grâce à notre procédure nous pouvons la calculer. Par le lemme 38 page 150, nous savons que σ est une solution conservatrice du système de contraintes $\mathcal{C}$. Soit $\mathscr{C}'$ le système de contraintes une étape bien défini obtenu par l'application de l'algorithme de la section 10.5 page 153 sur le

système de contraintes $\mathcal{C}$. Par le lemme 40 page 153 σ est une solution conservatrice de $\mathscr{C}'$. Grâce au lemme 41 page 155, il existe $\mathcal{C}_\theta$ un système M_E de contrainte qui a une solution non effondrante. Donc $\mathcal{C}_\theta$ est un système facteur préservant par le lemme 42 page 158. Grâce au lemme 44 page 162, $\mathcal{C}_\theta^\rho$ a une solution sur $\{0, h, +\} \cup \mathcal{F}_0$. Donc le lemme 2 page 163 nous permet de conclure.

□

Ce théorème nous permet d'analyser le protocole TMN décrit à la section 2.5.2.4 page 39. En suivant notre procédure et en modélisant le chiffrement asymétrique du serveur par la fonction homomorphique h sur le « ou exclusif », nous retrouvons l'attaque similaire à celle donnée par G. J. Simmons en 1994 [DLLT05].

Chapitre 11

Conclusion et perspectives.

« If people do not believe that mathematics is simple,
it is only because they do not realize how complicated life is. »
John Louis von Neumann.

En observant les progrès réalisés ces dernières années en vérification de protocoles cryptographiques, nous remarquons que les nouvelles approches visent à affaiblir l'hypothèse du chiffrement parfait en augmentant les capacités de l'intrus afin d'analyser de manière plus réaliste les protocoles. Prendre en compte les propriétés algébriques des primitives cryptographiques et des spécifications des protocoles contribue à l'affaiblissement de cette hypothèse. Ces nouvelles considérations rendent beaucoup plus complexes l'analyse des protocoles. Dans cette thèse, nous avons développé de nouvelles méthodes formelles de vérification concernant la propriété de secret en présence de théories équationnelles homomorphiques pour les problèmes de déduction de l'intrus et de sécurité en nombre borné de sessions dans les protocoles cryptographiques. Ainsi nous avons affaibli l'hypothèse du chiffrement parfait.

Classification et modélisation : Nous avons d'abord, dans le chapitre 2, dégagé une liste des différents protocoles existants utilisant des propriétés algébriques soit dans les méthodes de chiffrement employées, soit dans la spécification même du protocole. Nous avons également, pour chacun des exemples de propriétés algébriques, exhibé lorsqu'ils existent les travaux de vérification essayant de les prendre en compte afin d'affaiblir l'hypothèse du chiffrement parfait. Nous soulignons parmi toutes ces propriétés algébriques que de nombreux protocoles utilisent la propriété d'homomorphisme. Dans le chapitre 3, nous présentons le modèle de l'intrus standard de Dolev-Yao qui modélise les capacités d'un intrus. Nous donnons ensuite l'extension classique de ce modèle par une théorie équationnelle représentée par un système de réécriture convergent (modulo AC). Nous nous sommes alors focalisés sur la vérification de protocoles cryptographiques en présence de la propriété d'homomorphisme. Nous nous sommes intéressés plus particulièrement à la propriété d'homomorphisme avec un opérateur associatif et commutatif, pour les théories équationnelles suivantes : ACh, ACUNh, AGh, $\mathsf{ACUN}\{.\}_{.}$, $\mathsf{AG}\{.\}_{.}$, $\mathsf{ACUN}\{.\}_{.}$ avec chiffrement commutatif et $\mathsf{AG}\{.\}_{.}$ avec chiffrement commutatif.

Intrus passif : Dans le chapitre 4, nous avons proposé une approche originale pour redémontrer le problème de déduction de l'intrus dans le modèle standard de Dolev-Yao étendu, en utilisant l'approche par localité introduite par McAllester. Puis dans le chapitre suivant, nous avons considéré le problème de déduction de l'intrus en présence de différentes théories équationnelles modélisant la propriété d'homomorphisme, plus particulièrement les théories équationnelles associatives commutatives, du « ou exclusif » et des groupes abéliens en présence d'un symbole homomorphique. Cette propriété assure une réelle interaction entre deux symboles. Nous avons commencé par étudier le cas de l'intrus passif en présence de ces théories équationnelles. Nous avons essayé d'étendre les études existantes pour les théories du « ou exclusif » et des groupes abéliens. Après de nombreuses tentatives infructueuses dans cette direction, nous avons développé une nouvelle approche basée sur des transformations de preuves pour obtenir la décidabilité du problème de déduction de l'intrus en présence de ces théories équationnelles. Ensuite, nous nous sommes attachés aux théories équationnelles d'un chiffrement distributif sur le « ou exclusif » et les groupes abéliens, lorsque le chiffrement est commutatif ou non ($\mathsf{ACUN}\{.\}_.\ \mathsf{AG}\{.\}_.$). Dans ces théories équationnelles, l'interaction entre les symboles de chiffrement et l'opérateur associatif, commutatif est renforcée. Ces théories équationnelles prennent en compte autant de symboles distributifs sur cet opérateur qu'il y a de clefs de chiffrement différentes. De plus, dans ce cas contrairement au cas où il n'y a qu'un symbole homomorphique, les symboles homomorphiques peuvent être détruits par un déchiffrement. Toutes ces nouvelles propriétés algébriques nous ont amenés à étudier plus précisément la forme des preuves, pour arriver à obtenir la décidabilité du problème de déduction de l'intrus pour ces deux théories équationnelles. Nous avons pour chacune d'entre elles montré une borne de complexité inférieure dans le cas de preuves binaires. Enfin nous montrons, dans le chapitre 6, que l'unification et le problème de déduction de l'intrus sont deux problèmes indépendants. Après avoir résolu le cas passif, nous nous sommes penchés sur le cas de l'intrus actif pour la théorie équationnelle d'un symbole homomorphique et du « ou exclusif ». Nous pourrions poursuivre ces travaux pour un intrus passif en essayant d'obtenir de nouveaux résultats pour de nouvelles théories équationnelles. Plus généralement nous souhaiterions démontrer un résultat de combinaison pour le problème de déduction de l'intrus concernant des théories équationnelles disjointes

