THÈSE

présentée à l'École Normale Supérieure de Cachan

pour obtenir le grade de

Docteur de l'École Normale Supérieure de Cachan

par : Vincent BERNAT

Spécialité : INFORMATIQUE

Théories de l'intrus pour la vérification des protocoles cryptographiques

Soutenue le 1er juin 2006

Composition du Jury :

- Roberto AMADIO — rapporteur
- Hubert COMON-LUNDH — directeur de thèse
- Jean GOUBAULT-LARRECQ — examinateur
- Jean-Pierre JOUANNAUD — président du jury
- Benjamin WERNER — rapporteur

Remerciements

Je remercie en premier lieu mon directeur de thèse, Hubert Comon-Lundh, pour sa disponibilité, ses corrections avisées, ses conseils, sa patience et sa détermination qui a permis de continuer à avancer dans les moments de découragement.

Je remercie également les différents membres du jury : Roberto Amadio et Benjamin Werner pour avoir accepté d'être rapporteurs, Jean Goubault-Larrecq pour avoir accepté de faire partie du jury en tant qu'examinateur et Jean-Pierre Jouannaud pour en avoir été le président.

Je remercie également les membres du LSV de m'avoir accueilli dans un cadre convivial et tout particulièrement les différentes personnes avec qui j'ai pu partager mon bureau pendant ces années : Véronique, Nicolas, Stéphanie, Pascal et Nathalie.

J'ai pu passer durant ma thèse quelques mois au Stanford Research Institute et, à ce titre, je remercie Harald Ruess et Natarajan Shankar de m'avoir accueilli dans leur laboratoire, ainsi que Gloria Pena pour avoir réglé mes problèmes de visa une fois sur place.

Il convient également de remercier un certain nombre de cachanais qui m'ont soutenu et supporté pendant plusieurs années, à savoir, et dans ordre ostensiblement aléatoire voire alphabétique pour ne froisser personne : Fred, M@nu, moSaN et Xabi. Il y en a beaucoup d'autres qui m'excuseront sans doute de réduire les remerciements à si peu de personnes.

Je remercie aussi les différentes personnes qui ont soutenu (et soutiennent encore !) la vie du campus de l'École, parfois au détriment de leur scolarité, en s'investissant dans les associations et les clubs animant les soirées et les week-ends : (par ordre alphabétique, j'espère ne pas avoir oublié trop de monde ou du moins, de ne pas avoir oublié de personnes rancunières) Adso, Albert, Aline, Amélie, Anne Line, Arkem, Astérix, Aude, Augustin, Ayman, Badger, Benoit, Bilou, Blaise, Blung, Bob, Brad, Braice, Bébert, Cap, Caro, Carlos, Casa, Castagn, Charles, Conan, Crevard, Crevette, Crouz, Davidor, Denis, Djoumé, Dodo, Dom, Elsa, Esteban, Fabrice, Fabricio, Flaco, Flappie, Flavien, Flodums, Floflo, Foustala, Frach, François, Fred, Freecorp, Grrrr, Grégoire, Hébus, Hélène, HF, Iso, JL, Jas, Jee, Jojo, Jude, Julie, JulienD, Juliette, Kick, Kif, Kro, Kwazar, L'ankou, Lain, Lapinot, Lau, Lo, Lolo, Loulou, Ludo, Lynn, M@nu, Maeva, Maitre Newbie, Malo, Mamie Isa, Manathan, Marco, Marine, Marion, Marionnette, Matt, MavericK, Milouse, Mit, Mog, moSaN, Moun, Muzo, Neuneu, Nico, Nik, Nurbo, O-live, Oliv, Opla, Ouille, Pastis, Pat, Pauline, Pierrot, Ping, Po, Pufff, Pédrito, Quink1, Raf, Rafik, Ratio, Regala, Renaud, Rno, Ryo, Sam, Sandor, Sayan, Schultzy, Schumpy, Sebxu, Sgnb, Sheemie, Simon, Sky, Sophie, Stan, Tab, Tibo, Tiresias, Tophe, Tsuki, Tétart, Varich, Vilnius, Vince, Walter, Will, Willou, Xabi, Yo, Zab. Les activités organisées m'ont permis d'apprécier les années passées dans une chambre de 9 m^2.

Enfin, je remercie ma mère et ma belle-soeur, Sophie, pour s'être déplacées le jour de la soutenance en transportant avec eux la moitié de mon pot de thèse.

Table des matières

CHAPITRE 1

Introduction

Sommaire

1.1 Enjeux

L'usage croissant des communications électroniques dans la vie quotidienne s'accompagne d'un besoin grandissant de garantir la confidentialité, l'authenticité et l'intégrité de celles-ci. Malgré une histoire qui commence à se faire ancienne (les débuts du commerce électronique remontent au milieu des années 90 et les recherches dans ce domaine datent de la même période), la recherche académique sur les protocoles cryptographiques continue son essor.

La conception d'un protocole cryptographique obéit à de nombreux impératifs : algorithmes que l'on veut utiliser, propriétés à garantir, moyens d'identification des différentes parties, etc. Il apparaît donc régulièrement de nouveaux protocoles qu'il convient de vérifier. Par exemple, dans le domaine de l'embarqué ou de la carte à puce, il n'est pas possible d'utiliser des protocoles reposant sur des algorithmes trop gourmands en mémoire ou en puissance processeur. Dans le domaine des décodeurs, l'une des parties est dans l'incapacité d'envoyer des messages la plupart du temps. En ce qui concerne le vote électronique, de nombreuses propriétés sont nécessaires pour singer le vote classique : traçabilité, anonymat, vérification de son vote, secret, impossibilité de voter plusieurs fois, etc.

Le besoin en protocole cryptographique est actuellement si important que de nombreux industriels développent leurs propres protocoles en interne sans consulter la communauté de chercheurs. Les solutions sont alors déployées à grande échelle en dépit du peu d'audit réalisé et il devient alors difficile de corriger les failles qui apparaissent alors au grand jour [CJ97, SPO]. On trouve également de nombreux exemples dans l'actualité récente.

En 2004, la société Diebold vend à certains états américains des machines à voter pour les élections présidentielles. Ces machines à voter présentaient de nombreux défauts [KSRW04] : par exemple, il était possible de voter un nombre illimité de fois sans que le système ne détecte rien. Il était également possible de connaître le vote d'un utilisateur ou encore de modifier les résultats du vote.

Dans un autre domaine, les communications sans-fil, l'IEEE a publié les spécifications du protocole 802.11b [IEE02], destiné au grand public, sans prévoir dans un premier temps la sécurité. Le tir a été corrigé en incluant le WEP (pour *Wire Equivalent Privacy*). Hélas, ce protocole présente de nombreuses failles, notamment algébriques (dues à l'utilisation de l'opérateur « ou exclusif »). Ces failles ont fait rapidement l'objet d'un premier papier [FMS01] de Fluhrer, Mantin et Shamir. Ceux-ci ont identifié des trames faibles qui permettaient de récupérer de manière passive des informations sur la clef. Plutôt que d'abandonner ce protocole, les constructeurs se sont d'abord contentés de corriger la faille décrite en supprimant ces trames faibles. Le WEP a donc continué à être déployé et rapidement, de nouvelles failles ont été découvertes [BGW01, Oss04, HA03, CWHWW03, Bla04]. Notamment, un hacker connu sous le pseudo KoreK a publié un logiciel [Air] qui propose de nombreuses attaques, notamment des attaques actives par injection de paquets afin d'augmenter la quantité de données sur le réseau et attaquer celui-ci plus facilement ou encore une attaque qui permet de s'aider du point d'accès pour déchiffrer un paquet reçu auparavant en quelques minutes (en utilisant les propriétés du « ou exclusif »). Au final, un réseau sans fil protégé par le WEP résiste une quinzaine de minutes tout au plus. Les constructeurs se sont alors rabattus sur WPA. Ce protocole corrige une à une les faiblesses du WEP (cf [Bla04]) et fournit un niveau de confidentialité, d'authentification et d'intégrité correct pour l'usage auquel il est destiné. Il présente toutefois encore de certaines faiblesses, citées dans [MRH04]. WPA n'est qu'une transition vers WPA2 [CWMSW04]. En effet, il permet d'utiliser le matériel existant par simple mise à jour des logiciels (il utilise les mêmes primitives que WEP). WPA2 fait l'objet actuellement de nombreuses études et devrait permettre de stopper la spirale d'insécurité des réseaux sans-fil pour le grand public.

Ces deux exemples montrent qu'il est nécessaire d'être capable de vérifier simplement et rapidement un protocole quelconque mettant en jeu des primitives cryptographiques.

1.2 Vérification de protocoles cryptographiques

1.2.1 Difficulté de la vérification

Informellement, un protocole cryptographique est un programme utilisant certaines primitives cryptographiques, telles que le chiffrement, destiné à assurer diverses propriétés, telles que le secret ou l'authentification, sur des canaux contrôlés partiellement par un intrus.

Malgré l'apparente simplicité, concevoir un protocole est une tâche difficile et sujette à de nombreuses erreurs. Des failles pour certains protocoles ont été découvertes des années après leur conception. Des exemples comme ceux présentés ci-avant montrent que le problème reste d'actualité.

Le problème de vérification a trois données :

1. La protocole à vérifier. Il détermine les primitives cryptographiques que l'on peut utiliser, les vérifications effectuées par les agents honnêtes sur les messages leur arrivant.
2. Le pouvoir de l'intrus, c'est-à-dire les possibilités qui lui sont données, comme par

exemple la possibilité de déchiffrer un message s'il en possède la clef ou la possibilité de défaire des paires. Le modèle de l'intrus le plus courant est celui de Dolev-Yao. Il est souvent étendu pour, par exemple, ajouter la possibilité d'utiliser le « ou exclusif ».

3. La propriété à vérifier. Il s'agit souvent du secret, mais même dans ce cas, il peut être partiel, temporaire, au sein d'un groupe, etc. Parmi les autres propriétés, on trouve l'authentification [Low97b], l'équité [SM01] ou l'anonymat [SS96]. Ces propriétés peuvent souvent se traduire en terme d'accessibilité. Ce n'est pas toujours le cas ; par exemple, ce n'est pas le cas de la résistance aux denis de service ou même de l'authenticité.

La propriété la plus étudiée est le secret : un terme donné reste-t-il un secret gardé entre les différents acteurs honnêtes du protocole, même en présence d'un intrus ?

La vérification automatique de protocoles vis-à-vis de cette propriété est un problème complexe. Le nombre de termes pouvant être construits par l'intrus n'est pas fini, le nombre de sessions pouvant être jouées en parallèle n'est pas fini, la taille des messages que peuvent s'échanger les principaux n'est pas non plus bornée.

1.2.2 Résultats pour l'intrus de Dolev-Yao

Il a dans un premier temps été simplifié en considérant les opérateurs de chiffrement comme des boîtes noires : déchiffrer un message nécessite la clef correspondante. De manière plus générale, on considère que les messages sont les termes d'une algèbre libre. Malgré ces limitations, de nombreuses attaques ont été trouvées sur des protocoles avec cette hypothèse, dont la fameuse attaque [Low96] sur le protocole de Needham-Schroeder.

Cette simplification est toutefois insuffisante pour obtenir une procédure de décision [DLMS99]. Il existe toutefois un résultat général dans le cas d'un nombre borné de sessions pour l'intrus de Dolev-Yao : le problème de sécurité est co-NP complet [RT01]. Dans le cas d'un nombre non borné de sessions, il est possible de poser des restrictions supplémentaires pour obtenir la décidabilité : n'autoriser qu'une seule copie de messages par étape pour les protocoles sans nonces [CLC03a], ajouter des balises aux messages [BP03, RS03] ou encore borner la taille des messages [CKR+03]. [CDL06] contient de nombreux autres exemples de décidabilité.

Les recherches citées ont donné lieu à la naissance de nombreux outils [Cor02, BLP02, Bla01, Low97a, DM00, CV01, ABB+05]. Parmi ces outils, les plus aboutis semblent être [Bla01, ABB+05]. ProVerif [Bla01] travaille en interne avec des clauses de Horn et utilise donc certaines approximations. Il est capable de vérifier le secret et certaines propriétés d'authentification. AVISPA [ABB+05] a pour but de vérifier de manière efficace des protocoles utilisés dans la vie réelle. Il utilise des sur-approximations ou des sous-approximations selon que l'on souhaite rechercher une attaque ou prouver qu'il n'y en a pas.

1.2.3 Affaiblissement de l'hypothèse du chiffrement parfait

La vérification de la sécurité des protocoles cryptographiques dans le cas de l'intrus de Dolev-Yao et du secret semble désormais bien rodée. Les recherches s'orientent désormais vers l'extension de ce modèle pour se rapprocher de la réalité. En effet, les algorithmes de chiffrement utilisés habituellement ne sont pas parfaits : il est possible d'obtenir certaines informations sans être en possession de la clef. Par exemple, le chiffrement à clef publique RSA ou le chiffrement symétrique AES utilisant le mode ECB ou CBC ont des propriétés algébriques qui peuvent être exploitées par l'intrus. AES en mode CBC (qui est le mode conseillé, mais en utilisant une graine) est sensible à la propriété de préfixe : à partir de $\{\langle m_1, m_2\rangle\}_k$, il est possible de déduire $\{m_1\}_k$ sans connaître la clef k.

Il est donc possible que des protocoles déclarés sûrs contre l'intrus de Dolev-Yao présentent des failles contre un intrus plus puissant, c'est-à-dire capable d'exploiter certaines propriétés algébriques. Un exemple intéressant est le WEP qui souffre de nombreuses vulnérabilités graves en raison de la mauvaise utilisation du « ou exclusif ».

Parmi les propriétés algébriques à étudier, on dénombre, entre autres, l'associativité, la commutativité, le ou exclusif, les groupes abéliens, l'homomorphisme et la propriété de préfixe. De nombreux résultats de décision ont été établis pour les problèmes de sécurité en nombre borné de sessions en présence de ces propriétés. On citera, par exemple, [CKRT05] pour la propriété de commutation, [CKRT03b, CLS03] pour le cas du ou exclusif, [CKRT03a] pour le cas de Diffie-Hellman, [CRZ05] pour CBC. La combinaison de différentes théories équationnelles disjointes est traitée dans [CR05]. Pour une vue d'ensemble, des travaux réalisés dans ce domaine, on pourra consulter la thèse de Mathieu Turuani [Tur03] ou le panorama très complet proposé dans [CDL06].

Peu d'outils savent prendre en compte les propriétés algébriques. ProVerif [Bla01] a été modifié pour pouvoir exploiter les propriétés algébriques qui peuvent se traduire en clauses de Horn. Leur traitement est toutefois naïf et le prouveur peut ne pas terminer. On pourra consulter [CDL06] pour plus d'informations sur ce point.

1.3 Faiblesses

Les travaux ci-dessus présentent toutefois l'inconvénient de traiter chaque théorie équationnelle de manière *ad hoc*. [CKRT03b] introduit des règles d'oracle qui sont des conditions sur le pouvoir de l'intrus pour avoir un résultat de décidabilité. Les propriétés mises en avant ne sont toutefois pas purement syntaxiques et ne couvrent qu'une gamme réduite de théories. Les travaux de Mathieu Turuani [Tur03] suggèrent néanmoins que des résultats généraux peuvent êtres obtenus par normalisation de preuves.

Il n'existe actuellement aucun outil permettant de vérifier les protocoles qui soit paramétré par les pouvoirs de l'intrus.

1.4 Contribution

L'objectif de cette thèse est donc de contribuer à définir un formalisme et à donner des résultats dans lesquels le système de déduction de l'intrus est un paramètre.

Dans le cas où l'intrus ne fait qu'observer les messages circulant sur les réseaux sans les altérer (intrus passif), le problème de sécurité est un problème de déductibilité dans le système d'inférence représentant les capacités de l'intrus. Dans presque tous les cas, ce problème est décidable grâce à une propriété de sous-formule [CLT03] appelée *localité* suite à [McA93].

En se basant sur le travail préliminaire de [CL04], nous avons choisi de définir un modèle pour la sécurité des protocoles cryptographiques qui généralise le cas de l'intrus passif. Le protocole est vu comme une capacité supplémentaire de l'intrus : il est utilisé comme *oracle* pour obtenir de nouveaux messages.

Le problème de sécurité en présence d'un intrus actif devient donc un problème de déductibilité dans un système d'inférence augmenté par ces règles d'oracle. Dans le cas d'un nombre non borné de sessions, l'intrus a un nombre non borné d'accès à l'oracle.

L'objectif de la thèse est d'obtenir une propriété de localité pour le système de preuve obtenu en supposant une telle propriété pour le système correspondant à l'intrus passif. Classiquement, les propriétés de sous-formule sont obtenues en calcul des séquents en éliminant les coupures. De façon analogue, notre résultat principal est un théorème de normalisation de preuves. Les preuves en forme normale possèdent une propriété de sous-formule qui permet de réduire l'espace de recherche en nombre non borné de sessions et conduit à un résultat de décision en nombre borné de sessions. Dans tous les cas, notre résultat devrait permettre de construire un outil de preuve automatique.

En l'état, nous n'avons obtenu de résultats de normalisation de preuve qu'en l'absence de théorie équationnelle. Cependant, en se basant sur le travail de [CLD05], il est possible de se ramener dans la plupart des cas à une théorie équationnelle triviale ou réduite à associativité-commutativité (AC), au prix d'une modification du système de déduction de l'intrus. Mais comme notre résultat est paramétré par ce système de déduction, on peut espérer que notre résultat soit applicable à une large classe de théories équationnelles. Il resterait cependant à étendre le résultat au cas AC.

1.5 Plan de la thèse

Cette thèse est composée de trois parties. La première partie consiste en la présentation d'un nouveau modèle pour la sécurité des protocoles cryptographiques, la seconde est constituée d'un résultat de normalisation des preuves ouvrant sur la troisième partie qui présente un algorithme de recherche d'attaques sur des protocoles en nombre bornés de sessions ou non.

Modèles et théories de l'intrus Dans cette partie, le chapitre 2 présente les travaux existants pour définir un cadre de travail pour la vérification de protocoles cryptographiques ainsi que leurs limitations. Le chapitre 3 présente un système de transition proche des travaux existants et qui nous servira de référence dans la suite. Le chapitre 4 introduit notre première contribution : l'expression du secret comme un problème de déductibilité dans un calcul de séquents. Ce chapitre contient également de nombreux exemples. Le chapitre 5 apporte une preuve de complétude et de correction du modèle vis-à-vis du système de transition. Dans le chapitre 6, nous présentons un système de preuve restreint ayant la même expressivité que celui du chapitre 4.

Dans cette partie, les primitives cryptographiques peuvent satisfaire des propriétés algébriques définies par une théorie équationnelle : l'algèbre des messages n'est pas supposée libre.

Théoreme de normalisation Le chapitre 7 présente l'objectif de cette partie : démontrer un théorème permettant de normaliser les preuves de façon à ce que les termes utilisés se trouvent dans un ensemble calculable à partir des hypothèses et de la conclusion. Le chapitre 8 présente deux lemmes importants pour simplifier la preuve. Ces lemmes constituent une première normalisation de l'arbre de preuve. Le chapitre 9 présente un résultat d'indécidabilité pour un nombre borné de sessions. Nous introduisons alors des hypothèses sur les capacités de l'intrus qui nous permettrons de rétablir la décision en nombre borné de sessions. Ces hypothèses portent sur la forme des règles utilisées et il est donc aisé de vérifier si telle ou telle théorie les satisfait. Le chapitre 10 énonce et prouve le théorème. Il apporte les premières justifications des hypothèses du chapitre précédent à travers des exemples. Il contient également une application aux signatures en aveugle apportant un résultat nouveau.

Algorithmique de recherche de preuve La dernière partie indique les implications du résultat présenté dans la partie précédente en terme de procédures décisionnelles. Le chapitre 11 présente un algorithme non déterministe de recherche d'attaques en nombre borné de sessions. Le chapitre 12 présente un algorithme déterministe correct et complet pour la recherche d'attaque en nombre borné de sessions ou non. De plus, il termine pour un nombre borné de sessions.

Première partie

Modèles et théories de l'intrus

CHAPITRE 2

Modèles existants

Sommaire

La première étape vers la vérification d'un protocole cryptographique est son expression dans une syntaxe suffisamment générale pour permettre d'exprimer le protocole de façon précise. Il convient ensuite de choisir une syntaxe pour les propriétés à vérifier. Enfin, il reste à choisir le modèle permettant de définir la sémantique des protocoles et des propriétés. Le modèle doit être assez complet pour que la propriété à vérifier ait toujours un sens mais assez compact pour pouvoir faire travailler les outils de vérification de manière efficace.

Les syntaxes choisies dépendent généralement des propriétés que l'on cherche à vérifier. Il n'est donc pas rare qu'elles soient centrées sur le secret. Les autres propriétés sont généralement ramenées à des propriétés d'accessibilité en modifiant le protocole en lui-même plutôt qu'en travaillant sur la syntaxe des propriétés ; c'est le cas par exemple dans [Bla02].

Le travail académique s'est rapidement orienté vers la recherche d'attaques contre le secret ou l'authentification et les preuves automatiques, en négligeant la syntaxe des protocoles. On pourra par exemple citer la logique BAN [AT91] qui constitue un des premiers travaux. La logique proposée reste cependant difficile à manipuler et sujette à des erreurs d'interprétation. Elle avait cependant le mérite de permettre les premiers travaux sur l'authentification. Ensuite, chacun a proposé sa propre syntaxe, ce qui explique le foisonnement des syntaxes possibles pour décrire un protocole dans la littérature. Face à la multiplicité de ceux-ci, certains travaux ne définissent pas de syntaxe particulière et se rabattent sur celle de [CJ97] qui s'apparente plus à une description informelle du protocole, ce qui a le défaut de laisser en suspens certains détails du protocole : en effet, le protocole est décrit comme une suite de messages échangés par les principaux :

$$\begin{array}{ll} A \to B & \{A, N_A\}_{K_B} \\ B \to A & \{N_A, N_B\}_{K_A} \\ A \to B & \{N_B\}_{K_B} \end{array}$$

Cette notation permet d'avoir une intuition rapide du protocole, mais ne répond pas à certaines questions. Par exemple, A vérifie-t-elle le nonce N_A dans le message qu'elle reçoit de B ?

La comparaison des différentes syntaxes est également assez compliquée. Le spi-calcul a par exemple une notion bien différente du secret par rapport à un système basé sur la réécriture : dans le cas du spi-calcul, la propriété de secret est souvent exprimée par une propriété d'observabilité tandis que dans le cas d'un système de réécriture, on privilégie l'expression des propriétés par des notions d'accessibilité dans les traces. Il existe cependant des comparaisons entre modèles, comme [CDL+03, BCLM05, Cor03].