Intrus actif : Dans le chapitre 7, nous avons introduit une caractérisation des protocoles déterministes. Cette classe de protocoles déterministes définit les protocoles qui en fonction des messages échangés précédemment possèdent une seule manière de continuer le protocole. Puis nous avons développé une procédure de décision pour de tels protocoles. Tout protocole déterministe se transforme donc en un système de contraintes « bien défini ». Dans le chapitre 8, nous avons présenté un nouvel algorithme effectif d'unification modulo ACUNh basé sur les automates. La résolution de systèmes d'équations diophantiennes quadratiques est un problème indécidable en général, dans le chapitre 9, nous avons proposé une méthode de résolution de systèmes particuliers d'équations diophantiennes quadratiques : les système d'équations « dépendant ». Enfin dans le chapitre 10, en utilisant tous ces résultats préliminaires, nous décrivons en détail notre procédure de décision non-déterministe pour l'intrus actif pour un nombre borné de sessions pour le problème de sécurité en présence du « ou exclusif » et d'un symbole homomorphique (ACUNh). Cette procédure consiste à résoudre un système de contraintes bien défini, elle se décompose en quatre étapes : premièrement à partir du système de contraintes bien défini initial nous devinons un système de contraintes une étape bien défini équivalent. Deuxièmement, nous utilisons l'algorithme d'unification modulo ACUNh pour obtenir un nouveau système de contraintes où les contraintes ne sont plus que des applications des constructeurs des symboles homomorphique et du « ou exclusif ». Troisièmement, nous abstrayons ce système en un système d'équations « dépendant ». Quatrièmement, nous utilisons le résultat du chapitre 9 pour résoudre ce système d'équations. Nous obtenons ainsi une procédure de décision pour le problème de sécurité dans le cas actif en présence de la théorie équationnelle ACUNh. Remarquons que cette procédure peut être appliquée pour résoudre le même problème pour la théorie équationnelle de groupes abéliens et du « ou exclusif ». Nous pourrions sûrement améliorer notre procédure non-déterministe en effectuant une étude plus précise de sa complexité. Après avoir étudié la théorie ACUNh, une extension naturelle de ce travail serait, comme nous l'avons fait dans le cas passif, de regarder la théorie équationnelle avec chiffrement distributif sur le « ou exclusif » pour un chiffrement commutatif ou non.

Perspectives : Il serait intéressant de vérifier les protocoles cryptographiques en considérant de nouvelles théories équationnelles présentes dans les spécifications de nouveaux protocoles. Par exemple, nous pourrions regarder les propriétés algébriques des protocoles utilisant les courbes elliptiques comme fonction de chiffrement. D'autre part, des propriétés algébriques utilisées dans des protocoles issus du monde industriel ne sont pas encore formellement modélisées et étudiées. Une étude précise de certains de ces protocoles permettrait de dégager de nouvelles propriétés, et ensuite proposer une modélisation et une vérification formelle de ces protocoles. Enfin, ces nouveaux résultats obtenus à propos de théories équationnelles avec homomorphisme ne sont pas encore pris en compte par les outils de vérification de protocoles cryptographiques actuels. Le développement d'un module supplémentaire dans un outil de vérification de protocole existant semble être une perspective naturelle de ce travail théorique.

De nos jours, le réseau Internet prend une place de plus en plus importante et offre de nombreux services en ligne : commerce électronique, services bancaires, bibliothèques en ligne, vote électronique. Ces « Web Services » sont souvent plus complexes à analyser que les protocoles cryptographiques. En effet, plusieurs intervenants, souvent hétérogènes agissent dynamiquement dans la réalisation de ces services. De nombreux modèles voient le jour pour formaliser, analyser et vérifier ces Web Services. Il me semble pertinent d'essayer d'analyser les Web Services pour arriver à dégager les propriétés qu'ils utilisent. Ensuite, en se basant sur cette analyse, il serait intéressant de proposer une formalisation adaptée des Web Services pour ensuite les vérifier, en prenant en compte leur aspect dynamique et interactif ainsi que les propriétés algébriques sous-jacentes. Il semble aussi fort intéressant d'étudier de nouvelles propriétés comme les propriétés liées aux marqueurs temporels, au protocole de groupes ou au vote électronique de plus en plus usités.

Bibliographie

[80299] IEEE 802.11 Local and Metropolitan Area Networks : Wireless LAN Medium Acess Control (MAC) and Physical (PHY) Specifications, 1999.

[ABB+02] A. Armando, D. Basin, M. Bouallagui, Y. Chevalier, L. Compagna, S. Mödersheim, M. Rusinowitch, M. Turuani, L. Viganò, and L. Vigneron. The AVISS Security Protocol Analysis Tool. In *Proceedings of CAV'02*, LNCS 2404, pages 349–354. Springer-Verlag, 2002. URL of the AVISS and AVISPA projects : `www.avispa-project.org`.

[ABB+05] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. H. Drielsma, P.-C. Heám, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganò, and L. Vigneron. The avispa tool for the automated validation of internet security protocols and applications. In *Proceedings of CAV'2005*, LNCS 3576, pages 281–285. Springer-Verlag, 2005.

[Abd87] H. Abdulrab. Equations en mots. Technical Report R.G.7.87, Greco de Programmation, Université de Bordeaux 1, 1987.

[AC02] R. Amadio and W. Charatonik. On name generation and set-based analysis in the Dolev-Yao model. In *Proc. International Conference on Concurrency Theory (CONCUR'02)*, volume 2421 of *Lecture Notes in Computer Science*, pages 499–514, Brno, Czech Republic, 2002. Springer-Verlag.

[AC04a] M. Abadi and V. Cortier. Deciding knowledge in security protocols under equational theories. In *Proc. 31st International Colloquium on Automata, Languages, and Programming (ICALP'04)*, volume 3142 of *Lecture Notes in Computer Science*, pages 46–58, Turku, Finland, 2004. Springer-Verlag.

[AC04b] A. Armando and L. Compagna. Satmc : a sat-based model checker for security protocols. In *Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA'04)*, Lecture Notes in Artificial Intelligence 3229, pages 730–733, Lisbon, Portugal, September 2004. Springer-Verlag.

[AC05a] M. Abadi and V. Cortier. Deciding knowledge in security protocols under (many more) equational theories. In *CSFW '05 : Proceedings of the 18th IEEE Computer Security Foundations Workshop (CSFW'05)*, pages 62–76, Washington, DC, USA, 2005. IEEE Computer Society.

[AC05b] A. Armando and L. Compagna. An optimized intruder model for sat-based model-checking of security protocols. In A. Armando and L. Viganò, editors, *Electronic Notes in Theoretical Computer Science*, volume 125, pages 91–108. Elsevier Science

Publishers, March 2005. Presented to the IJCAR04 Workshop ARSPA, available at `http://www.avispa-project.org`.

[AG99] M. Abadi and A. D. Gordon. A calculus for cryptographic protocols : The spi calculus. *Information and Computation*, 148(1) :1–70, January 1999.

[ALV02] R. Amadio, D. Lugiez, and V. Vanackère. On the symbolic reduction of processes with cryptographic functions. *Theoretical Computer Science*, 290(1) :695–740, 2002.

[AN95] R. Anderson and R. Needham. Programming Satan's computer. In *Computer Science Today : Recent Trends and Developments*, volume 1000 of *Lecture Notes in Computer Science*, pages 426–440. Springer-Verlag, 1995.

[AP90] H. Abdulrab and J.-P. Pécuchet. Solving word equations., 1990.

[AR00] M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). In *Proc. 1st IFIP International Conference on Theoretical Computer Science (IFIP–TCS)*, volume 1872 of *Lecture Notes in Computer Science*, pages 3–22. Springer-Verlag, 2000.

[AST00] G. Ateniese, M. Steiner, and G. Tsudik. New multiparty authentication services and key agreement protocols. *IEEE Journal of Selected Areas in Communications*, 18(4) :628–639, 2000.

[Baa93] F. Baader. Unification in commutative theories, Hilbert's basis theorem and Gröbner bases. *Journal of the ACM*, 40(3) :477–503, 1993.