La définition d'un *framework* pour la vérification des protocoles cryptographiques nécessite de choisir une syntaxe pour les protocoles, une syntaxe pour les propriétés, une sémantique et un système de preuves. Ces éléments sont relativement indépendants et il est donc possible de piocher dans chacune des catégories pour construire un framework adapté.

2.1 La syntaxe

La syntaxe consiste à définir comment décrire :
- le protocole
- la propriété
- l'intrus

La plupart des syntaxes partagent des bases communes, à savoir la définition des termes et des messages à l'aide des principaux, des primitives cryptographiques et des atomes.

Il existe cependant des variations (au-delà du choix des notations) comme par exemple les syntaxes typées. Le typage permet d'éviter certaines attaques comme celle que l'on peut mener dans la première version du protocole de Needham-Schroeder modifié par Lowe [Low96]. Toutefois, les protocoles actuels n'implémentent pas de typage la plupart du temps, y compris les plus récents. Il existe de plus une méthode moins onéreuse que le typage pour arriver à des résultats similaires : les protocoles « balisés » tels que décrits dans [BP03, RS03]. L'implémentation d'un protocole utilisant cette technique est plus simple et moins sujette à erreur. On peut par exemple citer le protocole SSH qui utilise cette technique.

2.1.1 Les messages

Pour construire un message, il est fait appel à différents constructeurs. Les plus courants sont le chiffrement symétrique, le chiffrement asymétrique et la paire. On peut également considérer la génération d'un nonce comme un constructeur.

Le chiffrement consiste en la combinaison d'un message et d'une clef. Si le message est noté m et la clef k, le chiffrement est habituellement noté $\{m\}_k$. On ne distingue pas par cette notation les différents types de chiffrement qui peuvent exister, notamment le chiffrement à clef publique ou celui à clef secrète. On peut soit utiliser des opérateurs différents, comme

$\{m\}_k^p$ et $\{m\}_k^s$, ou considérer que l'espace des clefs est disjoint en utilisant par exemple un opérateur pour obtenir une clef publique et une clef privée à partir d'un terme.

La paire est la résultat de la concaténation de deux messages m_1 et m_2 et est notée $< m_1, m_2 >$. Elle reste le plus souvent un symbole binaire.

Un nonce est une donnée aléatoire que l'intrus ne peut deviner. On lui octroie généralement un espace de noms distinct des autres termes.

Les constructeurs ci-dessus permettent de constituer la base du modèle de Dolev-Yao. Celui-ci peut être étendu en ajoutant de nouveaux opérateurs, comme par exemple le « ou exclusif », le hash, les signatures, les timestamps ou l'exponentiation modulaire. Ces opérateurs peuvent alors être accompagnés d'une théorie équationnelle.

Les messages sont également constitués d'atomes, comme par exemple des constantes, mais aussi le nom des principaux et souvent, un ensemble de clefs.

La combinaison de tous ces éléments permet de décrire une grammaire qui définira la forme des termes dans les messages. Cette grammaire est généralement très similaire dans tous les travaux existants. Les variations portent principalement sur les symboles supplémentaires ou sur les inclusions des ensembles. On pourra par exemple prendre :

$$\begin{array}{lcl} \text{Message} & ::= & \text{Nonces} \mid \text{Atomes} \mid \\ & & \langle \text{Message}, \text{Message} \rangle \mid \{\text{Message}\}_{\text{Message}} \mid \\ & & h(\text{Message}) \mid \text{Message} \oplus \ldots \oplus \text{Message} \end{array}$$

2.1.2 Le protocole et l'intrus

La description du protocole est beaucoup plus disparate. Elle peut être basée sur des processus, des logiques, des règles de réécriture ou encore des descriptions *ad hoc*. Nous allons donner ici les exemples les plus importants.

Lorsque l'intrus est limité à Dolev-Yao, il n'est généralement pas décrit syntaxiquement. Ces capacités sont alors souvent décrites dans la sémantique.

2.1.2.1 Envoi de messages

Une des syntaxes les plus simples est celle utilisée dans [CJ97]. Elle consiste à décrire le protocole par les envois de messages entre les principaux dans le cas d'une exécution honnête sur une seule session. Par exemple, pour le protocole de Needham-Schroeder, le protocole serait décrit de cette façon :

$$\begin{array}{ll} A \rightarrow B & \{A, N_A\}_{K_B} \\ B \rightarrow A & \{N_A, N_B\}_{K_A} \\ A \rightarrow B & \{N_B\}_{K_B} \end{array}$$

Cette syntaxe est très pauvre et on lui préfère souvent une syntaxe plus riche différenciant les messages reçus des messages envoyés, introduisant des variables, mettant correctement en évidence les paramètres du protocole et les nonces utilisés par chacun des acteurs. Une telle syntaxe est décrite dans la section 3.3 page 28. Elle est également utilisée dans [RT01, Tur03].

L'intrus n'est pas décrit avec une telle syntaxe. On le décrit alors généralement avec des clauses de Horn, un système de réécriture ou un système de déduction.

2.1.2.2 Clauses de Horn

Une deuxième possibilité pour décrire les protocoles cryptographiques est de faire appel à des logiques telles que les clauses de Horn. L'avantage d'une telle approche est qu'elle donne accès immédiatement à un certain nombre d'outils (par exemple, Prolog) et de résultats connus. ProVerif [Bla01] accepte par exemple de spécifier le protocole en tant que clauses de Horn. C'est également le cas de [CLC03b].

Les messages sont des termes et des prédicats sont utilisés pour indiquer qu'un terme est connu de l'intrus : $I(m)$ indique que le message m est connu de l'intrus. Le pouvoir de l'intrus et les règles du protocoles sont alors exprimées sous forme de clauses de Horn. Par exemple, $I(\langle m_1, m_2\rangle) \rightarrow I(m_1)$ est une des deux règles de projection de la paire. Le secret s est alors aisément exprimable en indiquant que $I(s)$ n'est pas dérivable de l'ensemble des clauses. Bien que Turing-complet, ce formalisme est en pratique utilisé avec des approximations. Dans [Bla01], il autorise le rejeu d'une règle déjà jouée, interdisant l'exploitation d'un secret temporaire dans le protocole. En contre-partie, il est possible de traiter un nombre infini de sessions sans efforts particuliers.

2.1.2.3 Règles de réécriture

Un protocole peut également être décrit à l'aide d'un système de réécriture, comme dans [CDL+99] basé sur la logique linéaire. Les protocoles y sont décrits en utilisant le *multiset rewriting*. Cette syntaxe permet d'utiliser Lygon [HPW96] pour effectuer la vérification ou la recherche d'attaques. Dans un tel formalisme, la première règle du protocole de Needham-Schroeder peut s'écrire : $A_o(k) \rightarrow \exists x.A_1(k, k', x), N_1(\mathrm{enc}(k', x))$. Une syntaxe équivalente est utilisé dans l'outil AVISPA [ABB+05].

Enfin, on citera [Bla05] qui montre une équivalence entre ce formalisme et celui des clauses de Horn.

2.1.2.4 Algèbre de processus

Une autre façon courante de formaliser un protocole est d'utiliser une algèbre de processus. Deux témoins de cette façon de modéliser sont CSP [Sch97] et le spi-calcul [AG97].

CSP (*Communicating Sequential Processes*) permet de décrire un protocole à l'aide de processus. L'intrus est également décrit à l'aide d'un processus. Ceux-ci peuvent être composés séquentiellement ou parallèlement. Le spi-calcul est dérivé du π-calcul : il contient en plus des primitives cryptographiques.

Les deux syntaxes sont suffisamment riches pour inclure de nouveaux opérateurs. Ils permettent de travailler avec plusieurs canaux de communication. Ils sont donc particulièrement expressifs. Il n'existe actuellement aucun outil automatique manipulant des processus arbitraires en CSP ou en spi-calcul. ProVerif [Bla01] accepte les entrées en spi-calcul mais travaille en réalité avec un sous-ensemble des clauses de Horn, moins expressif.

2.1.2.5 Programmes en C

Il est également possible de décrire un protocole avec un langage de programmation classique tel que le C. Il est peu courant de concevoir un protocole directement en C, mais un protocole prouvé correct dans un formalisme abstrait peut présenter des erreurs d'implémentation une fois traduit dans un langage de programmation classique. Ainsi, [GP05] présente un outil de recherche d'attaques prenant comme entrée du code en C.

Une telle syntaxe n'est généralement pas utilisée pour décrire l'intrus.

2.1.3 Les propriétés

Lorsque les propriétés à vérifier sont peu nombreuses, typiquement, juste une définition du secret, aucun syntaxe pour les propriétés n'est définie.

Le système de preuve utilisé par la suite a également une influence sur la syntaxe choisie pour exprimer les propriétés. Si le système de preuve est manuel, les propriétés seront énoncées en langage naturel : « à tout moment, le terme t est inconnu de l'intrus si les agents A et B sont honnêtes », par exemple.

Lorsque la syntaxe permet de traduire le protocole en un ensemble de traces, les propriétés peuvent être définies en utilisant une logique sur les traces, comme par exemple l'inaccessibilité d'un état présentant le secret. Les syntaxes pouvant aisément donner lieu à des traces sont les algèbres de processus, les clauses de Horn et les systèmes de réécriture. Pour les syntaxes ne présentant pas une telle facilité, il est toutefois possible d'utiliser le système de transition de la sémantique pour exprimer les propriétés à l'aide, par exemple, d'une logique temporelle. On pourra consulter [Ber03] sur ce point.

Les syntaxes à base d'algèbres de processus permettant également d'exprimer les propriétés de secret sous forme d'équivalence observationnelle, comme par exemple dans [AG97]. Dans de tels cas, l'algèbre de processus permet également d'exprimer les propriétés recherchées.

Les propriétés d'authentification peuvent également être exprimées en modifiant directement le protocole pour se ramener à un problème proche de la recherche d'attaques sur le secret. C'est l'approche de [Bla02].

Il n'y a pas de lien entre la syntaxe utilisée pour exprimer un protocole et celle pour décrire une propriété. Par exemple, en utilisant CSP pour décrire les protocoles, [Sch97] exprime les propriétés d'authentification dans une logique de traces.

2.2 La sémantique

Une fois les syntaxes pour le protocole, les propriétés et éventuellement l'intrus choisies, il nous faut définir la sémantique. Bien que peu souvent explicitée de cette façon, il s'agit toujours d'un système de transition.

L'avantage de modéliser un protocole dans un système de transition est la possibilité d'utiliser ensuite les outils d'exploration des états. Malheureusement, ceux-ci sont souvent en nombre infinis, limitant ainsi une telle utilisation.

De plus, lorsqu'une attaque est trouvée, il est relativement facile de la reconstituer facilement en suivant le chemin qui mène à cette attaque. Il s'agit de plus de structures bien connues qui peuvent être étendues de diverses manières. Par exemple, il est possible d'ajouter des contraintes de temps ou des transitions probabilistes. On pourra se référer à [Bau05] pour un exemple concret d'une telle extension. Ce papier étend le modèle de Dolev-Yao en y incluant la possibilité pour l'intrus d'utiliser un algorithme de type Las Vegas pour déchiffrer des messages sans la clef.

2.2.1 L'intrus

Si l'intrus n'a pas été défini au niveau syntaxique, il peut être défini au niveau du modèle en définissant les transitions autorisées entre les états selon les pouvoirs conférés à l'intrus. Le système de transition final peut être vu comme le produit du système de transition décrivant le protocole avec celui décrivant l'intrus. La théorie équationnelle est également exploitée à ce niveau.

L'intrus est l'entité centrale de la vérification de protocoles cryptographiques. L'intrus peut disposer de pouvoirs étendus. Par exemple, dans le cas de Dolev-Yao, l'intrus peut intercepter et modifier tous les messages circulant entre deux agents. Il dispose de plus de la possibilité de fabriquer les messages de son choix à partir des constructeurs et des opérations de décomposition comme le déchiffrement. Il peut également exploiter une éventuelle théorie équationnelle afin d'exploiter les propriétés algébriques de certains opérateurs.

Certains modèles bornent les capacités de l'intrus en mémoire ou en capacité de calcul. Ces restrictions sont utiles pour des protocoles mettant en œuvre des mécanismes basés sur la difficulté de certains calculs ou encore pour les protocoles probabilistes. On pourra consulter [CM97] pour un exemple d'un tel protocole. Les propriétés algébriques que nous considérons restent suffisamment abstraites pour ne pas avoir à faire de telles hypothèses.

2.2.2 Définition du système de transition

Un exemple de sémantique possible est le modèle [MR00] qui ne peut toutefois traiter que le secret car il se limite à un modèle de traces sans superposer une logique. Il se base sur les travaux de Paulson [Pau98, Pau99]. Il s'agit de considérer l'historique des messages envoyés. Les règles de protocoles indiquent les évolutions possible de cet historique. Deux opérateurs « analz » et « synth » qui permettent de calculer les termes accessibles à un intrus à partir d'un ensemble donné de termes. Les états du système de transition contiennent les états locaux des agents pour chacune des sessions ainsi que les connaissances de l'intrus. Une transition correspond au jeu d'une règle de protocole. Le secret est alors une propriété d'accessibilité sur les connaissances de l'intrus dans ce système de transition. Les informations contenues dans les états sont finies mais le nombre d'états est infini. [Cor03] montre que ce modèle est équivalent au modèle des clauses de Horn. Généralement, un modèle basé sur un système de transition a pour ambition d'exprimer des résultats qui seront valables dans d'autres modèles.

Un autre exemple de sémantique est donné dans le chapitre 3 page 25. Il s'agit du système de transition que nous utiliserons par la suite.

La sémantique va fixer de plus un certain nombre d'hypothèses sur le protocole. Elle fixe, par exemple, le comportement des agents. Les agents sont les acteurs du protocole. Ils sont rattachés à un rôle du protocole dont ils ont la charge. Pour un message reçu, ils en vérifient la forme et renvoient la réponse. Pour construire la réponse, ils peuvent décomposer le message reçu selon leurs connaissances ou user de leur mémoire et donc des messages reçus lors des étapes précédentes. Cette mémoire est généralement propre à chaque session : un agent ne peut utiliser les termes reçus dans une autre session, à moins bien sûr que ceux-ci ne lui aient été envoyés avec le concours de l'intrus.

Certains rôles acceptent des agents malhonnêtes, c'est-à-dire des agents qui ne suivent pas le rôle qui leur est assigné. Souvent, il est sous l'influence totale de l'intrus, c'est-à-dire que celui-ci a accès aux données échangées qui devraient normalement rester privées. Dans certains modèles, comme ceux destinés à vérifier les protocoles de signature de contrats, un agent malhonnête ne suit pas le protocole mais ne divulgue pas d'informations à l'intrus (cf. [CKS01]).

En ce qui concerne les nonces. Lorsque l'on travaille avec un nombre fini de sessions, on préfère les remplacer par des constantes distinctes en utilisant par exemple un symbole de fonction paramétré par le numéro de la session et la position du nonce dans la session. Pour un nombre non fini de sessions, il est correct (mais pas complet) de considérer les nonces comme fonctions des messages reçus jusqu'à un certain point comme dans [Bla01]. Il existe cependant des résultats de décidabilité pour un nombre non fini de sessions en prenant en compte des nonces, comme par exemple [RS03].

2.2.3 Les propriétés

Dans le cas où les propriétés ont été définies sur le système de transition, il est possible d'exploiter des logiques dont on connaît déjà la décidabilité. Cette indépendance vis-à-vis des logiques que l'on peut y appliquer permet de poser les bases pour comprendre les subtilités entre différentes propriétés : le modèle ne limite pas le pouvoir de l'intrus, les capacités des protocoles ou les propriétés à exprimer. [Ber03] définit un système de transition suffisamment généraliste pour s'accommoder de la plupart des protocoles, tout en permettant d'exprimer les propriétés d'authentification de [Low97b, Sch97].

Par contre, si les propriétés ont été définies sur la syntaxe, il convient de les traduire dans le système de transition. Dans le cas de l'équivalence observationnelle, la propriété de secret peut être vue comme une bisimulation : un protocole respecte le secret si et seulement si il n'est pas possible de distinguer, pour un observateur extérieur, deux processus utilisant un secret différent. Une propriété semblable est utilisée pour l'authentification. On pourra se référer à [BAF05] pour plus de renseignements sur cette approche. [AG97, AFG00] utilisent également l'équivalence observationnelle. Du fait de l'expression particulière de ces propriétés, il est difficile de comparer celles-ci avec celles exprimées en terme d'accessibilité comme c'est le cas habituellement.

2.3 Le système de preuve

Enfin, à partir du système de transition, il est possible de définir un système de preuve destiné à effectuer la recherche d'attaques sur les propriétés définies auparavant.

Dans le cas d'un système de transition défini par des clauses de Horn, comme dans [Bla01], le système de preuve est la résolution des clauses. Un prototypage rapide est alors possible avec un logiciel tel que Prolog.

De manière plus générique, il est possible d'exploiter les outils dédiés au model-checking. Le système de transition étant généralement infini (en nombre de sessions, en taille des messages, etc.), il convient de limiter certains paramètres ou d'effectuer des approximations que l'on espère le plus souvent complètes si l'on veut utiliser des outils qui effectuent une exploration exhaustive.

Selon la propriété recherchée, le système de preuve doit être correct ou complet vis-à-vis de la sémantique. Pour le secret, on lui demande souvent d'être complet. Idéalement, il devrait être correct et complet.

2.4 Application à cette thèse

Dans les chapitres suivants, nous allons définir la syntaxe, la sémantique et le système de preuve que nous allons utiliser. La *syntaxe des protocoles* sera basée sur l'envoi de messages. Nous nous limitons au secret et il n'y aura alors pas de syntaxe pour exprimer les propriétés. La propriété de secret sera basée sur le système de transition. L'*intrus* sera défini à l'aide de séquents (que l'on peut voir comme des règles de réécriture) ainsi qu'à l'aide d'une théorie équationnelle. La *sémantique* sera traduite par un système de transition dont les états présenteront les connaissances de l'intrus et l'état local de chaque agent.

Enfin, le système de preuve est basé sur des arbres de preuve regroupant en son sein les règles de protocoles et le pouvoir de déduction de l'intrus. Les règles de protocoles seront ainsi vues par l'intrus comme des oracles qu'il peut utiliser au même titre que les règles de déduction classique. Il s'agit d'une généralisation des règles d'oracle introduites dans [CKRT03b]. Il généralise les schémas de preuve habituellement limités aux règles de déduction de l'intrus en y incluant de manière naturelle dans la preuve les « nouvelles » possibilités offertes à l'intrus via les règles de protocoles.

Une telle approche ne s'intéresse pas au déroulement normal d'un protocole mais seulement aux diverses façons qu'un adversaire peut l'exploiter.

Nous montrerons que ce système de preuve est correct et complet vis-à-vis de la sémantique.

CHAPITRE 3

Système de transitions

Sommaire

Nous allons introduire dans ce chapitre un système de transitions destiné à définir la sémantique des protocoles cryptographiques. Il sera utilisé dans le chapitre 5 page 45 pour démontrer la complétude du système de preuve du chapitre 4. Ce système de transition est compatible avec [DLMS99, CMR01, CLC03b].

3.1 Messages

Les messages sont les termes d'une algèbre construite sur l'alphabet $\mathcal{F}$. Cet ensemble contient l'opération de chiffrement $\{\_\}_{\_}$ (on utilise la même notation pour le chiffrement symétrique et le chiffrement asymétrique), l'opérateur d'appariement $\langle \_, \_ \rangle$ et des constantes telles que $0, 1$, des noms d'agents et des clefs symétriques. Si l'on veut utiliser le chiffrement asymétrique, on y rajoute un opérateur pour construire des clefs publiques et des clefs privées. Pour modéliser le déchiffrement explicite, on rajoute un opérateur de déchiffrement. Enfin, selon les théories équationnelles que l'on souhaite utiliser, on peut inclure des fonctions telles que le ou exclusif, la multiplication, l'exponentiation, etc.

$T(\mathcal{F}, X)$ désigne l'algèbre libre des termes avec variables X. Les messages sont des éléments de l'algèbre quotient $T(\mathcal{F}, X)$ par $=_{\mathcal{E}}$. On pourra se reporter à [BN98] pour plus de détails sur les algèbres de termes.

Voici quelques exemples typiques de théories équationnelles $\mathcal{E}$:

La théorie libre correspond au modèle classique de Dolev-Yao sans opérateur explicite de projection ou de déchiffrement.

Le déchiffrement explicite correspond à cet ensemble d'équations :

$$\begin{array}{rclcrcl} \pi_1\langle x,y\rangle &=& x & \quad & \pi_2\langle x,y\rangle &=& y \\ \mathsf{dec}(\{x\}_y, y^{-1}) &=& x & & \{\mathsf{dec}(x,y)\}_{y^{-1}} &=& x \\ \mathsf{dec}(\{x\}_{y^{-1}}, y) &=& x & & \{\mathsf{dec}(x,y^{-1})\}_y &=& x \\ (x^{-1})^{-1} &=& x & & & & \end{array}$$

Cette théorie correspond au modèle de Dolev-Yao tel que décrit dans [DY81] étendu au cas des arbres. Les opérations de déchiffrement et de projection sont explicitement utilisées. Ce modèle n'est pas équivalent au précédent étant donné que l'intrus peut décomposer des termes qui n'étaient pas des termes composés. Ainsi, $\mathsf{dec}(\langle a,b\rangle, k)$ est un terme autorisé dans cette théorie. Cela correspond mieux à la réalité où le chiffrement donne une suite de bits et il n'est donc pas possible de savoir si celle-ci est réellement due à une opération de chiffrement.

La théorie AC , l'associativité et la commutativité de f :

$$\begin{array}{rcl} f(x,f(y,z)) &=& f(f(x,y),z) \\ f(x,y) &=& f(y,x) \end{array}$$

La théorie du ou exclusif qui inclut la théorie AC pour le symbole $\oplus$. Celui-ci est considéré comme un opérateur variadique et on ajoute les deux équations suivantes :

$$\begin{array}{rcl} x\oplus 0 &=& x \\ x\oplus x &=& 0 \\ x\oplus x\oplus y &=& y \end{array}$$

La dernière équation est redondante avec la seconde, mais elle est conservée sous cette forme afin que le système de réécriture associé soit convergent.