[BAN89] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. In *Proc. 12th ACM Symposium on Operating System Principles (SOSP'89)*, pages 1–13, Litchfield Park, Arizona, USA, 1989. ACM Press.

[Bau05] M. Baudet. Deciding security of protocols against off-line guessing attacks. In *Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS'05)*, pages 16–25, Alexandria, Virginia, USA, November 2005. ACM Press.

[Bau06] M. Baudet. *Sécurité des protocoles cryptographiques : aspects logiques et calculatoires.* PhD thesis, École Normale Supérieure de Cachan, Cachan, France, September 2006.

[BB03] M. Boreale and M. G. Buscemi. Symbolic analysis of crypto-protocols based on modular exponentiation. In *Proc. 28th International Symposium on Mathematical Foundations of Computer Science (MFCS'03)*, volume 2747 of *Lecture Notes in Computer Science*, pages 269–278, Bratislava, Slovak Republic, 2003. Springer-Verlag.

[BCK05] M. Baudet, V. Cortier, and S. Kremer. Computationally sound implementations of equational theories against passive adversaries. In L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, editors, *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP'05)*, volume 3580 of *Lecture Notes in Computer Science*, pages 652–663, Lisboa, Portugal, July 2005. Springer.

[BCLM03] S. Bistarelli, I. Cervesato, G. Lenzini, and F. Martinelli. Relating process algebras and multiset rewriting for security protocol analysis. In *Proc. 3rd Workshop on Issues in the Theory of Security (WITS'03)*, Warsaw, Poland, 2003.

[Bel96] S. M. Bellovin. Problem areas for the IP security protocols. In *Proc. 6th USENIX Security Symposium*, pages 1–16, San José, California, USA, 1996. Usenix.

[BEL04] L. Bozga, C. Ene, and Y. Lakhnech. A symbolic decision procedure for cryptographic protocols with time stamps. In *Proc. 15th International Conference on Concurrency*

Theory (CONCUR'04), Lecture Notes in Computer Science, London, England, 2004. Springer-Verlag. To appear.

[Ben87] J. Benaloh. Secret sharing homomorphisms : Keeping shares of a secret sharing. In *Proc. Advances in Cryptology (CRYPTO'86)*, volume 263 of *Lecture Notes in Computer Science*, pages 251–260, Santa Barbara, California, USA, 1987. Springer-Verlag.

[Ber06] V. Bernat. *Théories de l'intrus pour la vérification des protocoles cryptographiques.* PhD thesis, École Normale Supérieure de Cachan, Cachan, France, June 2006.

[BG00] A. Blumensath and E. Grädel. Automatic structures. In *Proc. 15th IEEE Symposium on Logic in Computer Science (LICS'00)*, pages 51–62, Santa Barbara, California, USA, 2000. IEEE Comp. Soc. Press.

[BGW01] N. Borisov, I. Goldberg, and D. Wagner. Intercepting mobile communications : The insecurity of 802.11. In *Proc. 7th Annual International Conference on Mobile Computing and Networking (MOBICOM'01)*, pages 180–188, Rome, Italy, 2001. ACM Press.

[BHK05] Y. Boichut, P.-C. Héam, and O. Kouchnarenko. Automatic verification of security protocols using approximations. Research Report RR2005-01, LIFC - Laboratoire d'Informatique de l'Université de Franche Comté, January 2005.

[BKV06] Y. Boichut, N. Kosmatov, and L. Vigneron. Validation of prouvé protocols using the automatic tool TA4SP. In *Proc. of the Third Taiwanese-French Conference on Information Technology (TFIT'06)*, pages 467–480, Nancy, France, March 2006.

[Bla01] B. Blanchet. An efficient cryptographic protocol verifier based on prolog rules. In *Proc. 14th Computer Security Foundations Workshop (CSFW'01)*, pages 82–96, Cape Breton, Canada, 2001. IEEE Comp. Soc. Press.

[Bla04] B. Blanchet. *Cryptographic Protocol Verifier User Manual*, 2004.

[BLP03] L. Bozga, Y. Lakhnech, and M. Perin. HERMES : An Automatic Tool for Verification of Secrecy in Security Protocols. In *Computer Aided Verification*, 2003.

[BLP06] L. Bozga, Y. Lakhnech, and M. Périn. Pattern-based abstraction for verifying secrecy in protocols. *International Journal on Software Tools for Technology Transfer (STTT'06)*, 8 :57–76, Feb 2006.

[BMV03] D. Basin, S. Mödersheim, and L. Viganò. An On-The-Fly Model-Checker for Security Protocol Analysis. In E. Snekkenes and D. Gollmann, editors, *Proceedings of ESORICS'03*, volume 2808 of *Lecture Notes in Computer Science*, pages 253–270. Springer-Verlag, Heidelberg, 2003.

[BMV05a] D. Basin, S. Mödersheim, and L. Viganò. Algebraic intruder deductions. In G. Sutcliffe and A. Voronkov, editors, *LPAR 2005*, volume 3835 of *LNAI*, pages 549–564. Springer-Verlag, December 2005.

[BMV05b] D. Basin, S. Mödersheim, and L. Viganò. Ofmc : A symbolic model checker for security protocols. *International Journal of Information Security*, 4(3) :181–208, June 2005. Published online December 2004.

[BO97] J. Bull and D. J. Otway. The authentication protocol. Technical Report DRA/CIS3/PROJ/CORBA/SC/1/CSM/436-04/03, Defence Research Agency, 1997.

[Bor01] M. Boreale. Symbolic trace analysis of cryptographic protocols. In *Proc. 28th International Colloquium on Automata, Languages, and Programming (ICALP'01)*, volume 2076 of *Lecture Notes in Computer Science*, pages 667–681, Crete, Greece, 2001. Springer-Verlag.

[BP03] B. Blanchet and A. Podelski. Verification of cryptographic protocols : Tagging enforces termination. In *Proc. 6th International Conference on Foundations of Software Science and Computation Structures (FOSSACS'03)*, volume 2620 of *Lecture Notes in Computer Science*, pages 136–152, Warsaw, Poland, 2003. Springer-Verlag.

[BP04] M. Backes and B. Pfitzmann. Symmetric encryption in a simulatable dolev-yao style cryptographic library. *17th IEEE Computer Security Foundations Workshop (CSFW'04)*, page 204, 2004.

[BP05] M. Backes and B. Pfitzmann. Relating symbolic and cryptographic secrecy. *IEEE Transactions on Dependable and Secure Computing 2/2*, pages 109–123, 2005.

[BPW03] M. Backes, B. Pfitzmann, and M. Waidner. A composable cryptographic library with nested operations. In *ACM Conference on Computer and Communications Security*. ACM SIGSAC, 2003.

[BS96] F. Baader and K. U. Schulz. Unification in the union of disjoint equational theories : Combining decision procedures. *J. Symbolic Computation*, 21 :211–243, 1996.

[BS01] F. Baader and W. Snyder. Unification theory. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 8, pages 445–532. Elsevier Science, 2001.

[Buc01] J. Buchmann. *Introduction to cryptography*. Undergraduate texts in mathematics. Springer-Verlag, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 2001.

[Cau92] D. Caucal. On the regular structure of prefix rewriting. *Theoretical Computer Science*, 106(1) :61–86, 1992.

[CB83] J. Corbin and M. Bidoit. A rehabilitation of robinson's unification algorithm. In *Proc. IFIP '83*, pages 909–914. North-Holland, 1983.

[CC01] H. Comon and V. Cortier. Tree automata with one memory, set constraints and cryptographic protocols. Research Report LSV-01-13, Laboratoire Spécification and Vérification, ENS de Cachan, France, 2001.

[CDL06] V. Cortier, S. Delaune, and P. Lafourcade. A survey of algebraic properties used in cryptographic protocols. *Journal of Computer Security*, 14(1) :1–43, 2006.

[CDM$^+$00] Cervesato, Durgin, Mitchell, Lincoln, and Scedrov. Relating strands and multiset rewriting for security protocol analysis. In *PCSFW : Proceedings of The 13th Computer Security Foundations Workshop*. IEEE Computer Society Press, 2000.

[CDS94] R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Proc. 14th Annual International Cryptology Conference (CRYPTO'94)*, volume 963 of *Lecture Notes in Computer Science*, pages 174–187, Santa Barbara, California, USA, 1994. Springer-Verlag.

[CES06] R. Corin, S. Etalle, and A. Saptawijaya. A logic for constraint-based security protocol analysis. In *IEEE Symposium on Security and Privacy*, 2006.

[CF85] J. Cohen and M. Fischer. A robust and verifiable cryptographically secure election scheme. In *Proc. 26th Annual Symposium on Foundations of Computer Science (FOCS'85)*, pages 372–382, Portland, Oregon, USA, 1985. IEEE Comp. Soc. Press.

[CGS97] R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. In *Proc. International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT'97)*, volume 1233 of *Lecture Notes in Computer Science*, pages 103–118, Konstanz, Germany, 1997. Springer-Verlag.

[Che03] Y. Chevalier. *Résolution de problèmes d'accessibilité pour la compilation et la validation de protocoles cryptographiques.* PhD thesis, Université Henri Poincaré, Nancy, France, December 2003.

[CJ97] J. Clark and J. Jacob. A survey of authentication protocol literature. `http://www.cs.york.ac.uk/~jac/papers/drareviewps.ps`, 1997.

[CKR+03a] Y. Chevalier, R. Küsters, M. Rusinowitch, M. Turuani, and L. Vigneron. Deciding the security of protocols with Diffie-Hellman exponentiation and product in exponents. In *Proc. 23rd Conference on Foundations of Software Technology and Theoretical Computer Science (FST&TCS'03)*, volume 2914 of *Lecture Notes in Computer Science*, pages 124–135, Mumbai, India, 2003. Springer-Verlag.

[CKR+03b] Y. Chevalier, R. Küsters, M. Rusinowitch, M. Turuani, and L. Vigneron. Extending the Dolev-Yao intruder for analyzing an unbounded number of sessions. In *Proc. 17th International Workshop in Computer Science Logic (CSL'03)*, volume 2803 of *Lecture Notes in Computer Science*, pages 128–141, Vienna, Austria, 2003. Springer-Verlag.

[CKRT03] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. An NP decision procedure for protocol insecurity with XOR. In *Proc. of 18th Annual IEEE Symposium on Logic in Computer Science (LICS'03)*, pages 261–270, Ottawa, Canada, 2003. IEEE Comp. Soc. Press.

[CKRT04] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. Deciding the security of protocols with commuting public key encryption. In *Proc. Workshop on Automated Reasoning for Security Protocol Analysis (ARSPA'04)*, pages 53–63, Cork Ireland, 2004.

[CL04] H. Comon-Lundh. Intruder theories (ongoing work). In *Proc. 7th International Conference on Foundations of Software Science and Computation Structures (FOSSACS'04)*, volume 2987 of *Lecture Notes in Computer Science*, pages 1–4, Barcelona, Spain, 2004. Springer-Verlag.

[CLC03] H. Comon-Lundh and V. Cortier. New decidability results for fragments of first-order logic and application to cryptographic protocols. In *Proc. 14th International Conference on Rewriting Techniques and Applications (RTA'03)*, volume 2706 of *Lecture Notes in Computer Science*, pages 148–164, Valencia, Spain, 2003. Springer-Verlag.

[CLS03] H. Comon-Lundh and V. Shmatikov. Intruder deductions, constraint solving and insecurity decision in presence of exclusive or. In *Proc. of 18th Annual IEEE Symposium on Logic in Computer Science (LICS'03)*, pages 271–280, Ottawa, Canada, 2003. IEEE Comp. Soc. Press.

[CLT03] H. Comon-Lundh and R. Treinen. Easy intruder deductions. In N. Dershowitz, editor, *Verification : Theory & Practice, Essays Dedicated to Zohar Manna on the Occasion of His 64th Birthday*, volume 2772 of *Lecture Notes in Computer Science*, pages 225–242. Springer-Verlag, 2003.

[CM96] E. Contejean and C. Marché. CiME : Completion Modulo E. In H. Ganzinger, editor, *7th International Conference on Rewriting Techniques and Applications*, volume 1103 of *Lecture Notes in Computer Science*, pages 416–419, New Brunswick, NJ, USA, July 1996. Springer-Verlag.

[CMSS03] R. Chadha, J. C. Mitchell, A. Scedrov, and V. Shmatikov. Contract signing, optimism, and advantage. In R. Amadio and D. Lugiez, editors, *CONCUR 2003 — Concurrency*

Theory, volume 2761 of *Lecture Notes in Computer Science*, pages 366–382, Marseille, France, September 2003. Springer-Verlag.

[CR05] Y. Chevalier and M. Rusinowitch. Combining intruder theories. In L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, editors, *ICALP*, volume 3580 of *Lecture Notes in Computer Science*, pages 639–651. Springer, 2005.

[Cre] C. Cremers. The scyther tool : Automatic verification of security protocols.

[CRZ05] V. Cortier, M. Rusinowitch, and E. Zalinescu. A resolution strategy for verifying cryptographic protocols with cbc encryption and blind signatures. In *PPDP*, pages 12–22, 2005.

[CV02] Y. Chevalier and L. Vigneron. Automated unbounded verification of security protocols. In *Proc. 14th International Conference on Computer Aided Verification (CAV'02)*, volume 2404 of *Lecture Notes in Computer Science*, pages 324–337, Copenhagen, Denmark, 2002. Springer-Verlag.

[Dav73] M. Davis. Hilbert's Tenth Problem is unsolvable. *The American Mathematical Monthly*, 80(3) :233–269, March 1973. Reprinted with corrections in the Dover edition of Davis [1958].

[DCJW04] Y. Deswarte, F. Cuppens, S. Jajodia, and L. Wang, editors. *Security and Protection in Information Processing Systems, IFIP 18th WorldComputer Congress, TC11 19th International Information Security Conference, 22-27 August 2004, Toulouse, France.* Kluwer, 2004.

[DDHY92] D. L. Dill, A. J. Drexler, A. J. Hu, and C. H. Yang. Protocol verification as a hardware design aid. In *1992 IEEE International Conference on Computer Design : VLSI in Computers and Processors*, pages 522–525. IEEE Computer Society, 1992. Cambridge, MA, October 11-14.