La théorie des groupes abéliens inclut la théorie AC pour $\bullet$ qui est, comme pour $\oplus$, considéré comme un symbole variadique. On y ajoute les équations suivantes :

$$\begin{array}{rcl} x\bullet 1 &=& x \\ x\bullet x^{-1} &=& 1 \\ x\bullet y\bullet y^{-1} &=& x \\ (x^{-1})^{-1} &=& x \\ (x\bullet y)^{-1} &=& x^{-1}\bullet y^{-1} \end{array}$$

La propriété d'homomorphisme s'applique sur deux symboles f et g. Elle est décrite par l'équation $g(f(x,y)) = f(g(x),g(y))$. Elle peut être généralisée à des arités différentes.

On considère que $\mathcal{E}$ est donnée sous forme d'un système de réécriture *convergent* modulo l'associativité et la commutativité. Il est donc nécessaire d'orienter les égalités données ci-dessus. Dans le cas de l'associativité et la commutativité, le système doit de plus être *cohérent*. On se reportera à [DJ90] pour les définitions correspondantes. En orientant les équations données ci-dessus de la gauche vers la droite, on obtient des systèmes AC-convergents. Dans ce cas, chaque terme t admet une unique forme normale notée $t\downarrow$.

Voici quelques formes normales :

$$\begin{array}{rcl}(\pi_2\pi_1\langle x,y\rangle)\downarrow & = & \pi_2 x\\ (\mathsf{dec}(\{\mathsf{dec}(x,y^{-1},\})_y,z^{-1}))\downarrow & = & \mathsf{dec}(x,z^{-1})\\ (x\oplus(y\oplus z\oplus x)\oplus z\oplus x)\downarrow & = & x\oplus y\\ ((x\bullet x\bullet y)^{-1}\bullet y)\downarrow & = & x^{-1}\bullet x^{-1}\end{array}$$

On considère de plus que $\mathcal{E}$ est cohérent : $\mathcal{E}\not\models x=y$ pour x et y deux variables distinctes, ce qui est le cas lorsqu'il est défini par un système de réécriture AC-convergent.

Par la suite, on considérera l'ensemble Σ des substitutions normalisées, c'est-à-dire des substitutions telles que l'image de toute variable est un terme sous forme normale. Pour $\sigma\in\Sigma$, on note $t\sigma$ l'application de la substitution σ à t.

3.2 Pouvoir de déduction de l'intrus

À partir de certains termes contenus dans sa connaissance initiale, l'intrus peut déduire d'autres termes en utilisant ses règles de déduction. On note T un sous-ensemble de $T(\mathcal{F},X)$ (l'algèbre des termes utilisant les symboles $\mathcal{F}$ et les variables X). Pour $u\in T(\mathcal{F},X)$, on note $T\vdash u$ le séquent dont la signification est « à partir de la connaissance T, il est possible de déduire u ».

La théorie exploitant le symbole $\vdash$ est notée $\mathcal{S}$ et est définie par un ensemble de règles d'inférence. On distingue deux types de règles d'inférence dans la théorie $\mathcal{S}$:

Les règles de composition consistent en l'ajout en tête des hypothèse d'un symbole de composition *public*. $\mathcal{F}$ est en effet séparé en deux parties disjointes : les symboles publics (utilisables par l'intrus) et les symboles privés.

$$\frac{T\vdash u_1\cdots T\vdash u_n}{T\vdash f(u_1,\ldots,u_n)\downarrow}\,\mathcal{C}$$

Le symbole f fait partie de l'ensemble $\mathcal{C}$ qui est l'ensemble des symboles de composition. Il s'agit d'un sous-ensemble de $\mathcal{F}$. La règle d'axiome est incluse dans les règles de composition : $T,u\vdash u$.

Les règles de décomposition constituent les autres règles. La distinction nous sera utile par la suite car des contraintes supplémentaires seront appliquées sur celles-ci.

Quand il existe une preuve de s en utilisant la connaissance T en utilisant les règles d'inférence de $\mathcal{S}$, on écrit $T\vdash_{\mathcal{S}} s$.

Exemple 3.1 (Dolev-Yao) *Dans le cas de Dolev-Yao (pour les clefs symétriques uniquement), les règles d'inférence sont les suivantes :*

$$\frac{T\vdash u_1\qquad T\vdash u_2}{T\vdash\langle u_1,u_2\rangle}\qquad\frac{T\vdash\langle u_1,u_2\rangle}{T\vdash u_1}\qquad\frac{T\vdash\langle u_1,u_2\rangle}{T\vdash u_2}$$

$$\frac{T\vdash u_1\qquad T\vdash u_2}{T\vdash\{u_1\}_{u_2}}\qquad\frac{T\vdash\{u_1\}_{u_2}\qquad T\vdash u_2}{T\vdash u_1}\qquad\frac{}{T,u\vdash u}$$

Exemple 3.2 (Destructeurs explicites) *À titre d'exemple, voici ce que l'on peut rajouter pour avoir le déchiffrement et la projection explicite :*

$$\frac{T \vdash u_1 \qquad T \vdash u_2}{T \vdash \mathit{dec}(u_1, u_2)} \qquad \frac{T \vdash u}{T \vdash \pi_1(u)} \qquad \frac{T \vdash u}{T \vdash \pi_2(u)}$$

3.3 Protocoles

Un protocole est constitué en un ensemble fini de *rôles* $R_1, \ldots R_m$. Un rôle comprend :

- La donnée d'un certain nombre de noms d'agents, dont l'un se distingue en étant l'acteur principal du rôle. Les agents sont donnés en paramètre du rôle et permettent à l'acteur principal de connaître les principaux auxquels il doit s'adresser (même si le modèle comprend uniquement l'envoi de messages sur le réseau, c'est une donnée nécessaire pour utiliser par exemple les bonnes clefs publiques). On note λa la génération d'un nom d'agent.
- Un certain nombre de nonces. On note νN la génération d'un nonce. Celui-ci est considéré comme unique et aléatoire (l'intrus ne peut pas le deviner). Dans le cas d'un nombre fini de sessions, cela revient à définir une nouvelle constante inconnue de l'intrus.
- Un ensemble de règles de la forme $u_i \Rightarrow v_i$ où u_i et v_i sont soit vides, soit des termes en forme normale dans l'algèbre $T(\mathcal{F}, X)$. On considère qu'il existe un ordre total sur les règles du rôle. Ainsi, il n'est possible de jouer une des règles que si la précédente a été jouée. Il s'agit d'une restriction par rapport à d'autres modèles, mais il est possible de se ramener facilement à un ordre partiel en dédoublant à l'envie les rôles. Il n'y a donc pas de perte d'expressivité.
  On considère également que les variables de v_i sont contenues dans les paramètres, les nonces et les variables des u_j avec $j \leq i$. Il n'est donc pas possible d'introduire une variable non connue jusque là. Ainsi, si les messages injectés sur le réseau sont clos, les principaux n'envoient que des messages clos également.

On notera :

$$R = \lambda a \lambda x_1 \ldots \lambda x_n \nu N_1 \ldots \nu N_k u_1 \Rightarrow v_1 \vdots u_l \Rightarrow v_l$$

Exemple 3.3 (Denning Sacco) *Voici un exemple de protocole à deux règles inspiré du protocole de Denning et Sacco écrit dans un formalisme simple :*

$$\begin{array}{ll} A \to B & \left\langle A, \{\{K_{ab}\}_{pub(A)^{-1}}\}_{pub(B)} \right\rangle \\ B \to A & \{s_b\}_{K_{ab}} \end{array}$$

Dans ce protocole, Alice envoie son identité et une nouvelle clef destinée à être partagée avec Bob. Cette clef est signée par Alice et chiffrée avec la clef publique de Bob. Bob répond en chiffrant un secret avec la clef ainsi envoyée par Alice. Cela correspond aux deux rôles suivant :

rôle « Alice » $\lambda a, \lambda b, \nu K_{ab}$.
$\quad \Rightarrow \left\langle a, \{\{K_{ab}\}_{pub(a)^{-1}}\}_{pub(b)} \right\rangle$
$\quad \{x\}_{K_{ab}} \Rightarrow$

rôle « Bob » $\lambda b, \nu s_b$.
$\quad \left\langle y, \{\{z\}_{pub(y)^{-1}}\}_{pub(b)} \right\rangle \Rightarrow \{s_b\}_z$

Si on utilise le déchiffrement et la projection explicite, la règle du rôle de Bob peut également se réécrire ainsi :

$$y \Rightarrow \{s_b\}_{dec(dec(\pi_2 y, pub(b)^{-1}), pub(\pi_1 y))}$$

La description d'un protocole comme décrit ci-dessus ne permet pas d'implanter en pratique le protocole : les vérifications effectuées par les différents acteurs ne sont pas explicites. On considère habituellement que ceux-ci effectuent le maximum de vérifications.

3.4 Système de transition

À partir des règles de l'intrus et des règles de protocoles, nous allons définir un système de transition définissant la sémantique des protocoles cryptographiques. Ce système est inspiré de [Ber03].

3.4.1 Définition

On définit $\mathcal{A}$ comme un ensemble infini de noms d'agents. Cet ensemble est séparé en deux parties disjointes. Un agent peut être honnête ou malhonnête : $\mathcal{A} = \mathcal{A}_h \uplus \mathcal{A}_m$. Une *session* est un entier naturel.

Les *nonces* sont des fonctions indicées par la position du nonce dans les règles du protocole parmi les autres nonces qui à une session associe un nonce : $N_1(s), \ldots, N_k(s)$.

Un *état* q est l'aggrégation des données suivantes :

- un ensemble $\mathcal{I}_q$ de termes représentant la *connaissance de l'intrus*
- pour chaque nom d'agent a, un *état local* propre à cet agent.

Un état local est une fonction partielle $M_{q,a}$ dont l'ensemble de définition est contenu dans $\mathbb{N}$ (représentant l'ensemble des sessions) et telle que $M_{q,a}(s)$ est un tuple contenant un rôle, un numéro d'étape et un ensemble d'affectations de variables (variables paramétrant le protocole ou le rôle) sous forme de substitution.

Pour l'état initial q_0, la connaissance de l'intrus $\mathcal{I}_0$ est limitée aux connaissances initiales. La relation $M_{q_0,a}$ est vide pour tout a.

Il est possible d'effectuer une transition de l'état q vers l'état q' si et seulement si ces deux états diffèrent par la connaissance de l'intrus et par l'état local d'un unique agent a pour une unique session s, c'est à dire qu'il existe a et s tels que pour tout $a' \neq a$ et pour tout s', $M_{q,a'}(s') = M_{q',a'}(s')$ et pour tout $s \neq s'$, $M_{q,a}(s') = M_{q',a}(s')$. Il existe de plus un rôle R tel que l'une des deux conditions suivantes soit vérifiée :

Début de session Soit $u \Rightarrow v$ la première règle du rôle R et σ l'affectation des variables de sessions telle que l'acteur principal du rôle est a. Ou bien u est vide, ou bien $\mathcal{I}_q \vdash_{\mathcal{S}} u\theta\downarrow$ avec θ tel que $x\sigma = x\theta$ pour les variables communes aux domaines de θ. Cette condition signifie que l'intrus est capable de fabriquer le message nécessaire à l'exécution de la règle.

$s \notin \mathrm{Dom}(M_{q,a})$ et $M_{q',a}(s) = (R, 1, \sigma \uplus \theta)$. Cela signifie également qu'un numéro de session n'est pas partagé entre plusieurs acteurs jouant un même protocole.

Les connaissances de l'intrus sont augmentées du terme $v\theta\downarrow$ (qui correspond au message envoyé). Si a est compromis ou malhonnête, on considère qu'il lui cède ses connaissances et particulièrement les nonces qu'il a généré. On ajoute donc $N_1(s), \ldots, N_k(s)$.

On a donc $\mathcal{I}_{q'} = \mathcal{I}_q \cup \{v\theta\downarrow\}$ si a est honnête ou bien $\mathcal{I}_{q'} = \mathcal{I}_q \cup \{v\theta\downarrow, N_1(s), \ldots, N_k(s)\}$, si a est malhonnête.

Suite de session Considérons que $u \Rightarrow v$ est la règle numérotée k de R. $M_{q,a}(s)$ est défini et égal à $(R, k-1, \sigma)$ pour un certain σ. Il doit y avoir une substitution θ telle que $\mathcal{I}_q \vdash_{\mathcal{S}} u\theta\downarrow$ et $x\sigma = x\theta$ sur les variables communes.

Dans ce cas, $M_{q',a}(s) = (R, k, \sigma \uplus \theta)$. Les connaissances de l'intrus sont augmentées du message $v\theta\downarrow$ envoyé sur le réseau : $\mathcal{I}_{q'} = \mathcal{I}_q \cup \{v\theta\downarrow\}$.

Afin de simplifier le système de transition, on considère que deux instances distinctes de rôle ne partagent pas les mêmes variables. Cela permet de définir pour un état q la substitution σ_q telle que $x\theta = x\sigma_q$ pour tout couple a, s telle que $M_{q,a}(s) = (R, k, \theta)$ et pour toute variable x dans le domaine de θ.

3.4.2 Secret

Il est maintenant possible de définir le secret pour un terme dans ce modèle.

Définition 3.1 (Définition du secret) *Il y a une attaque sur le* secret *d'un terme $t \in T(\mathcal{F}, X)$ si et seulement si il existe une substitution σ et un état q accessible tel que $\mathcal{I}_q \vdash_{\mathcal{S}} t\sigma\downarrow$ et tel que toutes les variables de session apparaissant dans t sont substitutées par des sessions paramétrées uniquement par des acteurs honnêtes.*

3.4.3 Exemple

À moins que tous les rôles soient constitués de 0 règle, le système de transition est infini.

Nous reprenons l'exemple du protocole de Denning Sacco défini dans l'exemple 3.3 page 28. Voici quelques états du système de transition correspondant.

q_0 correspond à l'état initial. On considère que les connaissances initiales de l'intrus se limitent aux identités A, B, I et $\mathrm{pub}(I)^{-1}$. Les agents A et B sont honnêtes, I est malhonnête.

$$q_0 \left\{ \begin{array}{lcl} I_{q_0} & : & A, B, I, \mathrm{pub}(I)^{-1} \\ M & : & M_{q_0,A} = M_{q_0,B} = M_{q_0,I} = \perp \end{array} \right.$$

Imaginons que l'acteur A joue dans le rôle d'Alice la première règle du protocole. Il n'y a aucune précondition. On suppose qu'il tente de la jouer avec l'intrus sous sa propre identité (donc I).

$$q_1 \left\{ \begin{array}{lcl} I_{q_1} & : & A, B, I, \mathrm{pub}(I)^{-1}, \left\langle A, \{\{K_{ab}(1)\}_{\mathrm{pub}(A)^{-1}}\}_{\mathrm{pub}(I)} \right\rangle \\ M & : & M_{q_1,A}(1) = (\mathrm{Alice}, 1, \{a \mapsto A, b \mapsto I\}) \end{array} \right.$$

L'intrus est capable de déduire de $\left\langle A, \{\{K_{ab}(1)\}_{\mathrm{pub}(A)^{-1}}\}_{\mathrm{pub}(I)} \right\rangle$ le terme $K_{ab}(1)$. Il est donc capable de construire le terme $\{I\}_{K_{ab}(1)}$. Il peut donc répondre à A. Cela permet de construire l'état q_2 :

$$q_2 \left\{ \begin{array}{lcl} I_{q_2} & : & A, B, I, \mathrm{pub}(I)^{-1}, \left\langle A, \{\{K_{ab}(1)\}_{\mathrm{pub}(A)^{-1}}\}_{\mathrm{pub}(I)} \right\rangle \\ M & : & M_{q_1,A}(1) = (\mathrm{Alice}, 1, \{a \mapsto A, b \mapsto I\}) \\ & & M_{q_2,A}(1) = (\mathrm{Alice}, 2, \{a \mapsto A, b \mapsto I, x \mapsto I\}) \end{array} \right.$$

Pour continuer l'exemple, nous allons commencer également une session avec B. L'intrus peut à partir des connaissances contenues dans l'état q_2 construire le terme suivant :

$$\left\langle B, \{\{K_{ab}(1)\}_{\text{pub}(I)^{-1}}\}_{\text{pub}(B)}\right\rangle$$

Il pouvait aussi le faire au niveau de l'état q_1, il existera donc un état très proche de celui que l'on va construire. Il va transmettre ce terme à B pour lui inciter à ouvrir une session. On obtient alors l'état suivant :

$$q_3 \left\{ \begin{array}{lcl} I_{q_3} & : & A, B, I, \text{pub}(I)^{-1}, \left\langle A, \{\{K_{ab}(1)\}_{\text{pub}(A)^{-1}}\}_{\text{pub}(I)}\right\rangle \\ & & \{s_b(2)\}_{K_{ab}(1)} \\ M & : & M_{q_1,A}(s_1) = (\text{Alice}, 1, \{a \mapsto A, b \mapsto I\}) \\ & & M_{q_2,A}(1) = (\text{Alice}, 1, \{a \mapsto A, b \mapsto I, x \mapsto I\}) \\ & & M_{q_3,B}(2) = (\text{Bob}, 1, \{b \mapsto B, y \mapsto I, z \mapsto K_{ab}(1)\}) \end{array} \right.$$

Comme le terme utilisé par l'intrus pouvait être construit dès l'état q_1, on a également l'état suivant :

$$q_4 \left\{ \begin{array}{lcl} I_{q_4} & : & A, B, I, \text{pub}(I)^{-1}, \left\langle A, \{\{K_{ab}(1)\}_{\text{pub}(A)^{-1}}\}_{\text{pub}(I)}\right\rangle \\ & & \{s_b(3)\}_{K_{ab}(1)} \\ M & : & M_{q_1,A}(1) = (\text{Alice}, 1, \{a \mapsto A, b \mapsto I\}) \\ & & M_{q_4,B}(3) = (\text{Bob}, 1, \{b \mapsto B, y \mapsto I, z \mapsto K_{ab}(1)\}) \end{array} \right.$$

Au niveau des transitions, on a $q_0 \to q_1 \to q_2 \to q_3$ et $q_1 \to q_4$.

CHAPITRE 4

Les protocoles utilisés comme oracle par un intrus

Sommaire

Dans le chapitre précédent, l'intrus disposait d'un certain nombre de règles de réécriture, présentées sous forme de règles de déduction, pour construire certains termes. Nous pourrions qualifier ce modèle d'intrus passif. Nous allons maintenant compléter ces règles en permettant à l'intrus d'utiliser les règles de protocole comme des règles d'oracle et de les exploiter pour construire des termes supplémentaires pouvant mener à une attaque sur le secret. Nous obtiendrons alors un modèle d'intrus actif.

Les règles de protocole seront donc transformées en règle de déduction que l'intrus pourra utiliser au même titre que, par exemple, l'opération de chiffrement. Elles devront traduire pour une règle $u \Rightarrow v$ que si l'intrus connaît une instance $u\sigma$, alors, sous certaines conditions, il connaît l'instance correspondante $v\sigma$. Ces règles de déduction supplémentaires vont permettre de définir un système de preuve.

Une attaque sur un terme va alors se traduire en une substitution σ et un entrelacement de règles de protocole, compatible avec la spécification du protocole. Afin de garder intacte la substitution σ, témoin de l'attaque, nous allons utiliser des séquents contraints. La contrainte correspondra aux substitutions effectuées dans les variables du protocole pour mener à bien l'attaque.

Ce système de preuve sera ensuite prouvé équivalent dans le chapitre 5 page 45 au modèle défini dans le chapitre 3.

4.1 Séquents contraints

4.1.1 Définition du séquent contraint

Définition 4.1 (Séquent contraint) *Un séquent contraint est un triplet T, u, E, noté $T \vdash u \;\; [\![E]\!]$, où :*
- *T est un sous-ensemble fini de $T(\mathcal{F})$,*
- *u est un terme de $T(\mathcal{F}, X)$,*
- *E est une conjonction finie $\bigwedge_i u_i = v_i$ où $u_i, v_i \in T(\mathcal{F}, X)$.*

E est vu comme une contrainte sur des variables introduites par les futures règles de protocole du modèle. Les équations qui y sont présentes sont interprétées modulo $\mathcal{E}$. On dit alors que σ est une *$\mathcal{E}$-solution de E* si pour chaque équation $u = v$ de E, $u\sigma =_{\mathcal{E}} v\sigma$. Cette formule peut aussi s'écrire $\sigma \models_{\mathcal{E}} u = v$.

Notons que $T \vdash u \;\; [\![E]\!]$ peut également être vu sous la forme d'un ensemble de séquents non contraints. Plus précisément, il s'agit de l'ensemble des séquents $T\sigma \vdash u\sigma$ tels que $\sigma \models_{\mathcal{E}} E$.

4.1.2 Calcul de séquents contraints

Les règles utilisées dans le calcul de séquents contraints sont de la forme suivante :

$$\frac{T \vdash t_1 \;\; [\![E_1]\!] \quad \ldots \quad T \vdash t_n \;\; [\![E_n]\!]}{T \vdash t\downarrow \;\; [\![E]\!]} \text{ Si } C$$

La condition C est de forme arbitraire. Afin de définir l'applicabilité d'une règle, on impose le non partage de variables entre $t_1, \ldots, t_n, E_1, \ldots, E_n$. Ainsi,

$$\frac{T \vdash \{x\}_y \;\; [\![E_1]\!] \quad T \vdash y \;\; [\![E_2]\!]}{T \vdash x \;\; [\![E_1 \wedge E_2]\!]}$$

devra être réécrite de façon à ne faire apparaître y qu'une seule fois :

$$\frac{T \vdash \{x\}_y \;\; [\![E_1]\!] \quad T \vdash z \;\; [\![E_2]\!]}{T \vdash x \;\; [\![E_1 \wedge E_2]\!]} \text{ si } y = z$$

Cette linéarisation concerne également les contraintes. Imaginons que l'on souhaite disposer de la règle suivante :

$$\frac{T \vdash a \;\; [\![E]\!] \quad T \vdash b \;\; [\![E]\!]}{T \vdash c \;\; [\![E]\!]}$$

alors, il faudra la réécrire sous la forme :

$$\frac{T \vdash a \;\; [\![E]\!] \quad T \vdash b \;\; [\![E']\!]}{T \vdash c \;\; [\![E]\!]} \text{ si } E = E'$$

Nous pouvons alors définir l'applicabilité d'une règle.