[Del06a] S. Delaune. Easy intruder deduction problems with homomorphisms. *Information Processing Letters*, 97(6) :213–218, 2006.

[Del06b] S. Delaune. An undecidability result for AGh. Research Report LSV-06-02, Laboratoire Spécification et Vérification, ENS Cachan, France, February 2006. 9 pages.

[DG04] G. Delzanno and P. Ganty. Automatic verification of time sensitive cryptographic protocols. In *Proc. 10th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'04)*, volume 2988 of *Lecture Notes in Computer Science*, pages 342–356, Barcelona, Spain, 2004. Springer-Verlag.

[DH76] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Society*, 22(6) :644–654, 1976.

[DH77] W. Diffie and M. Hellman. Exhaustive cryptanalysis of the NBS Data Encryption Standard. *IEEE Computer*, 10 :74–84, 1977.

[Dic13] L. Dickson. Finiteness of the odd perfect and primitive abundant numbers with n prime factors. *American Journal Mathematical Society*, 35 :413–422, 1913.

[DJ04] S. Delaune and F. Jacquemard. A decision procedure for the verification of security protocols with explicit destructors. In V. Atluri, B. Pfitzmann, and P. McDaniel, editors, *Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS'04)*, pages 278–287, Washington, D.C., USA, October 2004. ACM Press.

[DK02] H. Delfs and H. Knebl. *Introduction to Cryptography.* Springer-Verlag, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 2002.

[DLLT05] S. Delaune, P. Lafourcade, D. Lugiez, and R. Treinen. Symbolic protocol analysis in presence of a homomorphism operator and *exclusive or*. Research Report LSV-05-20, Laboratoire Spécification et Vérification, ENS Cachan, France, November 2005. 44 pages.

[DLMS99] N. Durgin, P. Lincoln, J. Mitchell, and A. Scedrov. Undecidability of bounded security protocols. In *Proc. Workshop on Formal Methods and Security Protocols (FMSP'99)*, Trento, Italy, 1999.

[DMS04] R. Dingledine, N. Mathewson, and P. Syverson. Tor : The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[DR01] J. Daemen and V. Rijmen. Algorithm alley : Rijndael : The Advanced Encryption Standard. *Dr. Dobb's Journal of Software Tools*, 26(3) :137–139, March 2001.

[DR02] J. Daemen and V. Rijmen. *The design of Rijndael : AES — the Advanced Encryption Standard*. Springer-Verlag, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 2002.

[DS81] D. E. Denning and G. M. Sacco. Timestamps in key distribution protocols. *Communications of the ACM*, 24(8) :533–536, 1981.

[DW83] M. D. Davis and E. J. Weyuker. *Computability, complexity and languages*, chapter 7, pages 128–132. Computer Science and Applied Mathematics. Academic Press, 1983.

[DY81] D. Dolev and A. Yao. On the security of public key protocols. In *Proc. of the 22nd Symp. on Foundations of ComputerScience*, pages 350–357, Nashville, Tennessee, USA, 1981. IEEE Computer Society Press.

[DY83] D. Dolev and A. Yao. On the security of public-key protocols. In *Transactions on Information Theory*, volume 29, pages 198–208. IEEE Computer Society Press, March 1983.

[EG83] S. Even and O. Goldreich. On the security of multi-party ping-pong protocols. In *Proc. 24th Annual Symposium on Foundations of Computer Science (FOCS'83)*, pages 34–39, Tucson, Arizona, USA, 1983. IEEE Comp. Soc. Press.

[EGS86] S. Even, O. Goldreich, and A. Shamir. On the security of ping-pong protocols when implemented using the RSA. In *Proc. Advances in Cryptology (CRYPTO'85)*, volume 218 of *Lecture Notes in Computer Science*, pages 58–72, Santa Barbara, California, USA, 1986. Springer-Verlag.

[El 85] T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Proc. Advances in Cryptology (CRYPTO'84)*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18, Santa Barbara, California, USA, 1985. Springer-Verlag.

[ES00] N. Evans and S. Schneider. Analysing time dependent security properties in CSP using PVS. In *Proc. 6th European Symposium on Research in Computer Security (ESORICS'00)*, volume 1895 of *Lecture Notes in Computer Science*, pages 222–237, Toulouse, France, 2000. Springer-Verlag.

[Fag84] F. Fages. Associative-commutative unification. In *Proceedings 7th International Conference on Automated Deduction*, volume 170 of *Lecture Notes in Artificial Intelligence*, pages 194–208. Springer-Verlag, 1984.

[FHG98] F. J. T. Fabrega, J. Herzog, and J. D. Guttman. Strand spaces : Why is a security protocol correct ? In *1998 IEEE Symposium on Security and Privacy.* IEEE Computer Society Press, May 1998.

[FPS01] P.-A. Fouque, G. Poupard, and J. Stern. Sharing decryption in the context of voting or lotteries. In *Proc. 4th International Conference on Financial Cryptography (FC'00)*, volume 1962 of *Lecture Notes in Computer Science*, pages 90–104, Anguilla, British West Indies, 2001. Springer-Verlag.

[GK00] T. Genet and F. Klay. Rewriting for cryptographic protocol verification (extended version). In *Proc. of the 17th International Conference on Automated Deduction (CAD'00)*, volume 1831 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, January 2000.

[GL00] J. Goubault-Larrecq. A method for automatic cryptographic protocol verification. In *Proc. of the 15 IPDPS 2000 Workshops*, volume 1800 of *Lecture Notes in Computer Science*, pages 977–984, Cancun, Mexico, May 2000. Springer Verlag.

[GLRV04] J. Goubault-Larrecq, M. Roger, and K. N. Verma. Abstraction and resolution modulo AC : How to verify Diffie-Hellman-like protocols automatically. *Journal of Logic and Algebraic Programming*, 2004. To appear.

[GM84] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2) :270–299, 1984.

[GNW00] Q. Guo, P. Narendran, and D. A. Wolfram. Complexity of nilpotent unification and matching problems. *Information and Computation*, 162(1-2) :3–23, 2000.

[Gol99] D. Gollmann. What is authentication ? In B. Christianson, B. Crispo, J. A. Malcolm, and M. Roe, editors, *Security Protocols Workshop*, volume 1796 of *Lecture Notes in Computer Science*, pages 65–72. Springer, 1999.

[Gol00] D. Gollmann. On the verification of cryptographic protocols. Presentation at Karlstad University, 11 February 2000.

[Gon89] L. Gong. Using one-way functions for authentication. *SIGCOMM Computer Communication*, 19(5) :8–11, 1989.

[Gou05] J. Goubault-Larrecq. Deciding $\mathcal{H}_1$ by resolution. *Information Processing Letters*, 95(3) :401–408, August 2005.

[H$^+$00] D. R. Hankerson et al. *Coding theory and cryptography : the essentials*, volume 234 of *Monographs and textbooks in pure and applied mathematics*. Marcel Dekker, New York, NY, USA, second edition, 2000.

[Her30] J. Herbrand. *Recherches sur la Théorie de la Démonstration*. PhD thesis, University of Paris, 1930.

[HO04] J. Y. Halpern and K. R. O'Neill. Anonymity and information hiding in multiagent systems. *Journal of Computer Security*, 2004.

[Hoa85] C. A. R. Hoare. *Communicating sequential processes*. Prentice Hall, 1985. ISBN 0 13 153271 5 (Hard), 0 13 153289 8 (Pbk).

[HS04] D. Hughes and V. Shmatikov. Information hiding, anonymity and privacy : A modular approach. *Journal of Computer Security*, 12(1) :3–36, 2004.

[HS06] M. Hussain and D. Seret. A comparative study of security protocols validation tools : Hermes vs. avispa. In *The 8th International Conference Advanced Communication Technology, ICACT'06*, volume 1, pages 303– 308, 2006.

[Hul80] J.-M. Hullot. Canonical forms and unification. In W. Bibel and R. Kowalski, editors, *Proceedings of the 5th Conference on Automated Deduction*, volume 87 of *Lecture Notes in Computer Science*, pages 318–334, Les Arcs, France, July 1980. springer.

[Hüt02] H. Hüttel. Deciding framed bisimulation. In *Proc. 4th International Workshop on Verification of Infinite-State Systems (INFINITY'02)*, pages 1–20, Brno, Czech Republic, 2002.

[Jac] F. Jacquemard. Security protocols open repository. Available at `http://www.lsv.ens-cachan.fr/spore/index.html`.

[Jaf90] J. Jaffar. Minimal and complete word unification. *Journal of the ACM*, 37(1) :47–85, 1990.

[JP02] M. Jakobsson and D. Pointcheval. Mutual authentication for low-power mobile devices. In *Proc. 5th International Conference on Financial Cryptography (FC'01)*, volume 2339 of *Lecture Notes in Computer Science*, pages 178–195, Grand Cayman, British West Indies, 2002. Springer-Verlag.

[JRV00] F. Jacquemard, M. Rusinowitch, and L. Vigneron. Compiling and verifying security protocols. In *Proc. of 7th International Conference on Logic for Programming and Automated Reasoning (LPAR'00)*, volume 1955 of *Lecture Notes in Computer Science*, pages 131–160, Reunion Island, France, 2000. Springer-Verlag.

[Kir89] C. Kirchner. From unification in combination of equational theories to a new ac-unification algorithm. In H. Ait-Kaci and M. Nivat, editors, *Resolution of Equations in Algebraic Structures (Volume II) : Rewriting Techniques*, pages 171–210. Academic Press, London, 1989.

[KKS87] E. Kaltofen, M. S. Krishnamoorthy, and B. D. Saunders. Fast parallel computation of hermite and smith forms of polynomial matrices. *SIAM J. Algebraic Discrete Methods*, 8(4) :683–690, 1987.

[KKW06] D. Kähler, R. Küsters, and T. Wilke. A dolev-yao-based definition of abuse-free protocols. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP'06)*, Lecture Notes in Computer Science, pages 95–106, Venise, Italy, jul 2006. Springer.

[KNW03] D. Kapur, P. Narendran, and L. Wang. An E-unification algorithm for analyzing protocols that use modular exponentiation. In *Proc. 14th International Conference on Rewriting Techniques and Applications (RTA'03)*, volume 2706 of *Lecture Notes in Computer Science*, pages 165–179, Valencia, Spain, 2003. Springer-Verlag.

[Kob85] N. Koblitz. *Introduction to Elliptic Curves and Modular Forms*. Springer-Verlag, 1985.

[Kob00] N. Koblitz. *Towards a quarter-century of public key cryptography*. Kluwer Academic Press, Norwell, MA, USA, and Dordrecht, The Netherlands, 2000. A special issue of Designs, codes, and cryptography, an international journal, volume 19, no. 2/3 (2000).

[KR02] S. Kremer and J.-F. Raskin. Game analysis of abuse-free contract signing. In *Proceedings of the 15th IEEE Computer Security Foundations Workshop (CSFW'02)*, pages 206–220, Cape Breton, Nova Scotia, Canada, June 2002. IEEE Computer Society Press.

[KR05a] S. Kremer and M. D. Ryan. Analysing the vulnerability of protocols to produce known-pair and chosen-text attacks. In R. Focardi and G. Zavattaro, editors, *Proceedings of the 2nd International Workshop on Security Issues in Coordination Models, Languages and Systems (SecCo'04)*, volume 128 of *Electronic Notes in Theoretical Computer Science*, pages 84–107, London, UK, May 2005. Elsevier Science Publishers.

[KR05b] S. Kremer and M. D. Ryan. Analysis of an electronic voting protocol in the applied pi-calculus. In M. Sagiv, editor, *Programming Languages and Systems — Proceedings of the 14th European Symposium on Programming (ESOP'05)*, volume 3444 of *Lecture Notes in Computer Science*, pages 186–200, Edinburgh, U.K., April 2005. Springer.

[Kwo03] T. Kwon. Summary of AMP (authentication and key agreement via memorable passwords). Draft Document, August 2003.

[Lau04] P. Laud. Encryption in automatic analyses for confidentiality against active adversaries. In *Proc. of 2004 IEEE Symp. on Security and Privacy, S&P 2004 (Berkeley, CA, USA, May 2004)*, pages 71–85. IEEE CS Press, Los Alamitos, CA, 2004.

[Lau05] P. Laud. Secrecy types for a simulatable cryptographic library. In *the 12th ACM Conference on Computer and Communications Security*, volume 11, pages 26–35, November 2005.

[LBB84] D. Lankford, G. Butler, and B. Brady. Abelian group unification algorithms for elementary terms. In W. W. Bledsoe and D. W. Loveland, editors, *Contemporary Mathematics : Automated Theorem Proving - After 25 Years*, pages 193–200. American Mathematical Society, Providence, RI, 1984.

[LLT04] P. Lafourcade, D. Lugiez, and R. Treinen. Intruder deduction for AC-like equational theories with homomorphisms. Research Report LSV-04-16, Laboratoire Spécification and Vérification, ENS de Cachan, France, November 2004. Available at `http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rapports-year-2004-lis%t.php`.

[LLT05] P. Lafourcade, D. Lugiez, and R. Treinen. Intruder deduction for AC-like equational theories with homomorphisms. In J. Giesl, editor, *Proceedings of the 16th International Conference on Rewriting Techniques and Applications (RTA'05)*, volume 3467 of *Lecture Notes in Computer Science*, pages 308–322, Nara, Japan, April 2005. Springer-Verlag.

[LM04] C. Lynch and C. Meadows. On the relative soundness of the free algebra model for public key encryption. In *Proc. 4th Workshop on Issues in the Theory of Security (WITS'04)*, pages 81–95, Barcelona, Spain, 2004.

[Low95] G. Lowe. An attack on the Needham-Schroeder public key authentication protocol. *Information Processing Letters*, 56(3) :131–133, November 1995.

[Low96] G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *Proc. 2nd International Workshop on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'96)*, volume 1055 of *Lecture Notes in Computer Science*, pages 147–166, Berlin, Germany, 1996. Springer-Verlag.

[Low97a] G. Lowe. Casper : A compiler for the analysis of security protocols. In *Proc. 10th Computer Security Foundations Workshop (CSFW'97)*, pages 18–30, Rockport, Massachusetts, USA, 1997. IEEE Comp. Soc. Press.