Définition 4.2 (Application valide d'une règle)

$$\frac{T \vdash u_1 \ \llbracket F_1 \rrbracket \quad \ldots \quad T \vdash u_n \ \llbracket F_n \rrbracket}{T \vdash u \ \llbracket F \rrbracket}$$

est une application valide d'une règle dans le calcul de séquents contraints s'il existe une règle

$$\frac{T \vdash t_1 \ \llbracket E_1 \rrbracket \quad \ldots \quad T \vdash t_n \ \llbracket E_n \rrbracket}{T \vdash t \ \llbracket E \rrbracket} \ si\ C$$

telle qu'il existe une substitution σ sur les variables de la règle telle que :

- $\forall i, t_i\sigma =_{\mathcal{E}} u_i$
- $\forall i, E_i\sigma =_{\mathcal{E}} F_i$
- $\sigma \models_{\mathcal{E}} C$
- $t\sigma \downarrow = u$ *(donc u est en forme normale)*
- $E\sigma = F$

Dans la suite de ce chapitre, nous allons donc définir le système de preuve $\widehat{\mathcal{S}}$ obtenu à partir de $\mathcal{S}$ et du pouvoir induit par les règles de protocole.

4.2 Pouvoir de déduction de l'intrus

Si $\mathcal{S}$ est l'ensemble des règles de l'intrus du chapitre 3, on lui associe $\widehat{\mathcal{S}}$ le système de règles d'inférence sur les séquents contraints défini dans la figure 4.1. La suite de cette section détaille les règles présentées dans la figure.

Supposons que la règle suivante soit dans $\mathcal{S}$:

$$\frac{T \vdash u_1 \quad \ldots \quad T \vdash u_k}{T \vdash u}$$

Dans $\widehat{\mathcal{S}}$, nous allons introduire la règle qui lui correspond. Il s'agit simplement de remplacer les séquents classiques par des séquents contraints.

$$\frac{T \vdash u_1 \ \llbracket E_1 \rrbracket \quad \ldots \quad T \vdash u_k \ \llbracket E_k \rrbracket}{T \vdash u\downarrow \ \ \llbracket E_1 \wedge \ldots \wedge E_k \rrbracket} \ \mathcal{S}$$

Toutefois, une telle règle ne répond pas aux conditions de notre calcul de séquents contraints qui impose le non partage de variables dans les hypothèses de la règle. Nous devons donc transformer de nouveau cette règle :

$$\frac{T \vdash u'_1 \ \llbracket E_1 \rrbracket \quad \ldots \quad T \vdash u'_k \ \llbracket E_k \rrbracket}{T \vdash u\downarrow \ \ \llbracket E_1 \wedge \ldots \wedge E_k \rrbracket} \ \mathcal{S}$$

$u'_1, \ldots, u'_k$ sont le résultat du processus de linéarisation sur $u_1, \ldots, u_k$ et R est l'ensemble des co-références. Plus précisément, pour chaque variable x apparaissant à deux positions distinctes dans les hypothèses, on remplace la seconde occurrence par une variable z n'apparaissant pas dans les hypothèses et on ajoute $x = z$ dans les co-références. Ce processus est itéré jusqu'au point fixe. Il n'est pas utile d'appliquer ce processus sur $E_1, \ldots, E_k$ car ce sont des variables distinctes.

Il est possible d'altérer légèrement le processus de linéarisation décrit ci-dessus de façon à choisir les positions à renommer de telle façon à ce que la conclusion soit toujours u.

Pourquoi linéariser les règles de composition et de décomposition ? Il est difficile d'expliquer à ce stade pourquoi il est nécessaire de linéariser ces règles. Nous allons tenter de donner ici une vague idée de la raison. Supposons que l'on dispose, dans le cas de Dolev-Yao, de preuves de $T \vdash \{a\}_x \ \ [\![x = b]\!]$ et de $T \vdash b \ [\![]\!]$. On voudrait pouvoir construire la preuve suivante :

$$\frac{T \vdash \{a\}_x \ \ [\![x = b]\!] \quad T \vdash b \ \ [\![]\!]}{T \vdash a \ \ [\![x = b]\!]}$$

Cependant, il ne s'agit pas de l'application exacte de la règle de déchiffrement. En effet, celle-ci ne prend normalement pas en compte la contrainte et donc ne peut pas prendre en compte le fait que $x = b$. La linéarisation des règles permet de contourner ce problème.

Toutefois, nous verrons plus loin (section 4.3.3 page 38) que nous allons doter l'intrus du pouvoir de transférer des informations de la contrainte vers le terme principal d'un séquent contraint, c'est-à-dire d'instancier des termes à l'aide de la contrainte. La preuve ci-dessus pourrait alors être réécrite de cette façon :

$$\frac{\dfrac{T \vdash \{a\}_x \ \ [\![x = b]\!]}{T \vdash \{a\}_b \ \ [\![x = b]\!]} \quad T \vdash b \ \ [\![]\!]}{T \vdash a \ \ [\![x = b]\!]}$$

Cependant, les instanciations portant sur des termes de profondeur trop importante nous poserons problème pour le théorème 10.1. Notamment, nous ne saurons pas normaliser une telle preuve :

$$\frac{\dfrac{T \vdash \{a\}_{\mathrm{pub}(x)} \ \ [\![x = b]\!]}{T \vdash \{a\}_{\mathrm{pub}(b)} \ \ [\![x = b]\!]} \quad T \vdash \mathrm{priv}(b) \ \ [\![]\!]}{T \vdash a \ \ [\![x = b]\!]}$$

La linéarisation nous évite d'avoir à recourir à des instanciations trop profondes. De plus, elle limite le nombre d'instanciations dans les preuves, ce qui permet de conserver le fait que la contrainte constitue une stratégie. Cette idée de conserver les contraintes à part, sans tenter de les résoudre ou d'effectuer des instanciations est similaire à ce que l'on peut trouver dans stratégies basiques [RV01].

4.3 Extension du pouvoir de l'intrus

Nous allons désormais étendre le pouvoir de l'intrus en lui ajoutant des règles supplémentaires :

- des règles lui permettant de faire jouer une règle de protocole,
- des règles lui permettant de récupérer les nonces des agents compromis,
- des règles lui permettant d'instancier des variables contenues dans les contraintes,
- une règle d'affaiblissement qui renforce la contrainte.

4.3.1 Règle de protocole

Il y a deux types de règles de protocole :
- les règles correspondant à l'ouverture d'une session,
- les règles correspondant à l'avancement d'une session.

Dans les deux cas, on introduit un nouvel ensemble de variables qui permettront d'indiquer la progression dans le protocole pour chaque session. Cet ensemble de variables est noté ξ et est disjoint des ensembles rencontrés jusqu'ici. Un membre de cet ensemble est noté $x_{k,s}$ où s est un numéro de session et k est l'étape du protocole jouée par l'acteur principal de la session pour le rôle correspondant.

Si $x_{k,s} = 1$, la règle k du rôle correspondant à la session s a été jouée. Pour simplifier la compréhension, on note également cette variable sous le nom $x_{R,k,s}$ où R est le rôle de la session s.

Voici la règle correspondant à une ouverture de session :

$$\frac{T \vdash u \ \llbracket E \rrbracket}{T \vdash w \downarrow \ \ \llbracket u = v \wedge E \wedge \sigma_s \wedge x_{R,1,s} = 1 \rrbracket} \mathcal{P}$$

Avec :

1. $v \Rightarrow w$ est la première règle du rôle R,
2. σ_s est l'ensemble des affectations initiales propres à la session s,
3. $E \not\models_{\mathcal{E}} x_{R,1,s} = 1$: s est un nouveau numéro de session.

L'intrus doit présenter un terme u unifiable avec v (modulo les équations de $\mathcal{E}$). Il obtient ainsi le terme w. On note que la règle 1 de la session s a été jouée et on ajoute dans les contraintes les affectations initiales du rôle.

Il est important de rappeler que les variables sont renommées de façon à être uniques à travers les différentes sessions. Ainsi, les variables introduites par l'égalité $u = v$ ne se retrouvent pas dans E. De façon rigoureuse, il faudrait écrire v_s et w_s, mais pour simplifier la notation, nous omettons les indices.

Pour l'avancement d'une session, nous introduisons la règle suivante :

$$\frac{T \vdash u \ \llbracket E \rrbracket}{T \vdash w \downarrow \ \ \llbracket u = v \wedge E \wedge x_{R,k,s} = 1 \rrbracket} \mathcal{P}$$

Avec :

1. $v \Rightarrow w$ est la règle numérotée $k > 1$ du rôle R
2. $E \not\models_{\mathcal{E}} x_{R,k,s} = 1$ et $E \models_{\mathcal{E}} x_{R,k-1,s} = 1$

Cette règle est très proche de la précédente. La différence majeure réside dans le fait que l'on vérifie que la règle précédente pour la session s a bien été jouée mais aussi que la règle que l'on tente de jouer n'a pas déjà été exécutée.

4.3.2 Agents compromis

Dans le système de transition, lors de l'ouverture d'une nouvelle session, l'intrus voyait ses connaissances enrichies des nonces de l'agent créant la session si celui-ci était compromis. Nous allons donc lui fournir une règle destinée à obtenir ce nonce à tout moment :

$$\frac{T \vdash u \ \ [\![E]\!]}{T \vdash N_i(s) \ \ [\![E]\!]} \mathcal{ND}$$

Avec :

1. $E \models_{\mathcal{E}} x_{R,1,s} = 1$,
2. l'acteur principal de la session s est compromis,
3. $N_i(s)$ est l'un des nonces généré à la création de la session s.

Le premier point permet de s'assurer que la session s a été ouverte, c'est-à-dire que le point de contrôle est au moins sur la première règle du rôle correspondant.

Cette règle est également indispensable pour éviter d'avoir à donner à l'intrus la possibilité de créer de nouveaux termes. C'est un point important, comme indiqué dans [DLMS99].

4.3.3 Instanciation

Les séquents contraints ont été introduits pour exprimer la stratégie de l'intrus. Toutefois, afin d'être complet vis-à-vis du modèle, nous allons autoriser dans un premier temps le transfert de données de la contrainte vers le terme.

Cela peut permettre à l'intrus d'instancier certaines variables apparaissant dans le terme pour appliquer, par exemple, une règle de déchiffrement.

On utilise alors cette règle d'instanciation :

$$\frac{T \vdash u \ \ [\![E]\!]}{T \vdash u\sigma \downarrow \ \ [\![\sigma]\!]} \mathcal{I}$$

Avec :

1. $\sigma \models_{\mathcal{E}} E$ et σ se limite aux variables de E,
2. pour tout $x \in \xi$, $x \in Var(\sigma)$ si et seulement si $x \in Var(E)$.

Autrement dit, σ est une solution quelconque de E et ne doit pas effectuer l'affectation d'une variable de contrôle (elles n'ont aucune raison d'apparaître en partie gauche).

4.3.4 Affaiblissement

Il peut être nécessaire pour l'intrus d'exécuter une des règles du protocole afin par exemple de pouvoir avoir accès à la suivante sans pour autant exploiter le résultat de la règle de protocole. Il est donc important de pouvoir faire progresser le point de contrôle sans se soucier des résultats de la règle de protocole.

Pour cela, on utilise la règle d'affaiblissement suivante :

$$\frac{T \vdash u_1 \ \ [\![E_1]\!] \quad T \vdash u_2 \ \ [\![E_2]\!]}{T \vdash u_1 \downarrow \ \ [\![E_1 \wedge E_2]\!]} \mathcal{W}$$

4.4 Exemples

Dans cette section, nous allons montrer deux exemples de preuve. La première est un exemple d'attaque sur le protocole de Needham-Schroeder (il s'agit de l'exemple classique).

Règle de $\mathcal{S}$.

$$\frac{T \vdash u'_1 \ \llbracket E_1 \rrbracket \quad \dots \quad T \vdash u'_k \ \llbracket E_k \rrbracket}{T \vdash u \downarrow \ \llbracket E_1 \wedge \dots \wedge E_k \rrbracket} \mathcal{S}$$

Ouverture de session . Si $v \Rightarrow w$ est la première règle du rôle R, σ_s est l'ensemble des affectations initiales propres à la session s et $E \not\models_{\mathcal{E}} x_{R,1,s} = 1$, alors :

$$\frac{T \vdash u \ \llbracket E \rrbracket}{T \vdash w \downarrow \ \llbracket u = v \wedge E \wedge \sigma_s \wedge x_{R,1,s} = 1 \rrbracket} \mathcal{P}$$

Avancement de session . Si $v \Rightarrow w$ est la règle numérotée $k > 1$ du rôle R, $E \not\models_{\mathcal{E}} x_{R,k,s} = 1$ et $E \models_{\mathcal{E}} x_{R,k-1,s} = 1$, alors :

$$\frac{T \vdash u \ \llbracket E \rrbracket}{T \vdash w \downarrow \ \llbracket u = v \wedge E \wedge x_{R,k,s} = 1 \rrbracket} \mathcal{P}$$

Agent compromis . Si $E \models_{\mathcal{E}} x_{R,1,s} = 1$ et l'acteur principal de la session s est compromis, alors :

$$\frac{T \vdash u \ \llbracket E \rrbracket}{T \vdash N_i(s) \ \llbracket E \rrbracket} \mathcal{ND}$$

Règle d'instanciation . Si $\sigma \models_{\mathcal{E}} E$, σ limité aux variables de E et $\forall x \in \xi$, $x \in Var(\sigma) \Leftrightarrow x \in Var(E)$, alors :

$$\frac{T \vdash u \ \llbracket E \rrbracket}{T \vdash u\sigma \downarrow \ \llbracket \sigma \rrbracket} \mathcal{I}$$

Règle d'affaiblissement .

$$\frac{T \vdash u_1 \ \llbracket E_1 \rrbracket \quad T \vdash u_2 \ \llbracket E_2 \rrbracket}{T \vdash u_1 \downarrow \ \llbracket E_1 \wedge E_2 \rrbracket} \mathcal{W}$$

FIG. 4.1 - *Règles de définition du système* $\widehat{\mathcal{S}}$

4.4.1 Exemple avec Needham-Schroeder

Pour mémoire, la définition informelle du protocole est la suivante :

$$\begin{array}{ll} A \to B & \{A, N_A\}_{K_B} \\ B \to A & \{N_A, N_B\}_{K_A} \\ A \to B & \{N_B\}_{K_B} \end{array}$$

Ce protocole admet une attaque simple décrite dans la table 4.1. Nous allons donner la preuve de cette attaque dans notre système de preuve. Le secret est le nonce N_B. Comme il s'agit aussi d'un protocole d'authentification, on va fabriquer le terme $\left\langle N_B, \{N_B\}_{K_B} \right\rangle$ ce qui nous permettra d'obtenir une attaque contre une forme faible de l'authentification.

$$\begin{array}{ll|ll} A \to I & \{A, N_A\}_{K_I} & & \\ & & I(A) \to B & \{A, N_A\}_{K_B} \\ & & B \to I(A) & \{N_A, N_B\}_{K_A} \\ I \to A & \{N_A, N_B\}_{K_A} & & \\ A \to I & \{N_B\}_{K_I} & & \\ & & I(A) \to B & \{N_B\}_{K_B} \end{array}$$

TAB. 4.1 - *Attaque sur Needham-Schroeder*

Formellement, le protocole est décrit par deux rôles : le premier rôle est appelé « Alice », le second « Bob ». Voici le rôle « Alice » :

$$\text{Alice} \begin{cases} & \lambda a, \lambda b, \nu n_a & & \\ 1. & & \Rightarrow & \{\langle a, n_a \rangle\}_{\text{pub}(b)} \\ 2. & \{\langle n_a, x \rangle\}_{\text{pub}(a)} & \Rightarrow & \{x\}_{\text{pub}(b)} \end{cases}$$

Et voici le rôle « Bob » :

$$\text{Bob} \begin{cases} & \lambda c, \nu n_c & & \\ 1. & \{\langle y, z \rangle\}_{\text{pub}(c)} & \Rightarrow & \{\langle z, n_c \rangle\}_{\text{pub}(y)} \\ 2. & \{n_c\}_{\text{pub}(c)} & \Rightarrow & \end{cases}$$

Les règles sont écrites de façon à éviter tout conflit dans le nommage des variables (d'où l'utilisation de c au lieu de b pour Bob).

On considère que $T = \left\{A, B, I, pub(I)^{-1}\right\}$. Voici une première partie de la preuve. E_1 est l'ensemble $a_1 = A \wedge b_1 = I$.

$$\dfrac{\dfrac{\dfrac{}{T \vdash \{\langle a_1, n_a(1) \rangle\}_{\text{pub}(b_1)} \ \ [\![E_1 \wedge x_{\text{Alice},1,1} = 1]\!]}\mathcal{P} \quad T \vdash \text{pub}(I)^{-1} \ \ [\![]\!]}{T \vdash \langle a_1, n_a(1) \rangle \ \ [\![E_1 \wedge x_{\text{Alice},1,1} = 1]\!]}\mathcal{D} \quad \dfrac{T \vdash B \ \ [\![]\!]}{T \vdash \text{pub}(B) \ \ [\![]\!]}\mathcal{C}}{T \vdash \{\langle a_1, n_a(1) \rangle\}_{\text{pub}(B)} \ \ [\![E_1 \wedge x_{\text{Alice},1,1} = 1]\!]}\mathcal{C}$$

L'intrus va passer le terme ainsi obtenu à B puis donner le terme obtenu de B à A, sans le modifier. Dans cette partie de l'exemple, la contrainte est enrichie d'équations provenant du *matching* entre deux termes lors des deux règles de protocole. On note E_1 l'ensemble $c_2 = B$.

$$\cfrac{\cfrac{T \vdash \{\langle a_1, n_a(1)\rangle\}_{\text{pub}(B)} \;\; [\![E_1 \wedge x_{\text{Alice},1,1} = 1]\!]}{T \vdash \{\langle z_2, n_c(2)\rangle\}_{\text{pub}(y_2)} \;\; [\![E_1 \wedge x_{\text{Alice},1,1} = 1 \wedge E_2 \wedge x_{\text{Bob},1,2} = 1 \wedge y_2 = a_1 \wedge z_2 = n_a(1)]\!]}\,\mathcal{P}}{T \vdash \{x_1\}_{\text{pub}(b_1)} \;\; [\![\ldots \wedge x_1 = n_c(2) \wedge x_{\text{Alice},2,1} = 1]\!]}\,\mathcal{P}$$

La contrainte finale est notée E. Il s'agit de :

$$\begin{aligned} a_1 = A \wedge b_1 = I \wedge c_2 = B \\ &\wedge x_{\text{Alice},2,1} = 1 \wedge x_{\text{Bob},1,2} = 1 \wedge x_{\text{Alice},1,1} = 1 \wedge y_2 = a_1 \wedge z_2 = n_a(1) \wedge x_1 = n_c(2) \end{aligned}$$

On peut alors obtenir le terme $n_c(2)$ en déchiffrant le terme obtenu en en l'instanciant :

$$\cfrac{\cfrac{T \vdash \{x_1\}_{\text{pub}(b_1)} \;\; [\![E]\!] \quad T \vdash \text{pub}(I)^{-1} \;\; [\![]\!]}{T \vdash x_1 \;\; [\![E]\!]}\,\mathcal{D}}{T \vdash n_c(2) \;\; [\![E]\!]}\,\mathcal{I}$$

La première règle est valide car $E \models b_1 = I$. On peut également obtenir le terme $\{n_c(2)\}_{\text{pub}(B)}$ en dupliquant la preuve ci-dessus et en la continuant :

$$\cfrac{T \vdash n_c(2) \;\; [\![E]\!] \quad \cfrac{T \vdash B \;\; [\![]\!]}{T \vdash \text{pub}(B) \;\; [\![]\!]}\,\mathcal{C}}{T \vdash \{n_c(2)\}_{\text{pub}(B)} \;\; [\![E]\!]}\,\mathcal{C}$$

La contrainte E répond aux conditions nécessaires sur l'honnêteté des agents. Il s'agit donc bien d'une attaque sur le secret du nonce $n_c(2)$.

L'attaque étant linéaire dans son déroulement, nous n'avons pas eu besoin d'utiliser une règle d'affaiblissement. Modifions légèrement les règles d'Alice pour pouvoir fabriquer un exemple utilisant la règle d'affaiblissement. Imaginons que suite au protocole, Alice désire transmettre un secret v à Bob. Elle attend simplement que Bob lui envoie son identité en clair. La difficulté pour l'attaquant est donc de faire parvenir Alice assez loin dans le protocole pour pouvoir lui faire jouer cette règle. On modifie le rôle d'Alice de la façon suivante :

$$\text{Alice}\begin{cases} & \lambda a, \lambda b, \nu n_a, \nu v & & \\ 1. & & \Rightarrow & \{\langle a, n_a\rangle\}_{\text{pub}(b)} \\ 2. & \{\langle n_a, x\rangle\}_{\text{pub}(a)} & \Rightarrow & \{x\}_{\text{pub}(b)} \\ 3. & b & \Rightarrow & v \end{cases}$$

Nous sommes donc intéressés par faire avancer Alice au point de contrôle 2, mais pas par le résultat de la règle de protocole. Une preuve possible pour l'attaque est alors la suivante :

$$\cfrac{\cfrac{T \vdash B \;\; [\![]\!] T \vdash \{x_1\}_{\text{pub}(b_1)} \;\; [\![E]\!]}{T \vdash B \;\; [\![E]\!]}\,\mathcal{W}}{T \vdash v(1) \;\; [\![E \wedge x_{\text{Alice},3,1} = 1]\!]}\,\mathcal{P}$$

Le point important de la preuve est que l'application de la règle de protocole n'est possible que si $x_{\text{Alice},2,1} = 1$ fait partie de la contrainte, ce qui explique l'intérêt de la règle d'affaiblissement.

Il existe une attaque beaucoup plus simple qui consiste à laisser A et B jouer ensemble. Si Alice chiffrait son secret avec le nonce échangé par B, une telle attaque ne serait pas possible, mais l'attaque décrite ci-dessus serait toujours valable.

4.4.2 Exemple avec Needham-Schroeder modifié

Pour parer à l'attaque précédente, Lowe a proposé de rajouter l'identité de Bob dans le second message [Low96]. Une variation est d'inclure l'identité de Bob en effectuant un « ou exclusif » avec le nonce envoyé par Alice :

$$\begin{array}{ll} A \to B & \{A, N_A\}_{K_B} \\ B \to A & \{N_A \oplus B, N_B\}_{K_A} \\ A \to B & \{N_B\}_{K_B} \end{array}$$

L'attaque précédente n'est alors plus possible. Toutefois, si on se place dans un modèle prenant en compte les propriétés du « ou exclusif », il existe l'attaque suivante présentée dans la table 4.2.

$$\begin{array}{ll|ll} A \to I & \{A, N_A\}_{K_I} & & \\ & & I(A) \to B & \{A, N_A \oplus B \oplus I\}_{K_B} \\ & & B \to I(A) & \{N_A \oplus B \oplus I \oplus B, N_B\}_{K_A} \\ I \to A & \{N_A \oplus B \oplus I \oplus B, N_B\}_{K_A} & & \\ A \to I & \{N_B\}_{K_I} & & \\ & & I(A) \to B & \{N_B\}_{K_B} \end{array}$$

TAB. 4.2 - *Attaque sur Needham-Schroeder modifié*

L'attaque réside dans le fait que $N_A \oplus B \oplus I \oplus B = N_A \oplus I$. La différence par rapport à la section précédente, pour notre exemple, est que nous avons désormais une théorie équationnelle. Il faudra donc veiller à la bonne application de l'opérateur de normalisation.

Voici les règles du rôle d'Alice :

$$\text{Alice}\left\{\begin{array}{lrcl} & \lambda a, \lambda b, \nu n_a & & \\ 1. & & \Rightarrow & \{\langle a, n_a\rangle\}_{\text{pub}(b)} \\ 2. & \{\langle n_a \oplus b, x\rangle\}_{\text{pub}(a)} & \Rightarrow & \{x\}_{\text{pub}(b)} \end{array}\right.$$

Et voici les nouvelles règles pour Bob :

$$\text{Bob}\left\{\begin{array}{lrcl} & \lambda c, \nu n_c & & \\ 1. & \{\langle y, z\rangle\}_{\text{pub}(c)} & \Rightarrow & \{\langle z \oplus c, n_c\rangle\}_{\text{pub}(y)} \\ 2. & \{n_c\}_{\text{pub}(c)} & \Rightarrow & \end{array}\right.$$

Comme précédemment, on considère que $T = \{A, B, I, pub(I)^{-1}\}$. Avec E_1 l'ensemble $a_1 = A \wedge b_1 = I \wedge x_{\text{Alice},1,1} = 1$ et E_2 l'ensemble $c_2 = B \wedge x_{\text{Bob},1,2} = 1$, la première partie de la preuve est présentée dans la table 4.3 page ci-contre.

$$
\cfrac{
  \cfrac{
    T \vdash \langle A, n_a(1) \oplus B \oplus I \rangle \;\; \llbracket E_1 \rrbracket
  }{
  }
}{}
$$

$$
\dfrac{
  \dfrac{
    \dfrac{
      T \vdash A \;\; \llbracket\rrbracket
      \qquad
      \dfrac{
        \dfrac{
          \dfrac{
            \dfrac{}{T \vdash \{\langle a_1, n_a(1)\rangle\}_{\mathrm{pub}(b_1)} \;\; \llbracket E_1 \rrbracket}\,\mathcal{P}
            \qquad
            T \vdash \mathrm{pub}(I)^{-1} \;\; \llbracket\rrbracket
          }{
            T \vdash \langle a_1, n_a(1)\rangle \;\; \llbracket E_1 \rrbracket
          }\,\mathcal{D}
        }{
          T \vdash n_a(1) \;\; \llbracket E_1 \rrbracket
        }\,\mathcal{D}
        \qquad
        T \vdash B \;\; \llbracket\rrbracket
        \quad
        T \vdash I \;\; \llbracket\rrbracket
      }{
        T \vdash n_a(1) \oplus B \oplus I \;\; \llbracket E_1 \rrbracket
      }\,\mathcal{C}
    }{
      T \vdash \langle A, n_a(1) \oplus B \oplus I \rangle \;\; \llbracket E_1 \rrbracket
    }\,\mathcal{C}
    \qquad
    \dfrac{T \vdash B \;\; \llbracket\rrbracket}{T \vdash \mathrm{pub}(B) \;\; \llbracket\rrbracket}\,\mathcal{C}
  }{
    \dfrac{
      \dfrac{
        T \vdash \{\langle A, n_a(1) \oplus B \oplus I \rangle\}_{\mathrm{pub}(B)} \;\; \llbracket E_1 \rrbracket
      }{
        T \vdash \{\langle z_2 \oplus c_2, n_c(2)\rangle\}_{\mathrm{pub}(y_2)} \;\; \llbracket E_1 \wedge E_2 \wedge y_2 = a_1 \wedge z_2 = n_a(1) \oplus B \oplus I \rrbracket
      }\,\mathcal{P}
    }{
      T \vdash \{x_1\}_{\mathrm{pub}(b_1)} \;\; \llbracket \ldots \wedge x_1 = n_c(2) \wedge x_{\mathrm{Alice},2,1} = 1 \rrbracket
    }\,\mathcal{P}
  }\,\mathcal{C}
}{}
$$

TAB. 4.3 - *Première partie de la preuve d'attaque de NS avec* $\oplus$

La deuxième partie de la preuve ne change pas par rapport à la section précédente. On remarque que le traitement de $n_a(1) \oplus B \oplus I$ est effectué de manière transparente par la règle de protocole via la procédure de matching entre u et v : la dernière règle de protocole a pu être appliquée en résolvant les équations suivantes :

$$z_2 \oplus c_2 = n_a(1) \oplus b_1 \wedge z_2 = n_a(1) \oplus B \oplus I \wedge c_2 = B \wedge b_1 = I$$

CHAPITRE 5

Complétude et correction

Nous allons voir dans ce chapitre que le modèle présenté au chapitre 4 est correct et complet vis-à-vis du système de transition et de la propriété de secret présentés dans le chapitre 3.

5.1 Complétude

Le premier lemme montre qu'une attaque dans le système de transition se retrouve dans le système de preuve $\widehat{\mathcal{S}}$.

Lemme 5.1 (Complétude de $\widehat{\mathcal{S}}$) *Pour tout état q accessible et pour tout terme t, si $\mathcal{I}_q \vdash_{\mathcal{S}} t$, alors on peut construire une preuve Π dans $\widehat{\mathcal{S}}$ de $\mathcal{I}_0 \vdash t \downarrow \ \llbracket E \rrbracket$ de telle façon que $\sigma_q \models_{\mathcal{E}} E$.*

Preuve.

On dit qu'*un état q a atteint l'étape k dans une session s* quand il existe b, θ et $k' \geq k$ tels que $M_{q,b}(s) = (R, k', \theta)$. Comme noté précédemment, q, s permettent de déterminer de manière unique b, R et θ.

Nous allons procéder par induction sur le couple (n, m) où n est la longueur du chemin vers l'état q et m la taille de la preuve de $\mathcal{I}_q \vdash_{\mathcal{S}} t$ dans $\mathcal{S}$. On montre l'invariant suivant :

> Pour tout état q accessible en moins de n étapes, pour tout terme t, si $\mathcal{I}_q \vdash_{\mathcal{S}} t$, alors,on peut construire une preuve Π de $\mathcal{I}_0 \vdash t \downarrow \ \llbracket E \rrbracket$ telle que $\sigma_q \models_{\mathcal{E}} E$ et pour tout triplet R, k, s, $(E \models x_{R,k,s} = 1)$ si et seulement si l'état q a atteint l'étape k dans la session s.

Si $(n, m) = (0, 0)$, q est l'état initial et $t \in \mathcal{I}_0$. On peut alors considérer la preuve de $\mathcal{I}_0 \vdash t \downarrow \ \llbracket\rrbracket$ réduite à la règle d'axiome. σ_q étant vide, l'invariant est vérifié.

Si $m > 0$, on considère la dernière règle de la preuve de $\mathcal{I}_q \vdash_{\mathcal{S}} t$: $\mathcal{I}_q \vdash_{\mathcal{S}} t_1, \ldots, \mathcal{I}_q \vdash_{\mathcal{S}} t_r$ et t peut être déduit de $t_1, \ldots, t_r$ en une étape. On applique l'hypothèse de récurrence sur chaque t_i : il existe des preuves Π_i dans $\widehat{\mathcal{S}}$ de $\mathcal{I}_0 \vdash t_i \downarrow \ \llbracket E_i \rrbracket$ telles que $\sigma_q \models_{\mathcal{E}} E_i$ et $E_i \models x_{R,k,s} = 1$ si et seulement si l'état q a atteint l'étape k dans la session s.

Considérons alors la preuve suivante de $t \downarrow$:

$$\dfrac{\dfrac{\Pi_1}{\mathcal{I}_0 \vdash t_1\downarrow \ \ [\![E_1]\!]} \ \ \ldots \ \ \dfrac{\Pi_r}{\mathcal{I}_0 \vdash t_r\downarrow \ \ [\![E_r]\!]}}{\mathcal{I}_0 \vdash t\downarrow \ \ [\![E_1 \wedge \ldots \wedge E_r]\!]}$$

On a $\sigma_q \models_{\mathcal{E}} E_1 \wedge \ldots \wedge E_r$ car σ_q satisfait individuellement chaque ensemble d'équations. De plus, le fait d'atteindre l'étape k dans la session s pour q est indépendant de i et donc $E_1 \wedge \ldots \wedge E_r \models x_{R,k,s} = 1$ si et seulement si l'état q a atteint l'étape k dans la session s.

Si $n > 0$ et que $m = 0$, soit q' un état acessible en $n-1$ étapes depuis l'état initial et de telle façon que q soit accessible de q' en une seule étape. Soit s la session concernée par la transition, R le rôle, a l'acteur principal, σ les affectations initiales propres à la session et $u \Rightarrow v$ la règle en position k dans le rôle R. $M_{q,a}(s) = (R, k, \sigma \uplus \theta)$ et pour tout couple $(a', s') \neq (a, s)$, $M_{q,a'}(s') = M_{q',a'}(s')$. De plus, u est vide ou alors $\mathcal{I}_{q'} \vdash_{\mathcal{S}} u\theta\downarrow$.

Comme $m = 0$, $t \in \mathcal{I}_q$. En utilisant l'invariant, $\mathcal{I}_{q'} \vdash_{\mathcal{S}} u\theta\downarrow$ implique qu'il existe une preuve $\Pi_{u\theta}$ dans $\widehat{\mathcal{S}}$ de $\mathcal{I}_0 \vdash u\theta\downarrow \ \ [\![E_{u\theta}]\!]$ telle que $\sigma_{q'} \models_{\mathcal{E}} E_{u\theta}$ et $E_{u\theta} \models x_{R',k',s'} = 1$ si et seulement si l'état q' a atteint l'étape k' dans la session s'.

Soit Π_v la preuve suivante :

$$\dfrac{\dfrac{\Pi_{u\theta}}{\mathcal{I}_0 \vdash u\theta\downarrow \ \ [\![E_{u\theta}]\!]}}{\mathcal{I}_0 \vdash v\downarrow \ \ [\![E_{u\theta} \wedge u\theta\downarrow = u \wedge \sigma_s \wedge x_{R,k,s} = 1]\!]}\ \mathcal{P}$$

Si $k > 1$, σ_s est vide. Dans le cas contraire, il s'agit des affectations initiales pour la session k. Cette preuve est bien dans $\widehat{\mathcal{S}}$: $\mathcal{P}$ est soit une règle d'ouverture de session ou d'avancement de session. Si u est vide, on procède comme ci-dessus en supprimant $\Pi_{u\theta}$, $E_{u\theta}$ et les équations impliquant le terme u.

Si $t \in \mathcal{I}_{q'}$, l'invariant nous indique qu'il existe une preuve Π_0 de $\mathcal{I}_0 \vdash t\downarrow \ \ [\![E_0]\!]$ dans $\widehat{\mathcal{S}}$ de telle façon que $\sigma_{q'} \models E_0$ et pour tout R', k', s', $E_0 \models_{\mathcal{E}} x_{R',k',s'} = 1$ si et seulement si l'état q' a atteint l'étape k' dans s'.

Considérons la preuve suivante de $t\downarrow$:

$$\Pi = \dfrac{\dfrac{\Pi_0}{\mathcal{I}_0 \vdash t\downarrow \ \ [\![E_0]\!]} \quad \dfrac{\Pi_v}{\mathcal{I}_0 \vdash v\downarrow \ \ [\![E_v]\!]}}{\mathcal{I}_0 \vdash t\downarrow \ \ [\![E_0 \wedge E_v]\!]}\ \mathcal{W}$$

E_v est la conjonction $E_{u\theta} \wedge u\theta\downarrow = u \wedge \sigma_s \wedge x_{R,k,s} = 1$. On a alors $\sigma_q \models E_0 \wedge E_v$.

Pour tout triplet R', k', s', q a atteint l'étape k' dans la session s' si et seulement si q' a atteint l'étape k' dans la session s' et dans ce cas, par l'invariant, $E_0 \models x_{R',k',s'} = 1$, ou bien $k = k'$ et $s = s'$ et dans ce cas, $E_v \models x_{R,k,s} = 1$.

Inversement, si $E_0 \wedge E_v \models x_{R',k',s'} = 1$, q' a atteint l'étape k' dans la session s' ou $k' = k$ and $s' = s$. Dans les deux cas, q a atteint l'étape k' dans la session s'. Ainsi, Π satisfait les hypothèse demandées.

Plaçons-nous maintenant dans le cas où $t \notin \mathcal{I}_{q'}$. Par définition du système de transition, $t = v\theta\downarrow$ ou $t = N_i(s)$ (si la dernière étape est une ouverture de session dont l'acteur principal est compromis). Considérons le premier cas ; on construit la preuve Π de la façon suivante :

$$\dfrac{\dfrac{\Pi_v}{\mathcal{I}_0 \vdash v \downarrow \;\; [\![E_{u\theta} \wedge u\theta \downarrow = u \wedge \sigma_s \wedge x_{R,k,s} = 1]\!]}}{\mathcal{I}_0 \vdash v\theta \downarrow \;\; [\![E_{v\theta}]\!]}\,\mathcal{I}$$

Cette preuve est dans $\widehat{\mathcal{S}}$: $\theta \models_{\mathcal{E}} u\theta \downarrow = u$, donc il existe une extension commune de θ et σ_q qui satisfait les contraintes et dont la restriction aux variables de v coïncide avec θ. Π satisfait alors l'invariant.

Considérons ensuite le cas $t = N_i(s)$. On choisit alors Π de la façon suivante :

$$\dfrac{\dfrac{\Pi_v}{\mathcal{I}_0 \vdash v \downarrow \;\; [\![E_v]\!]}}{\mathcal{I}_0 \vdash N_i(s) \;\; [\![E_v]\!]}\,\mathcal{N}$$

□

5.2 Correction

Le second lemme indique que si l'on trouve une attaque sur le secret dans le système de preuve $\widehat{\mathcal{S}}$, alors on peut obtenir une attaque du même secret dans le système de transition du chapitre 3.

Lemme 5.2 (Correction de $\widehat{\mathcal{S}}$) *Soit Π une preuve dans $\widehat{\mathcal{S}}$ de $\mathcal{I}_0 \vdash t$ $[\![E]\!]$. Pour toute substitution σ telle que $\sigma \models_{\mathcal{E}} E$, il existe un état accessible q tel que $\mathcal{I}_q \vdash_{\mathcal{S}} t\sigma \downarrow$.*

Preuve.

Pour $\sigma \models_{\mathcal{E}} E$, on considère σ_t, sa restriction aux variables de t, et σ_c, sa restriction aux variables de ξ (les $x_{R,k,s}$).

Pour toutes les règles d'inférence d'une preuve dans $\widehat{\mathcal{S}}$, on notera que la contrainte de la conclusion implique chacune des contraintes de chacune des prémises. Cela signifie entre autres qu'un σ qui satisfait la contrainte de la conclusion satisfait également les contraintes de chacune des prémises.

Notons également que si $x_{R,k,s} \in Var(E)$ et E est la contrainte de la conclusion de la preuve Π, alors Π contient forcément une règle correspondant à l'ouverture de la session s. Le seul piège peut résider dans les règles d'instanciation, mais on impose que dans ce cas σ n'introduit pas de telles variables de contrôle. Ainsi, seule la règle d'ouverture de session et la règle de suite de session peut introduire de telles variables.

On prouve par récurrence sur la taille de Π le résultat suivant :

> Pour tout σ tel que $\sigma \models_{\epsilon} E$, il existe un état q accessible tel que $\mathcal{I}_q \vdash_{\mathcal{S}} t\sigma_t \downarrow$ et $\sigma = \sigma_c \uplus \sigma_q \uplus \sigma_j$ pour un certain σ_j.

Cela signifie que la restriction de σ aux variables apparaissant dans le protocole est la substitution σ_q.

Dans le cas de base, Π est réduit à une règle d'axiome. $t \in \mathcal{I}_0$ et E est vide. Ainsi, il suffit de prendre $q = q_0$. σ peut être n'importe quelle substitution. Celle-ci peut toujours être décomposée en $\sigma_c \uplus \sigma_{q_0} \uplus \sigma_j$ car σ_{q_0} est vide.

On considère désormais que Π n'est pas réduit à une règle d'axiome. On considère alors la dernière règle.

Règle d'instanciation

$$\frac{\mathcal{I}_0 \vdash u \;\; [\![E]\!]}{\mathcal{I}_0 \vdash u\theta \downarrow \;\; [\![\theta]\!]}$$

si $\theta \models_{\mathcal{E}} E$.

On utilise l'hypothèse de récurrence pour dire qu'il existe un état q' tel que pour toute substitution σ', si $\sigma' \models_{\mathcal{E}} E$, alors $\mathcal{I}_{q'} \vdash_{\mathcal{S}} u\sigma'_u \downarrow$ et $\sigma' = \sigma'_c \uplus \sigma'_{q'} \uplus \sigma'_j$. Prenons σ une substitution telle que $\sigma \models_{\mathcal{E}} \theta$. Dans ce cas, $\sigma \models_{\mathcal{E}} E$ et on peut choisir $\sigma' = \sigma$. Pour toute variable x, $\sigma \models_{\mathcal{E}} x = x\theta$ et donc $x\sigma =_{\mathcal{E}} x\theta\sigma \downarrow$, ce qui signifie $u\theta\sigma \downarrow =_{\mathcal{E}} u\sigma \downarrow$ et donc $\mathcal{I}_{q'} \vdash_{\mathcal{S}} u\theta\sigma \downarrow$. Ensuite, soit $q = q'$: $\mathcal{I}_q \vdash_{\mathcal{S}} u\theta\sigma_{u\theta} \downarrow$ et $\sigma = \sigma_c \uplus \sigma_q \uplus \sigma_j$.

Règles de l'intrus (issues de $\mathcal{S}$)

$$\frac{\mathcal{I}_0 \vdash u_1 \;\; [\![E_1]\!] \quad \ldots \quad \mathcal{I}_0 \vdash u_k \;\; [\![E_k]\!]}{\mathcal{I}_0 \vdash u \downarrow \;\; [\![E_1 \wedge \ldots \wedge E_k]\!]}$$

On note E pour $E_1 \wedge \ldots \wedge E_k$. On applique l'hypothèse de récurrence sur chacune des prémises $T \vdash u_i \;\; [\![E_i]\!]$. Pour chaque i, il y a donc un état atteignable q_i tel que, pour tout σ_i qui satisfait E_i, $\mathcal{I}_{q_i} \vdash_{\mathcal{S}} u_i\sigma_i \downarrow$ et $\sigma_i = \sigma_c^i \uplus \sigma_{q_i} \uplus \sigma_j^i$. Considérons une substitution σ telle que $\sigma \models E_1 \wedge \ldots \wedge E_k$: pour tout i, $\sigma \models E_i$. On choisit alors pour chaque i $\sigma_i = \sigma$. Cela nous donne $\sigma = \sigma_c \uplus \sigma_{q_i} \uplus \sigma_j^i$.

Montrons par récurrence sur la longueur N de la séquence de transitions de q_0 à q_1 que s'il y a deux états accessibles q_1 et q_2 tels que $\sigma = \sigma_c \uplus \sigma_{q_1} \uplus \sigma_1 = \sigma_c \uplus \sigma_{q_2} \uplus \sigma_2$, alors on peut trouver un troisième état accessible q_3 tel que $\sigma = \sigma_c \uplus \sigma_{q_3} \uplus \sigma_3$, σ_{q_3} étende à la fois σ_{q_1} et σ_{q_2} et $\mathcal{I}_{q_1} \cup \mathcal{I}_{q_2} \subseteq \mathcal{I}_{q_3}$.

On peut comparer cette propriété à une certaines propriété de confluence ou à une propriété de monotonie : « tout ce qui était possible dans tel état a été joué ou reste possible dans les états suivants ».

Si $N = 0$, alors, on peut choisir $q_3 = q_2$. Sinon, prenons $q'_1 \to q_1$ comme étant la dernière transition de la séquence de q_0 à q_1. On utilise l'hypothèse de récurrence qui nous indique qu'il existe un état q'_3 tel que $\sigma = \sigma_c \uplus \sigma_{q'_3} \uplus \sigma'_3$ et $\sigma_{q'_3}$ étende σ_{q_2} et $\sigma_{q'_1}$ et $\mathcal{I}_{q'_1} \cup \mathcal{I}_{q_2} \subseteq \mathcal{I}_{q'_3}$.

Disons que la transition $q'_1 \to q_1$ est une application de la règle en position k de la session s qui correspond au rôle R : $M_{q_1,a}(s) = (R, k, \sigma'_1 \uplus \theta_1)$ et $M_{q'_1,a}(s)$ est soit indéfini, soit égal à $(R, k-1, \sigma'_1)$. Dans le cas de la suite d'une session, $\sigma_{q_1} = \sigma_{q'_1} \uplus \theta_1$ et dans le cas d'une ouverture de session, $\sigma_{q_1} = \sigma_{q'_1} \uplus \sigma'_1 \uplus \theta_1$.

Si $M_{q'_3,a}(s)$ est défini et équal à (R, k', θ'), comme σ étend à la fois θ' et σ_{q_1}, θ' doit être compatible avec σ_{q_1} en ce qui concerne les valeurs des affectations initiales de la session (en particulier θ' étend σ'_1).

De plus, si $k' \geq k$, toutes les variables du domaine de θ_1 sont également dans le domaine de θ'. θ_1 est alors une restriction de θ' puisque ces deux substitutions doivent coïncider avec σ sur leurs domaines. Enfin, $\sigma_{q'_1}$ est une restiction de $\sigma_{q'_3}$ (cf plus haut) et donc σ_{q_1} est une restriction de $\sigma_{q'_3}$ et on peut choisir $q_3 = q'_3$.

Maintenant, soit $M_{q'_3,a}(s)$ n'est pas défini, soit il est égal à (R, k', θ') avec $k' < k$. On peut alors appliquer la règle en position k de la session s qui correspond au rôle R à partir de l'état q'_3. En fait, $\mathcal{I}_{q'_1} \vdash_{\mathcal{S}} u\theta_1 \downarrow$ implique que $\mathcal{I}_{q'_3} \vdash_{\mathcal{S}} u\theta_1 \downarrow$.

Nous avons donc un état q_3 tel que $\mathcal{I}_{q_3}$ contient à la fois $\mathcal{I}_{q_1}$ et $\mathcal{I}_{q'_3}$. De plus, σ_{q_3} étend $\sigma_{q'_3}, \sigma'_1, \theta_1$ et donc étend à lafois σ_{q_1} et σ_{q_2}.