[Low97b] G. Lowe. A hierarchy of authentication specifications. In *Proc. IEEE Computer Security Foundations Workshop*, 1997.

[Low98] G. Lowe. Towards a completeness result for model checking of security protocols. In *Proc. 11th Computer Security Foundations Workshop (CSFW'98)*, pages 96–106, Rockport, Massachusetts, USA, 1998. IEEE Comp. Soc. Press.

[LR97] G. Lowe and A. W. Roscoe. Using CSP to detect errors in the TMN protocol. *IEEE Transactions on Software Engineering*, 23(10) :659–669, 1997.

[LS75] M. Livesey and J. Siekmann. Termination and decidability results for string unification. Technical report memo csm-12, University of Essex, 1975.

[Ltd97] F. S. E. Ltd. Failures-divergence refinement : FDR 2 user manual. Technical report, Formal Systems (Europe) Ltd., http : //www.formal.demon.co.uk/, 1997.

[Mak77] G. Makanin. The problem of solvability of equations in a free semigroup. *Akad. Nauk. SSSR*, 233(2), 1977.

[Mat93] Y. V. Matiyasevich. *Hilbert's Tenth Problem*. MIT Press, Cambridge, Massachusetts, 1993.

[McA93] D. A. McAllester. Automatic recognition of tractability in inference relations. *Journal of the ACM*, 40(2) :284–303, April 1993.

[MD02] J. K. Millen and G. Denker. CAPSL and MuCAPSL. *Journal of Telecommunications and Information Technology*, 4 :16–27, 2002.

[Mea96] C. Meadows. Language generation and verification in the NRL protocol analyzer. In *Proc. 9th Computer Seurity Foundation Workshop (CSFW'96)*, pages 48–62, Kenmare, Ireland, 1996. IEEE Comp. Soc. Press.

[Mea00] C. Meadows. Extending formal cryptographic protocol analysis techniques for group protocols and low-level cryptographic primitives. In *Proc. 1st Workshop on Issues in the Theory of Security (WITS'00)*, pages 87–92, Geneva, Switzerland, 2000.

[MGK03] O. Markowitch, D. Gollmann, and S. Kremer. On fairness in exchange protocols. In P. J. Lee and C. H. Lim, editors, *Revised Papers of the 5th International Conference on Information Security and Cryptology (ICISC'02)*, volume 2587 of *Lecture Notes in Computer Science*, pages 451–464, Seoul, Korea, 2003. Springer.

[Mid94] A. Middeldorp. Completeness of combinations of conditional constructor systems. *Journal of Symbolic Computation*, 17(1) :3–21, January 1994.

[Mil85] V. S. Miller. Use of elliptic curves in cryptography. In H. C. Williams, editor, *Proc. CRYPTO 85*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426, 1985.

[Mil03] J. Millen. On the freedom of decryption. *Information Processing Letters*, 86(6) :329–333, 2003.

[MM82a] A. Martelli and U. Montanari. An efficient unification algorithm. *ACM Transactions on Programming Languages and Systems*, 4(2) :258–282, 1982.

[MM82b] E. W. Mayr and A. R. Meyer. The complexity of the word problems for commutative semigroups and polynomial ideals. *Advances in Mathematics*, 46(3) :305–329, 1982.

[MMS97a] J. C. Mitchell, M. Mitchell, and U. Stern. Automated analysis of cryptographic protocols using Murφ. In *Proc. 1997 IEEE Symposium on Security and Privacy*, pages 141–151, Oakland, California, USA, 1997. IEEE Comp. Soc. Press.

[MMS97b] J. C. Mitchell, M. Mitchell, and U. Stern. Automated analysis of cryptographic protocols using murphi. In *IEEE Symposium on Security and Privacy*, May 1997.

[Mol01] R. A. Mollin. *An Introduction to Cryptography*. The CRC Press series on discrete mathematics and its applications. Chapman and Hall/CRC, Boca Raton, FL, USA, 2001.

[Mon99] D. Monniaux. Abstracting cryptographic protocols with tree automata. In *Proc. of the 6th International Static Analysis Symposium (SAS'99)*, volume 1694 of *Lecture Notes in Computer Science*. Springer Verlag, 1999.

[MPS00] P. McKenzie, S. Patel, and R. Swaminathan. Password-authenticated key exchange based on RSA. In *Proc. 6th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'00)*, volume 1976 of *Lecture Notes in Computer Science*, pages 599–613, Kyoto, Japan, 2000. Springer-Verlag.

[MS01] J. Millen and V. Shmatikov. Constraint solving for bounded-process cryptographic protocol analysis. In *Proc. 8th ACM Conference on Computer and Communications Security (CCS'01)*. ACM Press, 2001.

[MS03] J. Millen and V. Shmatikov. Symbolic protocol analysis with products and Diffie-Hellman exponentiation. In *Proc. 16th Computer Security Foundation Workshop (CSFW'03)*, pages 47–62, Pacific Grove, California, USA, 2003. IEEE Comp. Soc. Press.

[MS05] J. Millen and V. Shmatikov. Symbolic protocol analysis with an abelian group operator or Diffie-Hellman exponentiation. *Journal of Computer Security*, 13(3) :515 – 564, 2005.

[MW04] D. Micciancio and B. Warinschi. Soundness of formal encryption in the presence of active adversaries. In M. Naor, editor, *Theory of Cryptography : First Theory of Cryptography Conference (TCC '04)*, volume 2951 of *Lecture Notes in Computer Science*, pages 133–151. Springer, 2004.

[Nar96] P. Narendran. Solving linear equations over polynomial semirings. In *Proc. of 11th Annual Symposium on Logic in Computer Science (LICS'96)*, pages 466–472, July 1996.

[Nat77] National Bureau of Standards. *FIPS Publication 46 : Announcing the Data Encryption Standard*, January 1977.

[Nat80] National Bureau of Standards. *FIPS Publication 81 : DES Modes of Operation*, December 1980.

[Nat88] National Bureau of Standards. *FIPS Publication 46-1 : Data Encryption Standard*, January 1988.

[Nat98] National Security Agency. *SKIPJACK and KEA algorithm secification,Version 2.0*, 1998.

[NNS02] F. Nielson, H. R. Nielson, and H. Seidl. Normalizable Horn clauses, strongly recognizable relations and Spi. In *9th Static Analysis Symposium (SAS)*. Springer Verlag LNCS 2477, 2002.

[NS78a] R. Needham and M. Schroeder. Using encryption for authentication in large networks of computers. *Communication of the ACM*, 21(12) :993–999, 1978.

[NS78b] R. Needham and M. Schroeder. Using encryption for authentification in large networks of computers. *Communications of the ACM*, 21(12) :993–999, 1978.

[NS97] D. Naccache and J. Stern. A new public-key cryptosystem. *Proc. International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT'97)*, 1233 :27–37, 1997.

[Nut90] W. Nutt. Unification in monoidal theories. In M. E. Stickel, editor, *10th International Conference on Automated Deduction*, volume 449 of *Lecture Notes in Artificial Intelligence*, pages 618–632, Kaiserslautern, Germany, July 1990. Springer-Verlag.