Appliquons cette nouvelle propriété sur les états $q_1, \ldots, q_k$. Par récurrence sur k, il existe un état q tel que $\mathcal{I}_q$ contient $\mathcal{I}_{q_1}, \ldots, \mathcal{I}_{q_k}$ et tel que σ_q étend $\sigma_{q_1}, \ldots, \sigma_{q_k}$ et pour un certain j, $\sigma = \sigma_q \uplus \sigma_c \uplus \sigma_j$. On a alors $\mathcal{I}_{q_i} \vdash_{\mathcal{S}} u_i\sigma \downarrow$ et donc pour tout i $\mathcal{I}_q \vdash_{\mathcal{S}} u_i\sigma \downarrow$. On peut alors compléter la preuve en appliquant cette règle :

$$\frac{\mathcal{I}_q \vdash u_1\sigma \downarrow \quad \ldots \quad \mathcal{I}_q \vdash u_k\sigma \downarrow}{\mathcal{I}_q \vdash u\sigma \downarrow}$$

Règle d'affaiblissement

$$\frac{\mathcal{I}_0 \vdash u_1 \ \llbracket E_1 \rrbracket \quad \mathcal{I}_0 \vdash u_2 \ \llbracket E_2 \rrbracket}{\mathcal{I}_0 \vdash u_1 \downarrow \ \llbracket E_1 \wedge E_2 \rrbracket} \mathcal{W}$$

Soit $\sigma \models_{\mathcal{E}} E_1 \wedge E_2$ et $\sigma \models E_1$. On utilise alors l'hypothèse de récurrence pour trouver un état q tel que $\mathcal{I}_q \vdash_{\mathcal{S}} u_1\sigma \downarrow$ et $\sigma = \sigma_c \uplus \sigma_q \uplus \sigma_j$.

Ouverture de session

$$\frac{\mathcal{I}_0 \vdash u \ \llbracket E \rrbracket}{\mathcal{I}_0 \vdash w \downarrow \ \llbracket u = v \wedge E \wedge \sigma_s \wedge x_{R,1,s} = 1 \rrbracket}$$

Soit $\sigma \models_{\mathcal{E}} u = v \wedge E \wedge \sigma_s \wedge x_{R,1,s} = 1$. En particulier, on a donc $\sigma \models E$. On utilise l'hypothèse de récurrence pour trouver un état q_1 tel que $\mathcal{I}_{q_1} \vdash_{\mathcal{S}} u\sigma \downarrow$ et $\sigma = \sigma_c \uplus \sigma_{q_1} \uplus \sigma_j^1$. Deux cas se présentent alors. Soit σ_{q_1} contient déjà les affectations initiales pour la session s et les affectations des variables de u et dans ce cas, on se contente de choisir $q = q_1$: $\mathcal{I}_{q_1}$ contient $w\sigma \downarrow$.

Dans le cas contraire, il existe une transition démarrant de q_1, ouvrant la session s et telle que les affectations initiales de la session s sont compatibles avec σ. On obtient alors un état q tel que $w\sigma \downarrow \in \mathcal{I}_q$ et $\sigma_q = \sigma_c \uplus \sigma_q \uplus \sigma_j$.

Progression de session Il s'agit d'un cas similaire dans lequel il n'y pas d'affectations initiales pour la session.

Compromission d'un agent

$$\frac{\mathcal{I}_0 \vdash u \ \llbracket E \rrbracket}{\mathcal{I}_0 \vdash N_i(s) \ \llbracket E \rrbracket}$$

Soit $\sigma \models_{\mathcal{E}} E$. Comme on a pu appliquer la règle de l'agent compromis, il existe un triplet R, k, s tel que $E \models x_{R,k,s} = 1$ avec l'acteur principal de la session s compromis. Cela signifie qu'il y a en amont dans la preuve une ouverture de la session s attachée au rôle R. Soit Π' la sous-preuve de Π qui se termine par une telle règle, avec comme conclusion $\mathcal{I}_0 \vdash w \ \llbracket E' \wedge u = v \wedge \sigma_s \wedge x_{R,1,s} = 1 \rrbracket$.

σ étant un modèle de E, il satisfait aussi la contrainte d'une telle conclusion. On applique donc l'hypothèse de récurrence sur Π'. On obtient un état q tel que $\mathcal{I}_q \vdash_{\mathcal{S}} w\sigma \downarrow$ et $\sigma = \sigma_c \uplus \sigma_q \uplus \sigma_j$.

De plus, par définition du système de transition, on a $N_i(s) \in \mathcal{I}_q$ et donc $\mathcal{I}_q \vdash_{\mathcal{S}} N_i(s)$.

□

En ce qui concerne la propriété de secret définie dans le chapitre 3, on peut donc utiliser indifféremment le système de transition ou le système de preuves $\widehat{\mathcal{S}}$ introduit lors du chapitre 4 pour mener la recherche d'attaque.

Lemme 5.3 *Il y a une attaque sur le secret de s si et seulement si il existe une preuve de $T \vdash s' \ \llbracket E \rrbracket$ telle que $s =_{\mathcal{E}} s'\sigma \downarrow$ et $\sigma \models_{\mathcal{E}} E$.*

Avant de passer au théorème de normalisation, nous allons définir dans le chapitre suivant un système de preuve $\overline{S}$ équivalent au système $\widehat{S}$ vis-à-vis du secret, mais dont les contraintes ont une forme plus sympathique. Cela permet de travailler avec une règle d'instanciation purement syntaxique puis par la suite de normaliser plus aisément les contraintes.

CHAPITRE 6

Restriction du modèle

Sommaire

Le modèle présenté dans le chapitre 4 page 33 présente quelques défauts gênants :

- la règle d'instanciation est complexe et particulièrement non déterministe et
- les contraintes sont des équations arbitraires pouvant parfois avoir plusieurs solutions incompatibles.

Les exemples présentés à la section 4.4 page 38 n'utilisent que des instanciations relativement simples et les contraintes sont présentées sous la forme d'un système résolu. Nous allons montrer comme restreindre le système de preuve $\widehat{\mathcal{S}}$ en un système de preuve $\overline{\mathcal{S}}$ ayant des caractéristiques similaires.

Dans ce chapitre, nous nous restreignons à des théories équationnelles $\mathcal{E}$ pour les quelles nous montrons que le système $\overline{\mathcal{S}}$ plus simple est correct et complet.

6.1 La propriété des variants finis

La propriété des variants finis va nous permettre de réduire certaines théories équationnelles à la théorie vide ou à AC. Pour se faire, nous allons remplacer certaines règles par un ensemble fini de règles sur lesquelles les normalisations ont été pré-calculées.

6.1.1 Définition

Définition 6.1 (Propriété des variants finis) *Une théorie équationnelle $\mathcal{E}$ respecte la propriété des variants finis si et seulement si elle peut s'écrire $\mathcal{E}_1 \uplus \mathcal{E}_2$ avec :*

1. *Les classes d'équivalence modulo $\mathcal{E}_2$ sont finies ;*
2. *$\mathcal{E}_1$ peut être transformée en un système de réécriture convergent modulo $\mathcal{E}_2$ et fini ; on note alors $t\downarrow$ la forme normale du terme t ;*
3. *pour tout terme t, il est possible de calculer un ensemble fini de termes $t_1, \ldots, t_k$ tels que $\{t\sigma\downarrow \mid \sigma \in \Sigma\} = \bigcup_{i=1}^{k}\{t_i\sigma \mid \sigma \in \Sigma\}$.*

En pratique, $\mathcal{E}_2$ sera soit vide, soit la théorie AC. $\mathcal{E}_1$ sera traduit en règles supplémentaires dans notre modèle. Cette propriété entraîne de plus que $\mathcal{E}$ est finitaire : toute équation a un nombre fini de « most general unifiers ».

Cette propriété est valable pour un grand nombre de théories intéressantes comme cela a été montré dans [CLD05] :

Lemme 6.1 ([CLD05]) *Les théories équationnelles **DYT** (Dolev-Yao), **KIT** (théorie de la clef inverse), **XOR** (ou exclusif), **AG** (groupes abéliens), **Hom** (homomorphisme) ont la propriété des variants finis.*

Exemple 6.1 (DYT) *La théorie vide se décompose en deux théories vides. $\mathcal{E}_1 = \mathcal{E}_2 = \emptyset$.*

Exemple 6.2 (KIT) *La théorie de la clef inverse peut se décomposer en $\mathcal{E}_1 \uplus \mathcal{E}_2$ avec $\mathcal{E}_2 = \emptyset$ et $\mathcal{E}_1$ l'ensemble de règles suivant :*

$$\begin{array}{rclrcl}
\mathsf{dec}(\{x\}_y, y^{-1}) & \to & x & \{\mathsf{dec}(x,y)\}_{y^{-1}} & \to & x \\
\mathsf{dec}(\{x\}_{y^{-1}}, y) & \to & x & \{\mathsf{dec}(x,y^{-1})\}_y & \to & x \\
(x^{-1})^{-1} & \to & x & & &
\end{array}$$

Exemple 6.3 (XOR) *La théorie du « ou exclusif » se décompose en $\mathcal{E}_2 = AC$ et $\mathcal{E}_1$ étant l'ensemble de règles suivant :*

$$\begin{array}{rcl}
x \oplus 0 & \to & x \\
x \oplus x & \to & 0 \\
x \oplus x \oplus y & \to & y
\end{array}$$

Exemple 6.4 (AG) *L'exemple des groupes abéliens est plus complexe. $\mathcal{E}_2$ est toujours AC mais $\mathcal{E}_1$ est le système de réécriture suivant (cf [CLD05] pour plus de détails) :*

$$\begin{array}{rclrcl}
x * 1 & \to & x & x^{-1^{-1}} & \to & x \\
1^{-1} & \to & 1 & (x^{-1} * y)^{-1} & \to & x * y^{-1} \\
x * x^{-1} & \to & 1 & x * (x^{-1} * y) & \to & y \\
x^{-1} * y^{-1} & \to & (x * y)^{-1} & x^{-1} * (y^{-1} * z) & \to & (x * y)^{-1} * z \\
(x * y)^{-1} * y & \to & x^{-1} & (x * y)^{-1} * (y * z) & \to & x^{-1} * z
\end{array}$$

6.1.2 Variants

Grâce à cette propriété, nous pouvons calculer les *variants* d'une règle de protocole de la façon suivante : pour chaque règle de protocole $v \Rightarrow w$, on considère l'ensemble fini de paires $v_1 \Rightarrow w_1, \ldots, v_n \Rightarrow w_n$ tel que $\{(v\sigma\downarrow, w\sigma\downarrow) \mid \sigma \in \Sigma\} = \bigcup_{i=1}^{n}\{(v_i\sigma, w_i\sigma) \mid \sigma \in \Sigma\}$. Les règles $v_i \Rightarrow w_i$ sont appelées les *variants* de la règle $v \Rightarrow w$. S'il y a une attaque, on peut choisir pour chaque règle le variant adéquat de façon à ce que, si $\mathcal{E}_2$ est vide ou AC, il n'y a aucun redex introduit par la règle de protocole.

Par exemple, si l'on dispose de la règle de protocole $x \Rightarrow x \oplus b$, on peut en déduire les trois variants suivants :

$$\begin{array}{rcl} b & \Rightarrow & 0 \\ x \oplus b & \Rightarrow & x \\ y & \Rightarrow & y \oplus b \end{array}$$

On notera qu'il existe d'autres variants possibles.

De la même façon, nous allons définir les *variants* d'une règle de $\mathcal{S}$. Examinons cette règle de $\mathcal{S}$:

$$\frac{T \vdash u_1 \ldots T \vdash u_n}{T \vdash u}\,\mathcal{S}$$

Si l'on y applique la propriété des variants finis, on peut définir un ensemble de variants de cette règle :

$$\frac{T \vdash v_{11} \ldots T \vdash v_{1n}}{T \vdash v_1}\,\mathcal{S}_1 \quad \ldots \quad \frac{T \vdash v_{m1} \ldots T \vdash u_{mn}}{T \vdash v_m}\,\mathcal{S}_m$$

Les v_{ij} sont de telle façon que :

$$\{(u_1\sigma\downarrow, \ldots, u_n\sigma\downarrow, u\sigma\downarrow) \mid \sigma \in \Sigma\} = \bigcup_{i=1}^{m}\{(v_{i1}\sigma, \ldots, v_{in}, v_i\sigma) \mid \sigma \in \Sigma\}$$

Comme pour les règles de protocole, s'il y a une preuve menant à une attaque, il est possible de choisir à chaque application de la règle de $\mathcal{S}$ la règle n'introduisant pas de redex, si $\mathcal{E}_2$ est vide ou égal à AC. Il n'est alors plus nécessaire d'imposer une forme normale sur la conclusion de la règle.

6.1.2.1 Le cas du « ou exclusif »

Prenons l'exemple du « ou exclusif ». Les variants des règles de $\mathcal{S}$ propres ou « ou exclusif » sont de la forme suivante :

$$\frac{T \vdash x \oplus y \quad T \vdash y}{T \vdash x} \qquad \frac{T \vdash x \quad T \vdash x \oplus y}{T \vdash y}$$

$$\frac{T \vdash x \oplus z \quad T \vdash z \oplus y}{T \vdash x \oplus y} \qquad \frac{T \vdash x \oplus v \oplus w \quad T \vdash y \oplus u \oplus v \quad T \vdash z \oplus u \oplus w}{T \vdash x \oplus y \oplus z}$$

$$\frac{T \vdash x \oplus u_{xy} \oplus u_{xz} \oplus u_{xw} \quad T \vdash y \oplus u_{xy} \oplus u_{yz} \oplus u_{yw} \quad T \vdash z \oplus u_{xz} \oplus u_{yz} \oplus u_{zw} \quad T \vdash w \oplus u_{xw} \oplus u_{yw} \oplus u_{zw}}{T \vdash x \oplus y \oplus z \oplus w}$$

Le dernier exemple de variants permet de généraliser facilement la construction des variants pour une arité donnée de l'opérateur $\oplus$.

L'opérateur $\oplus$ étant variadique, on construit ainsi un nombre infini de variants ; toutefois, pour chaque règle de $\mathcal{S}$, il n'y a qu'un nombre fini de variants qui lui corresponde.

6.1.2.2 Le cas des groupes abéliens

Prenons maintenant le cas des groupes abéliens. Comme pour le « ou exclusif », il y a une infinité de règles dans $\mathcal{S}$ en raison de l'opérateur variadique du groupe. Par contre, pour une arité donnée de l'opérateur de groupe, il y a beaucoup plus de variants. Voici quelques exemples de variants pour les groupes abéliens.

$$\frac{T \vdash 1}{T \vdash 1} \qquad \frac{T \vdash x^{-1}}{T \vdash x}$$

$$\frac{T \vdash x * y^{-1}}{T \vdash x^{-1} * y} \qquad \frac{T \vdash x \quad T \vdash 1}{T \vdash x}$$

$$\frac{T \vdash x \quad T \vdash x^{-1}}{T \vdash 1} \qquad \frac{T \vdash x^{-1} \quad T \vdash y^{-1}}{T \vdash (x * y)^{-1}}$$

$$\frac{T \vdash (x * y)^{-1} \quad T \vdash y}{T \vdash x^{-1}} \qquad \frac{T \vdash x^{-1} \quad T \vdash (y * z)^{-1}}{T \vdash (x * y)^{-1} * z}$$

Cette liste est loin d'être exhaustive, malgré sa limitation à l'arité 2 pour l'opérateur de groupe.

6.1.3 Formes résolues

Le fait de ne plus avoir de redex dans les contraintes nous permet de définir une forme résolue pour celles-ci :

Définition 6.2 (Forme résolue) *Une conjonction d'équations E est en* forme résolue *s'il s'agit d'une conjonction d'équations $x_1 = u_1 \wedge \ldots \wedge x_n = u_n$ où $x_1, \ldots, x_n$ sont les uniques variables de E et pour tout $1 \leq i \leq j \leq n$, x_i n'apparaît pas dans u_j. $x_1, \ldots, x_n$ sont classées dans l'ordre lexicographique.*

Si $\mathcal{E}$ a la propriété des variants finis, toute contrainte de E est équivalente à un ensemble fini de formes résolues.

Lemme 6.2 *Une forme résolue E a une unique solution la plus générale dont le domaine est $\{x_1, \ldots, x_n\}$.*

Quand E est une forme résolue, on note σ_E son unique solution la plus générale.

Une autre propriété des formes résolues est que du fait de l'ordre lexicographique choisi pour ordonner les variables, si E_1 et E_2 sont des formes résolues, si $E_1 \wedge E_2$ a une solution, après suppression arbitraire des doublons, il s'agit déjà d'une forme résolue. Il est bien entendu possible d'obtenir également une autre forme résolue de $E_1 \wedge E_2$. En effet, il n'y a pas une unique forme résolue pour un E donné.

6.2 Modèle restreint du système de preuves

Dans toute la suite, on suppose que $\mathcal{E}$ a la propriété des variants finis.

Les règles de $\widehat{\mathcal{S}}$ vont être restreintes de façon à utiliser uniquement des contraintes sous forme résolue dans les séquents : elles sont interprétées dans la théorie $\mathcal{E}_2$ qui sera par la suite la théorie vide ou AC. De plus, les instanciations auront une forme plus restrictive et nous imposerons un choix correct du variant pour les règles de protocole et de $\mathcal{S}$.

La normalisation des termes se limitera à la théorie $\mathcal{E}_2$ (donc vide ou AC). En effet, pour chaque règle, il sera possible de choisir le bon variant pour ne plus introduire de redex. Tout terme de la preuve se retrouvera ainsi sous forme normale. Il n'est donc plus nécessaire d'utiliser le symbole $\downarrow$.

La nouvelle théorie $\overline{\mathcal{S}}$ est alors ainsi définie :

- les règles de protocole sont modifiées pour inclure le variant utilisé au niveau des points de contrôle. On impose également l'utilisation de formes résolues :

$$\frac{T \vdash u \ \ [\![E]\!]}{T \vdash w \ \ [\![E']\!]}\,\mathcal{P}$$

  où E' est l'une des formes résolues de la conjonction suivante :

$$v = u \wedge E \wedge \sigma_s \wedge x_{R,k,s,v} = 1$$

  ξ est redéfini étant l'ensemble des variables de contrôle de la forme $x_{R,k,s,v}$. Cette formulation impose l'utilisation d'un variant n'utilisant pas de redex. Pour éviter d'alourdir la notation, on continuera cependant de noter $x_{R,k,s}$ en considérant que le variant est encodé dans le rôle R. Par exemple, si le protocole initial admettait deux rôles Alice et Bob et que l'application de la propriété des variants finis permet d'obtenir trois variantes du rôle d'Alice, on considérera que le protocole traité contient les rôles Alice_1, Alice_2, Alice_3 et Bob.
- la règle d'instanciation est modifiée ainsi :

$$\frac{T \vdash C[x]_p \ \ [\![E \wedge x = u]\!]}{T \vdash C[u]_p \ \ [\![E \wedge x = u]\!]}\,\mathcal{I}$$

  C est un contexte quelconque. L'instanciation se fait par position et non pas en utilisant des substitutions.
- les règles de $\mathcal{S}$ sont modifiées de façon à utiliser les variants de celles-ci. On impose l'utilisation de formes résolues et on interdit d'utilisation de règles introduisant un redex. La conclusion d'une telle règle n'a plus besoin d'être mise en forme normale, par cette restriction.
- les autres règles sont modifiées de façon à ce que la contrainte finale soit sous forme résolue. De plus, il n'est plus nécessaire d'utiliser la forme normale car tous les termes sont déjà en forme normale (selon la théorie vide ou AC).

Enfin, on considère que les termes de T sont déjà réduits. Les règles du système $\overline{\mathcal{S}}$ sont résumées dans la figure 6.1 page suivante.

6.3 Correction et complétude du modèle restreint

Nous allons nous attacher désormais à montrer la complétude et la correction de ce nouveau modèle par rapport à $\widehat{\mathcal{S}}$ présenté dans le chapitre 5 page 45. La principale difficulté dans

Contrairement à la figure 4.1 page 39, dans cette figure, les contraintes contenues dans les hypothèses sont réputées être sous forme résolue et la contrainte des conclusions doivent être mises sous forme résolue. La principale autre différence avec la figure 4.1 page 39 est l'absence de normalisation dans les conclusions dues à l'absence de redex.

Règle de $\mathcal{S}$.

$$\frac{T \vdash u'_1 \ \llbracket E_1 \rrbracket \quad \ldots \quad T \vdash u'_k \ \llbracket E_k \rrbracket}{T \vdash u \ \llbracket E_1 \wedge \ldots \wedge E_k \rrbracket} \, \mathcal{S}$$

Ouverture de session . Si $v \Rightarrow w$ est la première règle du rôle R, σ_s est l'ensemble des affectations initiales propres à la session s et $E \not\models_{\mathcal{E}} x_{R,1,s} = 1$, alors :

$$\frac{T \vdash u \ \llbracket E \rrbracket}{T \vdash w \ \llbracket u = v \wedge E \wedge \sigma_s \wedge x_{R,1,s} = 1 \rrbracket} \, \mathcal{P}$$

Avancement de session . Si $v \Rightarrow w$ est la règle numérotée $k > 1$ du rôle R, $E \not\models_{\mathcal{E}} x_{R,k,s} = 1$ et $E \models_{\mathcal{E}} x_{R,k-1,s} = 1$, alors :

$$\frac{T \vdash u \ \llbracket E \rrbracket}{T \vdash w \ \llbracket u = v \wedge E \wedge x_{R,k,s} = 1 \rrbracket} \, \mathcal{P}$$

Agent compromis . Si $E \models_{\mathcal{E}} x_{R,1,s} = 1$ et l'acteur principal de la session s est compromis, alors :

$$\frac{T \vdash u \ \llbracket E \rrbracket}{T \vdash N_i(s) \ \llbracket E \rrbracket} \, \mathcal{ND}$$

Règle d'instanciation .

$$\frac{T \vdash C[x]_p \ \llbracket E \wedge x = u \rrbracket}{T \vdash C[u]_p \ \llbracket E \wedge x = u \rrbracket} \, \mathcal{I}$$

Règle d'affaiblissement .

$$\frac{T \vdash u_1 \ \llbracket E_1 \rrbracket \quad T \vdash u_2 \ \llbracket E_2 \rrbracket}{T \vdash u_1 \ \llbracket E_1 \wedge E_2 \rrbracket} \, \mathcal{W}$$

FIG. 6.1 - *Règles de définition du système $\overline{\mathcal{S}}$*

cette section vient du fait que le système de réécriture utilisé pour la normalisation dans $\widehat{\mathcal{S}}$ peut être différent du système de réécriture utilisé pour les variants finis. On se retrouve alors avec deux possibilités différentes pour normaliser un même terme. Afin d'éviter le confusions, on renomme donc le symbole $\downarrow$ en $\downarrow_{\widehat{\mathcal{S}}}$ quand il s'agit du symbole utilisé pour la normalisation dans $\widehat{\mathcal{S}}$ et $\downarrow_{\overline{\mathcal{S}}}$ quand il s'agit de celui introduit dans ce chapitre par les variants finis. Notons simplement que pour tout terme t, $t\downarrow_{\overline{\mathcal{S}}}=_{\mathcal{E}} t\downarrow_{\widehat{\mathcal{S}}}$.