[OU98] T. Okamoto and S. Uchiyama. A new public-key cryptosystem as secure as factoring. In *Proc. International Conference on the Theory and Application of Cryptographic*

Techniques (EUROCRYPT'98), volume 1403, pages 308–318, Helsinki, Finland, 1998. Springer-Verlag. Lecture Notes in Computer Science.

[Pai99] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Proc. International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT'99)*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238, Prague, Czech Republic, 1999. Springer-Verlag.

[Pap94] C. H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[Pau01] L. C. Paulson. Relations between secrets : Two formal analyses of the yahalom protocol. *Journal of Computer Security*, 9(3) :197–216, 2001.

[Plo72] G. Plotkin. Building-in equational theories. *Machine Intelligence*, 7, 1972.

[PQ00] O. Pereira and J.-J. Quisquater. On the perfect encryption assumption. In *Proc. 1st Workshop on Issues in the Theory of Security (WITS'00)*, pages 42–45, Geneva, Switzerland, 2000.

[PQ01] O. Pereira and J.-J. Quisquater. A security analysis of the cliques protocols suites. In *Proc. 14th Computer Security Foundations Workshop (CSFW'01)*, pages 73–81, Cape Breton, Canada, 2001.

[PS82] C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization Algorithms and Complexity.* Prentice-Hall Inc., 1982.

[PW78] M. Paterson and M. Wegman. Linear unification. *Journal of Computer and System Sciences*, 17 :348–375, 1978.

[Rob65] J. A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12 :23–41, 1965.

[Ros94] A. Roscoe. Model-checking CSP. In A. Roscoe, editor, *A Classical Mind : Essays in Honour of C.A.R. Hoare*. Prentice-Hall, 1994.

[Ros97] A. W. Roscoe. *The Theory and Practice of Concurrency.* Prentice-Hall, 1997.

[RS98] P. Y. A. Ryan and S. A. Schneider. An attack on a recursive authentication protocol : A cautionary tale. *Information Processing Letters*, 65(1) :7–10, 1998.

[RS03a] R. Ramanujam and S. P. Suresh. A decidable subclass of unbounded security protocols. In *Proc. IFIP Workshop on Issues in the Theory of Security (WITS'03)*, pages 11–20, Warsaw, Poland, 2003.

[RS03b] R. Ramanujam and S. P. Suresh. Tagging makes secrecy decidable for unbounded nonces as well. In *Proc. 23rd Conference on Foundations of Software Technology and Theoretical Computer Science (FST&TCS'03)*, volume 2914 of *Lecture Notes in Computer Science*, pages 363–374, Mumbai, India, 2003. Springer-Verlag.

[RT01] M. Rusinowitch and M. Turuani. Protocol insecurity with finite number of sessions is NP-complete. In *Proc. 14th Computer Security Foundations Workshop (CSFW'01)*, pages 174–190, Cape Breton, Canada, 2001. IEEE Comp. Soc. Press.

[RT03] M. Rusinowitch and M. Turuani. Protocol insecurity with a finite number of sessions, composed keys is NP-complete. *Theoretical Computer Science*, 1-3(299) :451–475, 2003.

[SB88] M. E. Smid and D. K. Branstad. The Data Encryption Standard : Past and future. *Proc. IEEE*, 76(5) :550–559, 1988.

[Sch86] A. Schrijver. *Theory of Linear and Integer Programming.* Wiley, 1986.

[Sch96] B. Schneier. *Applied Cryptography Second Edition : protocols, algorithms, and source code in C.* J. Wiley & Sons, Inc., 1996.

[Sch98] S. Schneider. Verifying authentication protocols in CSP. *IEEE Transactions on Software Engineering*, 24(9) :741–758, 1998.

[Shm04] V. Shmatikov. Decidable analysis of cryptographic protocols with products and modular exponentiation. In *Proc. 13th European Symposium On Programming (ESOP'04)*, volume 2986 of *Lecture Notes in Computer Science*, pages 355–369, Barcelona, Spain, 2004. Springer-Verlag.

[Sie75] J. Siekmann. String-unification. Technical report memo csm-7, University of Essex, 1975.

[Sim94] G. Simmons. Cryptoanalysis and protocol failures. *Communications of the ACM*, 37(11) :56–65, 1994.

[Sin99] S. Singh. *The code book : the evolution of secrecy from Mary, Queen of Scots, to quantum cryptography.* Doubleday & Co., Garden City, N.Y., 1999.

[SS96] S. Schneider and A. Sidiropoulos. Csp and anonymity. In E. Bertino, H. Kurth, G. Martella, and E. Montolivo, editors, *ESORICS*, volume 1146 of *Lecture Notes in Computer Science*, pages 198–218. Springer, 1996.

[Sti75] M. E. Stickel. A complete unification algorithm for associative-commutative functions. In *Proc. of the 4th IJCAI*, pages 71–76, Tiblisi, USSR, 1975.

[STW96] M. Steiner, G. Tsudik, and M. Waidner. Diffie-Hellman key distribution extended to groups. In *Proc. 3rd ACM Conference on Computer and Communications in Security, (CCS'96)*, pages 31–37. ACM Press, 1996.

[SV05] H. Seidl and K. N. Verma. Flat and one-variable clauses : Complexity of verifying cryptographic protocols with single blind copying. In *Proc. of 11thth International Conference on Logic for Programming and Automated Reasoning (LPAR'04)*, Lecture Notes in Computer Science, page To appear, Montevideo, Uruguay, 2005. Springer-Verlag.

[TMN89] M. Tatebayashi, N. Matsuzaki, and D. B. Newman. Key distribution protocol for digital mobile communication systems. In *Proc. 9th Annual International Cryptology Conference (CRYPTO'89)*, volume 435 of *Lecture Notes in Computer Science*, pages 324–333, Santa Barbara, California, USA, 1989. Springer-Verlag.

[Tur03] M. Turuani. *Sécurité des protocoles cryptographiques : décidabilité et complexité.* PhD thesis, Université Henri Poincaré, Nancy, France, December 2003.

[Tur06] M. Turuani. The cl-atse protocol analyser. In *Proceedings of the 17th International Conference on Rewriting Techniques and Applications (RTA'06)*, Lecture Notes in Computer Science, Seattle, USA, August 2006. Springer-Verlag. To appear.

[Ver03] K. N. Verma. Two-way equational tree automata for AC-like theories : Decidability and closure properties. In *Proc. 14th International Conference on Rewriting Techniques and Applications (RTA'03)*, volume 2706 of *Lecture Notes in Computer Science*, pages 180–196, Valencia, Spain, 2003. Springer-Verlag.

[VS06] M.-H. W. V. Shmatikov. Measuring relationship anonymity in mix networks. In *Workshop on Privacy in the Electronic Society*, Alexandria, VA, USA, October 2006.

[Wei99] C. Weidenbach. Towards an automatic analysis of security protocols in first-order logic. In *CADE-16 : Proceedings of the 16th International Conference on Automated Deduction*, pages 314–328, London, UK, 1999. Springer-Verlag.

[Wu98] T. Wu. The secure remote password protocol. In *Proc. 1998 Internet Society Network and Distributed System Security Symposium*, pages 97–111, San Diego, California, USA, 1998.

Index

Symboles

A

B

C

D

E

F

G

H

I

é