6.3.1 Complétude

Nous montrons d'abord qu'une attaque dans $\widehat{\mathcal{S}}$ se traduit en une attaque dans le nouveau modèle $\overline{\mathcal{S}}$.

Lemme 6.3 (Complétude de $\overline{\mathcal{S}}$) *Pour toute preuve Π de $T \vdash s\ \llbracket E\rrbracket$ dans $\widehat{\mathcal{S}}$, pour tout σ forme résolue de E, il existe une preuve Π' de $T \vdash s'\ \llbracket\sigma\rrbracket$ dans $\overline{\mathcal{S}}$ telle que $s'\sigma\downarrow_{\widehat{\mathcal{S}}}=_{\epsilon_2} s\sigma\downarrow_{\overline{\mathcal{S}}}$.*

Preuve. Procédons par récurrence sur la taille de la preuve. Une preuve ne contenant que l'axiome est aussi bien dans $\widehat{\mathcal{S}}$ que dans $\overline{\mathcal{S}}$ (la contrainte est vide et le terme est déjà réduit car issu de T).

Examinons désormais la dernière règle de la preuve Π :

Règle d'affaiblissement

$$\dfrac{\dfrac{\Pi_1}{T \vdash u_1\ \llbracket E_1\rrbracket} \quad \dfrac{\Pi_2}{T \vdash u_2\ \llbracket E_2\rrbracket}}{T \vdash u_1\downarrow_{\widehat{\mathcal{S}}}\ \llbracket E_1 \wedge E_2\rrbracket}\ \mathcal{W}$$

Soit σ une forme résolue de $E_1 \wedge E_2$. On note σ_1 (respectivement σ_2) la forme résolue de E_1 (respectivement E_2) telle que $\sigma_1 \subset \sigma$ (respectivement $\sigma_2 \subset \sigma$). On applique l'hypothèse de récurrence sur les preuves Π_1 et Π_2 en utilisant σ_1 et σ_2. Il existe alors des preuves Π'_1 et Π'_2 de, respectivement, $T \vdash u'_1\ \llbracket\sigma_1\rrbracket$ et $T \vdash u'_2\ \llbracket\sigma_2\rrbracket$ dans $\overline{\mathcal{S}}$ telles que $u_1\sigma_1\downarrow_{\widehat{\mathcal{S}}}=_{\epsilon_2} u'_1\sigma_1\downarrow_{\overline{\mathcal{S}}}$ et $u_2\sigma_2\downarrow_{\widehat{\mathcal{S}}}=_{\epsilon_2} u'_2\sigma_2\downarrow_{\overline{\mathcal{S}}}$.

En appliquant la règle d'affaiblissement à ces deux preuves, on obtient une preuve Π' de $T \vdash u'_1\ \llbracket\sigma\rrbracket$. Étant donné que $u'_1\sigma_1$ et $u'_2\sigma_2$ sont clos, on a les égalités $u'_1\sigma_1\downarrow_{\overline{\mathcal{S}}}= u'_1\sigma_1\sigma\downarrow_{\overline{\mathcal{S}}}= u'_1\sigma\downarrow_{\overline{\mathcal{S}}}$ et de même $u_1\sigma_1\downarrow_{\widehat{\mathcal{S}}}= u_1\sigma\downarrow_{\widehat{\mathcal{S}}}$, ce qui permet de conclure $u_1\sigma\downarrow_{\widehat{\mathcal{S}}}= u'_1\sigma\downarrow_{\overline{\mathcal{S}}}$. Il en est de même avec le triplet u_2, u'_2, σ_2.

Règle de $\mathcal{S}$

$$\dfrac{\dfrac{\Pi_1}{T \vdash u_1\ \llbracket E_1\rrbracket} \quad \ldots \quad \dfrac{\Pi_k}{T \vdash u_k\ \llbracket E_k\rrbracket}}{T \vdash u\downarrow_{\widehat{\mathcal{S}}}\ \llbracket E_1 \wedge \ldots \wedge E_k\rrbracket}\ \mathcal{S}$$

Soit σ une forme résolue de $E_1 \wedge \ldots \wedge E_k$. On note σ_i la forme résolue de E_i telle que $\sigma_i \subset \sigma$ pour tout i. De plus $\biguplus_i \sigma_i = \sigma$.

Nous pouvons alors appliquer l'hypothèse de récurrence sur chacun des Π_i avec σ_i pour obtenir des preuves Π'_i de $T \vdash u'_i\ \llbracket\sigma_i\rrbracket$ dans $\overline{\mathcal{S}}$.

La propriété des variants finis permet d'affirmer qu'il existe un variant de la règle originale tel que son application aux preuves $T \vdash u'_i\ \llbracket\sigma_i\rrbracket$ (éventuellement instanciées par une succession de règles d'instanciations dans le cas où la règle de $\mathcal{S}$ ne peut plus s'appliquer) donne une preuve $T \vdash u' \;\; [\![\sigma]\!]$ dont la conclusion est réduite (et close car il n'y a pas d'introduction de variables à ce niveau).

Comme pour les règles d'affaiblissement, on a les égalités $u_i\sigma \downarrow_{\widehat{S}} = u'_i\sigma \downarrow_{\overline{S}}$ pour chaque i. La propriété des variants finis nous permet de conclure que $u\sigma \downarrow_{\widehat{S}} = u'\sigma \downarrow_{\overline{S}}$.

Règle de protocole

$$\dfrac{\dfrac{\Pi}{T \vdash u \;\; [\![E]\!]}}{T \vdash w \downarrow_{\widehat{S}} \;\; [\![u = v \wedge E \wedge \sigma_s \wedge x_{R,1,s} = 1]\!]} \mathcal{P}$$

Soit σ une forme résolue de $u = v \wedge E \wedge \sigma_s \wedge x_{R,1,s} = 1$. On note σ_1 la forme résolue de E telle que $\sigma_1 \subset \sigma$. L'hypothèse de récurrence appliquée sur Π nous fournit une preuve Π' de $T \vdash u' \;\; [\![\sigma_1]\!]$ telle que $u\sigma_1 \downarrow_{\widehat{S}} = u'\sigma_1 \downarrow_{\overline{S}}$.

On choisit le variant de la règle de protocole originale tel que $u = v$ n'introduit pas de redex et tel $w\sigma$ ne contienne pas de redex (w est considéré comme déjà réduit). L'application de cette règle nous donne une preuve de $T \vdash w \;\; [\![\sigma]\!]$ (c'est possible grâce au choix du variant). La seule difficulté réside dans le fait que, lors de l'avancement de session, ce soit également le variant v qui ait été utilisé pour l'étape $k - 1$. En fait, le variant est partiellement choisi à chaque étape : il est caractérisé par les variants des règles aux étapes précédentes mais les variants des règles suivantes ne sont pas encore choisis. Pour une session et une étape données, ce sera toujours les même variant qui sera utilisé dans toute la preuve : le terme et la contrainte considérés ensemble donnent toujours le même terme clos.

w n'est pas un terme clos et il n'est donc pas forcément possible de vérifier l'égalité de l'hypothèse de récurrence dans le cas où $\downarrow_{\widehat{S}}$ et $\downarrow_{\overline{S}}$ sont différents. Il convient donc d'effectuer des instanciations. σ étant sous forme résolue, il est possible de trouver à partir de toute variable un chemin vers un terme clos. Ce chemin va se traduire en une suite d'instanciation sur la position de la variable dans w. Ainsi pour toute position de la variable x de $w = C[x]_p$, on applique la règle d'instanciation pour obtenir une preuve de $C[u]_p$ si $x = u \in \sigma$. On garde la même contrainte σ. Le processus est itéré jusqu'à épuiser toutes les variables présentes. Il termine car σ est sous forme résolue. On obtient alors une preuve de $T \vdash w\sigma \;\; [\![\sigma]\!]$. Le variant a été choisi de façon à ce que $w\sigma$ soit déjà réduit. Les hypothèses demandées sont ainsi vérifiées.

Compromission d'un agent

$$\dfrac{\dfrac{\Pi}{T \vdash u \;\; [\![E]\!]}}{T \vdash N_i(s) \;\; [\![E]\!]} \mathcal{CA}$$

Il suffit d'appliquer l'hypothèse de récurrence sur la preuve Π puis d'utiliser la même règle pour obtenir une preuve de $T \vdash N_i(s) \;\; [\![E']\!]$ car $N_i(s)$ est un terme déjà réduit.

Règle d'instanciation

$$\dfrac{\dfrac{\Pi}{T \vdash u \;\; [\![E]\!]}}{T \vdash u\theta \downarrow_{\widehat{S}} \;\; [\![\theta]\!]} \mathcal{I}$$

Ici, θ est une forme résolue de E. On applique l'hypothèse de récurrence sur Π pour obtenir une preuve de $T \vdash u' \ \llbracket\theta\rrbracket$ telle que $u\theta \downarrow_{\widehat{S}} = u'\theta \downarrow_{\overline{S}}$. Cette preuve vérifie déjà les conditions de l'hypothèse de récurrence.

□

6.3.2 Correction

Nous montrons ensuite que, réciproquement, une attaque dans $\overline{\mathcal{S}}$ se traduit en une attaque dans $\widehat{\mathcal{S}}$.

Lemme 6.4 (Correction de $\overline{\mathcal{S}}$) *Pour tout preuve Π de $T \vdash s \ \llbracket E\rrbracket$ dans $\overline{\mathcal{S}}$, il existe une preuve Π' de $T \vdash s\sigma \downarrow_{\widehat{\mathcal{S}}} \ \llbracket E'\rrbracket$ dans $\widehat{\mathcal{S}}$ telle que $E \models\!\!\!\!\mid E'$ et $E \models \sigma$.*

Preuve. Nous allons de nouveau procéder par récurrence sur la taille de la preuve Π. Le cas de base est trivial. Considérons la dernière règle de Π.

Règle d'affaiblissement

$$\frac{T \vdash u_1 \ \llbracket E_1\rrbracket \quad T \vdash u_2 \ \llbracket E_2\rrbracket}{T \vdash u_1 \ \llbracket E_1 \wedge E_2\rrbracket}\ \mathcal{W}$$

On applique l'hypothèse de récurrence sur la preuve de $T \vdash u_1 \ \llbracket E_1\rrbracket$ et celle de $T \vdash u_2 \ \llbracket E_2\rrbracket$ pour obtenir respectivement des preuves de $T \vdash u_1\sigma_1 \downarrow_{\widehat{\mathcal{S}}} \ \llbracket E_1'\rrbracket$ et $T \vdash u_2\sigma_2 \downarrow_{\widehat{\mathcal{S}}} \ \llbracket E_2'\rrbracket$. On applique la règle d'affaiblissement sur ces deux preuves pour obtenir la preuve de $T \vdash u_1\sigma_1 \downarrow_{\widehat{\mathcal{S}}} \ \llbracket E_1' \wedge E_2'\rrbracket$. Comme $E_1 \models\!\!\!\!\mid E_1'$ et $E_2 \models\!\!\!\!\mid E_2'$, on a $E_1 \wedge E_2 \models\!\!\!\!\mid E_1' \wedge E_2'$. De plus, $E_1 \models \sigma_1$ donc $E_1 \wedge E_2 \models \sigma_1$.

Règle de $\mathcal{S}$

$$\frac{T \vdash u_1 \ \llbracket E_1\rrbracket \quad \ldots \quad T \vdash u_k \ \llbracket E_k\rrbracket}{T \vdash u \ \llbracket E_1 \wedge \ldots \wedge E_k\rrbracket}$$

On applique l'hypothèse de récurrence sur chacune des hypothèses pour obtenir les preuves de $T \vdash u_i\sigma_i \downarrow_{\widehat{\mathcal{S}}} \ \llbracket E_i'\rrbracket$ avec $E_i \models\!\!\!\!\mid E_i'$. La dernière règle de Π est un variant d'une règle provenant de $\mathcal{S}$. On applique la règle originale pour obtenir une preuve de $T \vdash u' \downarrow_{\widehat{\mathcal{S}}} \ \llbracket E_1' \wedge \ldots \wedge u_k'\rrbracket$. Comme $E_i' \models\!\!\!\!\mid E_i$, on a l'équivalence souhaitée sur les contraintes finales. On note σ l'union des σ_i. On a $E_1 \wedge \ldots \wedge E_k \models \sigma$ et $u' \downarrow_{\widehat{\mathcal{S}}} = u\sigma \downarrow_{\widehat{\mathcal{S}}}$ car $u\sigma - u'\sigma \downarrow_{\overline{\mathcal{S}}}$ par définition du variant.

Règle de protocole

$$\frac{T \vdash u \ \llbracket E\rrbracket}{T \vdash w \ \llbracket u = v \wedge E \wedge \sigma_s \wedge x_{R,k,s,v} = 1\rrbracket}$$

On applique l'hypothèse de récurrence pour obtenir une preuve de $T \vdash u\sigma \downarrow_{\widehat{\mathcal{S}}} \ \llbracket E'\rrbracket$ avec $E' \models\!\!\!\!\mid E$. On applique la règle de protocole originale $v_o \Rightarrow w_o$ correspondant au variant. Au besoin, on renomme les sessions pour éviter les collisions. On obtient alors une preuve de $T \vdash w_o \downarrow_{\widehat{\mathcal{S}}} \ \llbracket u\sigma \downarrow_{\widehat{\mathcal{S}}} = v_o \wedge E' \wedge \sigma_s \wedge x_{R,k,s} = 1\rrbracket$. Par définition du variant, $w\sigma = w_0\sigma \downarrow_{\overline{\mathcal{S}}}$ et donc $w\sigma \downarrow_{\widehat{\mathcal{S}}} = w_0\sigma \downarrow_{\widehat{\mathcal{S}}}$. Il est donc encore nécessaire de faire des instanciations sur $w_0 \downarrow_{\widehat{\mathcal{S}}}$ pour obtenir $w\sigma \downarrow_{\widehat{\mathcal{S}}}$. On effectue la suite d'instanciations σ (car $E \models \sigma$) pour obtenir une preuve de $w\sigma \downarrow_{\widehat{\mathcal{S}}}$ avec une contrainte logiquement équivalente à la contrainte $u\sigma \downarrow_{\widehat{\mathcal{S}}} = v_o \wedge E' \wedge \sigma_s \wedge x_{R,k,s} = 1$.

On a $v\sigma = v_0\sigma \downarrow_{\overline{\mathcal{S}}}$ et donc $v\sigma \downarrow_{\widehat{\mathcal{S}}} = v_0\sigma \downarrow_{\widehat{\mathcal{S}}}$ et on peut donc choisir la contrainte $u\sigma \downarrow_{\widehat{\mathcal{S}}} = v_o \wedge E' \wedge \sigma_s \wedge x_{R,k,s} = 1$ de façon à ce qu'elle soit logiquement équivalente à la contrainte originale (car celle-ci est un modèle de σ).

Compromission d'un agent Cette règle est triviale. Comme pour la règle de protocole, on doit juste prendre soit de renommer les sessions en cas de conflit.

Règle d'instanciation

$$\frac{T \vdash C[x]_p \ \ [\![E \wedge x = u]\!]}{T \vdash C[u]_p \ \ [\![E \wedge x = u]\!]}$$

On applique l'hypothèse de récurrence sur la preuve de $T \vdash C[x]_p \ [\![x = u \wedge E]\!]$ pour obtenir une preuve de $T \vdash C[x]_p\sigma \downarrow_{\widehat{\mathcal{S}}} \ \ [\![E']\!]$ avec $E' \models\!\mid x = u \wedge E$ et $x = u \wedge E \models \sigma$. On applique la règle d'instanciation pour obtenir une preuve de $T \vdash C[x]_p\sigma\theta \downarrow_{\widehat{\mathcal{S}}} \ \ [\![\theta]\!]$ avec $\theta \models\!\mid E'$. On a bien $E \wedge x = u \models \sigma \circ \theta$ car $E \wedge x = u$ est modèle de σ et de θ.

□

Il est ainsi possible d'exprimer le secret dans $\overline{\mathcal{S}}$:

Lemme 6.5 *Il y a une attaque sur le secret de s si et seulement si il existe une preuve de $T \vdash s'$ $[\![E]\!]$ dans $\overline{\mathcal{S}}$ telle que $s =_{AC} s'\sigma$ et $\sigma \models E$.*

$\overline{\mathcal{S}}$ permet de se débarrasse de la complexité des théories équationnelles tout en restant correct et complet vis-à-vis du secret pour les systèmes qui nous intéressent principalement. Il va nous permettre de donner une forme normale de preuve de manière plus aisée.

Deuxième partie

Théorème de normalisation

CHAPITRE 7

Objectifs

Les travaux [RT01, CKRT03b, CKRT03a, CKRT05] fournissent une preuve *ad hoc* à chaque théorie étudiée pour le problème du secret. Nous voulons éviter les preuves répétitives pour chaque nouvelle théorie de l'intrus et ainsi proposer un résultat général s'appliquant à un grand nombre de théories. Idéalement, l'ajout d'une nouvelle théorie ne devrait alors nécessiter que la preuve de quelques propriétés les plus syntaxiques possibles.

Jusqu'ici, les preuves de décision se fondent :

- sur un résultat de normalisation de preuve (localité ; propriété de la sous-formule) pour le pouvoir de l'intrus (cas passif) ;
- sur un résultat permettant de borner la taille des messages nécessaires à l'intrus pour produire l'attaque.

La seconde condition est proche de la première et peut alors se traduire également en une normalisation de preuve.

Si le protocole est vu comme un oracle pour l'intrus (cas de l'intrus actif), les règles correspondant aux protocoles sont simplement des règles supplémentaires augmentant le pouvoir de l'intrus. Nous sommes maintenant dans un cadre unifié permettant d'exprimer le résultat de [RT01] comme la conséquence d'un résultat de preuve normale. L'objectif est alors de montrer que l'on peut effectivement l'obtenir par normalisation de preuve.

Les avantages sont multiples :

1. La recherche de preuve peut se faire uniquement parmi les preuves en forme normale, que l'on suppose ou non un nombre borné de sessions ;
2. Les deux points présentés ci-dessus (localité et borne sur la taille maximale des messages messages nécessaires à l'intrus pour produire l'attaque) se trouvent unifiés ;
3. Les conditions suffisantes sur la théorie de l'intrus apparaissent clairement : ce résultat est paramétré par la théorie de l'intrus.

Nous allons dans un premier temps démontrer deux lemmes qui permettront de simplifier grandement la preuve de normalisation par la suite (chapitre 8 page 65). Malgré ces simplifications, il nous faudra poser des hypothèses supplémentaires sous peine de tomber sur des résultats d'indécidabilité (chapitre 9 page 71). Ces limitations sont comparables à ce que l'on peut trouver dans [CKRT03b] à propos des règles d'oracle. D'une manière plus générale, ces limitations sont nécessaires car le secret tel que défini dans notre modèle est indécidable [DLMS99]. Enfin, nous introduisons le théorème et sa preuve dans le chapitre 10 page 81.

CHAPITRE 8

Simplifications

Sommaire

Nous allons introduire dans ce chapitre deux lemmes différents destinés à simplifier la preuve du théorème du chapitre 10 page 81. Ces deux lemmes sont placés dans ce chapitre à part car ils ne nécessitent pas les hypothèses qui seront introduites dans le chapitre intermédiaire 9 page 71.

Le premier lemme, décrit dans la section 8.1, permet de s'assurer que dans une preuve, les contraintes sont sous la forme utilisant un maximum de variables. Le second lemme, décrit dans la section 8.2 page 67 permet de repousser au plus tard les instanciations. En effet, la profondeur de celles-ci sera problématique et les repousser permettra de supprimer les cas difficiles.

On rappelle que dans ce chapitre, la théorie équationnelle est AC ou vide.

8.1 Factorisaton des contraintes

Le modèle $\overline{S}$ défini dans le chapitre 6 page 51 impose les contraintes sous forme résolue. Toutefois, le choix de la forme résolue est laissé à chaque étape. Ainsi, la contrainte $x = 5 \wedge y = 5$ peut être présentée sous la forme $x = y \wedge y = 5$. La règle d'instanciation fonctionne, quant à elle, de manière syntaxique. Dans le cas de la première contrainte x ne peut être instancié que par 5. Dans le cas de la seconde, il ne peut être instancié que par y (qui peut ensuite être instancié par 5).

Nous allons voir dans cette section qu'il est possible de transformer toute preuve de $\overline{S}$ en une autre preuve de $\overline{S}$ où les contraintes sont factorisées au maximum. La définition suivante formalise cet aspect.

Définition 8.1 (Forme factorisée) *Une contrainte est dite sous forme factorisée si et seulement si :*

1. *elle est sous forme résolue (cf définition 6.2 page 54),*
2. *elle est de score minimal.*

Le score d'une contrainte est le nombre de symboles, à l'exclusion des symboles de variables, apparaissant dans celle-ci. Le score du symbole 1 *apparaissant dans une contrainte du type* $x_{R,k,s} = 1$ *est également nul, tandis qu'une forme du type* $x_{R,k,s} = z$ *a un score infini.*

Par exemple, la contrainte $x = f(y, y) \wedge z = f(y, y) \wedge y = a$ a comme score 3 (deux fois le symbole f et une fois le symbole a). La contrainte $x = z \wedge z = f(y, y) \wedge y = a$ a comme score 2 et la contrainte $x = f(a, y) \wedge z = f(y, a) \wedge y = a$ a comme score 5.

Le lemme suivant nous assure que toute variable dans une contrainte de la preuve Π est liée au même terme, modulo la contrainte E.

Lemme 8.1 *Pour toute preuve* Π *de* $T \vdash s \; [\![E]\!]$ *dans* $\overline{\mathcal{S}}$*, pour toute contrainte* E' *de* Π*,* $E \models E'$*.*

Preuve. Il s'agit d'une récurrence immédiate sur la taille de la preuve : pour chaque règle, la contrainte de la conclusion « contient » les contraintes des hypothèses. □

Il nous sera utile pour prouver le lemme principal de cette section :

Lemme 8.2 *Pour toute preuve* Π *de* $T \vdash s \; [\![E]\!]$ *dans* $\overline{\mathcal{S}}$*, il existe une preuve* Π' *de* $T \vdash s' \; [\![E']\!]$ *où* $E' \models\!\!\!\dashv E$ *et telle que* E' *est sous forme factorisée ainsi que toutes les contraintes de* Π'*.*

Preuve. On procède par récurrence sur la taille de la preuve. Le cas de base est trivial car la contrainte est vide et donc sous forme factorisée (T ne contient pas de variables). Considérons la dernière règle de Π.

Pour toutes les règles, à l'exception de l'instanciation, il suffit d'appliquer l'hypothèse de récurrence sur les prémisses puis d'appliquer la règle considérée et de choisir la contrainte sous forme factorisée pour conclure : seule l'instanciation travaille de manière syntaxique avec la contrainte.

Considérons alors la règle d'instanciation suivante :

$$\frac{T \vdash C[x] \;\; [\![E \wedge x = u]\!]}{T \vdash C[u] \;\; [\![E \wedge x = u]\!]} \mathcal{I}$$

En appliquant l'hypothèse de récurrence sur l'hypothèse, on obtient alors la preuve de $T \vdash C[x] \;\; [\![E' \wedge x = u']\!]$ où u' est de score inférieur à u. Il suffit alors d'appliquer une suite d'instanciation (une seule si $u' = u$) pour obtenir une preuve de $T \vdash C[u] \;\; [\![E' \wedge x = u']\!]$: il existe un chemin de $E \wedge x = u$ vers $E' \wedge x = u$ dont les transitions sont le remplacement d'un sous-terme t en partie droite par une variable y telle que $y = t$ apparaisse syntaxiquement dans $E \wedge x = u$ et il est possible de suivre ce chemin en sens inverse en effectuant des instanciations.

□

Notons que ce lemme doit être appliqué avant celui de la section suivante.

Cette forme factorisée permet de vérifier simplement la propriété $E \models x = y$. Si cette propriété est vérifiée, alors il y a dans E une suite d'égalités $x_i = x_{i+1}$ avec $x_i = x$ et $x_k = y$. Si ce n'était pas le cas, E ne serait pas sous forme factorisée car il existerait une contrainte équivalente dont le score serait strictement inférieur.

8.2 Retardement des instanciations

Dans le modèle $\overline{\mathcal{S}}$, les instanciations sont à profondeur arbitraire et peuvent intervenir à n'importe quel moment. Intuitivement, une instanciation n'est utile que si le terme instancié est nécessaire à la règle utilisée par la suite. Si ce n'est pas le cas, l'instanciation peut être supprimée ou repoussée. C'est ce que nous allons montrer dans cette section.

On considère la règle d'instanciation suivante :

$$\frac{T \vdash C[x]_p \;\; [\![x = u \wedge E]\!]}{T \vdash C[u]_p \;\; [\![x = u \wedge E]\!]}\,\mathcal{I}$$

Tout d'abord, on peut noter que cette preuve :

$$\cfrac{\cfrac{T \vdash C[x]_p[y]_q \;\; [\![x = u \wedge y = v \wedge E]\!]}{T \vdash C[u]_p[y]_q \;\; [\![x = u \wedge y = v \wedge E]\!]}\,\mathcal{I}}{T \vdash C[u]_p[v]_q \;\; [\![x = u \wedge y = v \wedge E]\!]}\,\mathcal{I}$$

peut se réécrire ainsi :

$$\cfrac{\cfrac{T \vdash C[x]_p[y]_q \;\; [\![x = u \wedge y = v \wedge E]\!]}{T \vdash C[x]_p[v]_q \;\; [\![x = u \wedge y = v \wedge E]\!]}\,\mathcal{I}}{T \vdash C[u]_p[v]_q \;\; [\![x = u \wedge y = v \wedge E]\!]}\,\mathcal{I}$$

dès que p et q ne sont pas comparables.

Définition 8.2 *Soit une preuve Π de $T \vdash u$ $[\![E]\!]$ dans $\overline{\mathcal{S}}$, on dit qu'une position p est primaire pour Π si, en appliquant autant que possible la règle de réécriture ci-dessus, il est possible d'obtenir une preuve Π' de $T \vdash u$ $[\![E]\!]$ telle que la dernière règle d'inférence est une instanciation à la position p.*

Pour simplifier, pour la suite, une preuve terminant par une instanciation sera considérée comme une preuve terminant par une instanciation à une position primaire : il conviendra de faire mentalement le remplacement qui s'impose.

Considérons alors les règles de réécriture présentées dans la figure 8.1 page suivante.

On notera que dans la troisième règle de réécriture, $C[u_i]_p = v \downarrow$ et $C[x]_p = v \downarrow$ se réduisent à la même forme résolue du fait de la présence de l'égalité $x = u_i$ dans E.

En ce qui concerne la dernière règle de réécriture de la figure 8.1, si la dernière règle de $\mathcal{S}$ est :

$$\frac{T \vdash t_1 \ldots T \vdash t_n}{T \vdash t}$$

et que $C[x]_p$ s'unifie avec t_i ($t_i\theta = C[x]_p$) et que y est une variable apparaissant dans t_i en tant que préfixe de p ($p = p_0 \cdot r$ et $t_i \mid p_0 = y$), $\mathcal{I}*$ est la séquence d'instanciations aux positions $p_1 \cdot r, \ldots, p_k \cdot r$ où $p_1, \ldots, p_k$ sont les positions de y dans t.

1. $$\dfrac{\dfrac{T \vdash C[x]_p \ \llbracket E \rrbracket}{T \vdash C[u_i]_p \ \llbracket E \rrbracket \dots}\mathcal{I}}{T \vdash C[u_i]_p \ \llbracket E' \rrbracket}\mathcal{W} \quad \Rightarrow \quad \dfrac{\dfrac{T \vdash C[x]_p \ \llbracket E \rrbracket \dots}{T \vdash C[x]_p \ \llbracket E' \rrbracket}\mathcal{W}}{T \vdash C[u_i]_p \ \llbracket E' \rrbracket}\mathcal{I}$$

2. $$\dfrac{\dfrac{T \vdash C[x]_p \ \llbracket E \rrbracket}{T \vdash C[u_i]_p \ \llbracket E \rrbracket}\mathcal{I}}{T \vdash N_i(s) \ \llbracket E \rrbracket}\mathcal{CA} \quad \Rightarrow \quad \dfrac{T \vdash C[x]_p \ \llbracket E \rrbracket}{T \vdash N_i(s) \ \llbracket E \rrbracket}\mathcal{CA}$$

3. $$\dfrac{\dfrac{T \vdash C[x]_p \ \llbracket E \rrbracket}{T \vdash C[u_i]_p \ \llbracket E \rrbracket}\mathcal{I}}{T \vdash w\downarrow \ \llbracket C[u_i]_p = v\downarrow \wedge \dots \rrbracket}\mathcal{P} \quad \Rightarrow \quad \dfrac{T \vdash C[x]_p \ \llbracket E \rrbracket}{T \vdash w\downarrow \ \llbracket C[x]_p = v\downarrow \wedge \dots \rrbracket}\mathcal{P}$$

4. $$\dfrac{T \vdash u_1 \ \llbracket E_1 \rrbracket \dots \quad \dfrac{T \vdash C[x]_p \ \llbracket E_i \rrbracket}{T \vdash C[u_i]_p \ \llbracket E_i \rrbracket}\mathcal{I} \quad \dots T \vdash u_n \ \llbracket E_n \rrbracket}{T \vdash u \ \llbracket E \rrbracket}\mathcal{S}$$

$$\Rightarrow \quad \dfrac{\dfrac{T \vdash u_1 \ \llbracket E_1 \rrbracket \dots T \vdash C[x]_p \ \llbracket E_i \rrbracket \dots T \vdash u_n \ \llbracket E_n \rrbracket}{T \vdash u' \ \llbracket E \rrbracket}\mathcal{S}}{T \vdash u \ \llbracket E \rrbracket}\mathcal{I}*$$

FIG. 8.1 - *Règles de réécriture pour repousser les instanciations*

Lemme 8.3 *Les règles de la figure 8.1 sont correctes et terminent. Les preuves de $\overline{\mathcal{S}}$ sur lesquelles ces règles ont été appliquées sont dites retardées vis-à-vis des instanciations.*

Preuve. Considérons d'abord la correction.

En ce qui concerne la première règle de réécriture, $E \subset E'$ par définition de la règle d'affaiblissement. Pour la seconde règle, la condition ne dépend que de E. Pour la troisième règle, $x\sigma_E = u_i$ et donc les contraintes $C[u_i]_p = v \wedge E$ et $C[x]_p = v \wedge E$ sont équivalentes.

Pour la quatrième règle, il est possible de compléter l'unificateur θ avec θ_1 de façon à ce que $t_i\theta_1 = u_i$ car les variables dans $t_1, \dots, t_n$ n'apparaissent qu'une seule fois en raison de la linéarisation des règles. De plus, la condition introduite par la linéarisation est une conséquence de $E_1 \wedge \dots \wedge E_n$. Ainsi :

$$\dfrac{T \vdash u_1 \ \llbracket E_1 \rrbracket \dots T \vdash C[x]_p \ \llbracket E_i \rrbracket \dots T \vdash u_n \ \llbracket E_n \rrbracket}{T \vdash u' \ \llbracket E \rrbracket}$$

est une instance d'une preuve de $\mathcal{S}$ telle que $u' = t\theta$. Si y est une variable présente dans le terme t_i à une position p_0 préfixe de p : $t_i[y]_{p_0}\theta = C[x]_{p_0 \cdot r}$, $y\theta = C_{|p_0}[x]_r$. $u' = t[y]_{p_1} \dots [y]_{p_k}\theta = t[C_{|p_0}[x]_r\theta]_{p_1} \dots t[C_{|p_0}[x]_r\theta]_{p_k}$ et $u = u'[x\theta]_{p_1 \cdot r} \dots [x\theta]_{p_k \cdot r}$.

En ce qui concerne la terminaison, il suffit d'interpréter les preuves comme des termes sur lesquels on applique les opérateurs $\mathcal{I}, \mathcal{W}, \mathcal{P}, \dots$. On considère alors le système suivant :

$$\begin{array}{rcl}
\mathcal{W}(\mathcal{I}(x)) & \longrightarrow & \mathcal{I}(\mathcal{W}(x)) \\
\mathcal{CA}(\mathcal{I}(x)) & \longrightarrow & \mathcal{CA}(x) \\
\mathcal{P}(\mathcal{I}(x)) & \longrightarrow & \mathcal{P}(x) \\
\mathcal{S}(x_1, \ldots, \mathcal{I}(x_i), \ldots, x_n) & \longrightarrow & \mathcal{I}^k(S(x_1, \ldots, x_n))
\end{array}$$

Ce système termine en considérant par exemple un ordre récursif sur les chemins dans lequel $\mathcal{I}$ est minimal. $\square$

CHAPITRE 9

Limitations

Sommaire

Nous allons montrer dans ce chapitre qu'il nous faut prendre des hypothèses supplémentaires pour obtenir des résultats de décidabilité, y compris quand le nombre de sessions est borné. Cela va nous amener à poser des restrictions supplémentaires sur ce modèle, sous forme d'hypothèses sur les règles de $\mathcal{S}$.

9.1 Indécidabilité du secret dans $\overline{\mathcal{S}}$

Afin de montrer l'indécidabilité dans $\overline{\mathcal{S}}$, nous allons considérer une réduction du problème de correspondance Post à la recherche du secret pour un système particulier de $\overline{\mathcal{S}}$. Rappelons d'abord la définition de Post :

Définition 9.1 (Problème de correspondance de Post) *Soit une liste finie de paires de chaînes $P = \{(u_1, v_1), \ldots, (u_n, v_n)\}$. Existe-t-il une séquence $(i_1, \ldots, i_m)$ d'entiers inférieurs à n telle que $u_{i_1} \ldots u_{i_m} = v_{i_1} \ldots v_{i_m}$.*

Ce problème est indécidable, y compris avec un nombre borné de sessions. Considérons alors les règles de la figure 9.1 page suivante. u_i et v_i correspondent à des compositions de fonctions a et b. Par exemple, si u_1 est le mot $abbab$, alors dans la figure 9.1, u_1 est vu comme la composition $a \circ b \circ b \circ a \circ b$.

Les symboles f, a et b sont libres et il n'y a pas de théories équationnelles. L'intrus dispose en plus de l'instanciation et de l'affaiblissement. Il n'y a pas de règle de compromission d'agents car il n'y a pas de nonces. On ne fait pas non plus appel à des règles de protocole.

$$\frac{T \vdash f(u_i(x), v_i(y)) \;\; [\![E]\!]}{T \vdash f(x,y) \;\; [\![E]\!]}\,\mathcal{S}$$

$$\frac{T \vdash x \;\; [\![E]\!]}{T \vdash f(x,x) \;\; [\![E]\!]}\,\mathcal{S}$$

$$\frac{T \vdash x \;\; [\![E]\!]}{T \vdash u_i(x) \;\; [\![E]\!]}\,\mathcal{S}$$

FIG. 9.1 - *Règles de $\overline{\mathcal{S}}$ pour la réduction de Post*

Nous pouvons ramener le problème de Post à la recherche du secret $f(s,s)$ à l'aide des règles d'intrus définies dans la figure 9.1. On définit $T = \cup_i \{u_i(s), v_i(s)\}$.

Pour mieux comprendre le codage de Post en utilisant ces règles, prenons un exemple. $u_1 = a$, $u_2 = ba$, $v_1 = ab$ et $v_2 = a$. L'idée est que l'intrus va construire la solution du problème de Post en utilisant la dernière règle de la figure 9.1 puis utiliser la seconde règle pour construire $f(x(s), x(s))$ où x est la solution de Post et enfin décomposer la solution en utilisant la première règle. Sur l'exemple, on considère $x = aba$. Voici une preuve de $T \vdash f(s,s) \;\; [\![]\!]$:

$$\cfrac{\cfrac{\cfrac{\cfrac{T \vdash ba(s) \;\; [\![]\!]}{T \vdash aba(s) \;\; [\![]\!]}\,\mathcal{S}}{T \vdash f(aba(s), aba(s)) \;\; [\![]\!]}\,\mathcal{S}}{T \vdash f(ba(s), a(s)) \;\; [\![]\!]}\,\mathcal{S}}{T \vdash f(s,s) \;\; [\![]\!]}\,\mathcal{S}$$

Notons qu'il est possible de remplacer la dernière règle par les constructeurs a et b. Nous le ferons par la suite afin de limiter la profondeur des règles de $\mathcal{S}$.

Lemme 9.1 *La recherche d'une solution au problème de correspondance Post se réduit à la recherche du secret $f(s,s)$ à partir de la connaissance initiale T avec les règles de la figure 9.1 dans $\overline{\mathcal{S}}$. La problème de recherche du secret dans $\overline{\mathcal{S}}$ est donc indécidable.*

Preuve. Supposons qu'il existe une preuve de $T \vdash f(s,s) \;\; [\![]\!]$ (sans règle de protocole, les contraintes sont toujours vides) pour une instance donnée du problème de correspondance de Post. Seules deux règles permettent d'obtenir un terme ayant f comme symbole de tête : la première et la seconde.

Supposons que ce soit la seconde qui ait été utilisée. Cela signifie que l'on a une preuve de $T \vdash s \ [\![]\!]$. Aucune règle ne permet d'obtenir un terme sans symbole de tête. De plus, il est facile de montrer que $T \not\vdash s$. On arrive donc à une contradiction car cette preuve ne peut pas non plus être obtenue avec une règle d'axiome. La preuve de $T \vdash f(s,s) \ [\![]\!]$ est donc obtenue via la première règle et on a donc une preuve de $T \vdash f(u_i(s), v_i(s)) \ [\![]\!]$ pour un certain $i \leq n$. Cette preuve est obtenue soit par la première règle, soit par la seconde. Tant que la première règle est appliquée, on obtient des preuves de $T \vdash f(u_{i_1} \dots u_{i_l}(s), v_{i_1} \dots v_{i_l}(s)) \ [\![]\!]$. Lorsque l'on peut appliquer la seconde règle, on a alors $u_{i_1} \dots u_{i_l}(s) = v_{i_1} \dots v_{i_l}(s)$, c'est à dire une solution du problème de Post.

Réciproquement, si l'on suppose que $(u_{i_1} \dots u_{i_l}, v_{i_1} \dots v_{i_l})$ est une solution du problème, alors on peut construire la preuve suivante :

$$\dfrac{\dfrac{\dfrac{\begin{array}{c} T \vdash u_{i_l}(s) \ [\![]\!] \\ \vdots\, \mathcal{S} \\ T \vdash u_{i_1} \dots u_{i_l}(s) \ [\![]\!] \end{array}}{T \vdash f(u_{i_1} \dots u_{i_l}(s), v_{i_1} \dots v_{i_l}(s)) \ [\![]\!]}\,\mathcal{S}}{\begin{array}{c} T \vdash f(u_{i_2} \dots u_{i_l}(s), v_{i_2} \dots v_{i_l}(s)) \ [\![]\!] \\ \vdots\, \mathcal{S} \\ T \vdash f(u_{i_l}(s), v_{i_l}(s)) \ [\![]\!] \end{array}}\,\mathcal{S}}{T \vdash f(s,s) \ [\![]\!]}\,\mathcal{S}$$

Il est donc possible de réduire le problème de Post à la recherche de preuve dans $\overline{\mathcal{S}}$.

□

Le lemme ci-dessus nous force donc à prendre des hypothèses supplémentaires sur les règles de $\mathcal{S}$. Nous allons énumérer les hypothèses prises dans la suite de ce chapitre.

9.2 Propriété de localité

9.2.1 Motivation

Le codage de Post présenté dans la section précédente fait appel à des termes arbitrairement grands par rapport au secret s à trouver. Dans [RT01] ou [CKRT03b], une borne est placée sur la taille des termes : s'il existe une preuve utilisant des termes en dehors de cette borne, alors il est possible de simplifier cette preuve en une autre preuve utilisant des termes plus petits. En pratique, la preuve peut comporter des termes artificiellement gros construits par l'intrus et dont la structure n'a pas d'importance. Il est possible alors de remplacer ces termes par une constante.

Nous allons exiger l'existence d'une fonction F, calculable, travaillant sur les ensembles finis de termes et dont les images sont des ensembles finis de termes telle que pour tout ensemble fini de termes T, $T \subseteq F(T)$ et $F(F(T)) = F(T)$. Cette fonction doit de plus vérifier la propriété de localité définie par la suite.

Supposer que $\vdash_{\mathcal{S}}$ est décidable est insuffisant : on peut effectuer le même codage que dans le paragraphe précédent avec une seule règle de protocole et une seule session en enlevant la règle qui permet de déduire $f(x,x)$ des règles de $\mathcal{S}$ et en la transposant en une règle de protocole.

La propriété qui suit va nous amener à distinguer dans $\mathcal{S}$ les règles de construction des règles de décomposition. Intuitivement, les secondes permettent d'obtenir des termes contenus dans d'autres termes tandis que les premières composent un terme englobant les hypothèses. En réalité, la distinction est plus subtile et la définition exacte ne peut se faire que par le choix de la fonction F.

9.2.2 Définition

Définition 9.2 (Propriété de localité) *Pour toute preuve $T \vdash s$ dans $\mathcal{S}$, il existe une preuve Π de $T \vdash s$ dans $\mathcal{S}$ telle que :*
- *soit la dernière règle de Π est une règle de construction et pour tout séquent $T \vdash u$ apparaissant dans Π, $u \in F(T \cup \{s\})$,*
- *soit la dernière règle de Π n'est pas une règle de construction et tout séquent $T \vdash u$ est tel que $u \in F(T)$.*

9.2.3 Pertinence de la propriété

La plupart des théories qui nous intéressent vérifient la propriété de localité 1. Celle-ci est suffisante pour décider $\vdash_{\mathcal{S}}$: on calcule d'abord $F(T \cup \{s\})$ et on utilise alors un algorithme de point fixe pour déduire successivement les termes déductibles en une étape des termes déjà connus. Cette propriété est une généralisation des *théories locales* de [McA93].

Lemme 9.2 ([CLS03, CKRT03b]) *Dans le cas de Dolev-Yao, du « ou exclusif » et des groupes abéliens, il existe une fonction F qui satisfait la propriété de localité et qui est close par itération. De plus $|F(T)| = O(|T|)$.*

La fonction F en question est la fonction renvoyant l'ensemble des sous-termes dans le cas de Dolev-Yao, la fonction renvoyant les sous-termes en considérant l'opérateur $\oplus$ comme étant variadique pour le « ou exclusif » et la fonction telle que $t \in F(T) \implies I(t) \downarrow \in F(T)$, clôturée par sous-termes (de la même façon que pour le « ou exclusif » dans le cas des groupes abéliens.

Ce lemme a été prouvé sans l'utilisation des variants. A priori, rien ne semble garantir que la propriété de localité s'applique encore lorsque l'on travaille avec les variants. Nous devons donc compléter ce lemme pour inclure le cas des théories avec variants.

Lemme 9.3 *Pour une théorie donnée $\mathcal{S}, \epsilon$, s'il existe une fonction F qui satisfait la propriété de localité et qui est close par itération, alors il existe une fonction F' satisfaisant la propriété de localité et close par itération pour la théorie $\mathcal{S}', \epsilon'$ telle que ϵ convergent modulo ϵ', ϵ' classes d'équivalence finies et $\mathcal{S}'$ sont les variants de $\mathcal{S}$.*

Preuve. Nous reprenons les notations de la section 6.3 page 55 en ce qui concerne le symbole de normalisation. Nous définissions les images de F' comme étant les images de F sur lesquelles la fonction de normalisation $\downarrow_{\overline{\mathcal{S}}}$ a été appliquée. Cette fonction est également close par itération.

Étant donné une preuve Π de $T \vdash s$ utilisant les variants des règles de $\mathcal{S}$, il est possible de la transformer en une preuve Π' de $T \vdash s$ qui n'utilise pas les variants mais dont les termes sont normalisés. Π' utilise les mêmes règles que Π, modulo les variants. Nous appliquons la propriété de localité sur cette preuve Π' : tous les termes sont contenus dans $F(T)$. Pour chaque terme t dans Π correspond un terme t' normalisé (par $\downarrow_{\widehat{\mathcal{S}}}$ dans Π'. Dans Π, t est également normalisé (en choisissant le bon variant), mais selon $\downarrow_{\overline{\mathcal{S}}}$. Comme $t' \in F(T)$, $t' \downarrow_{\overline{\mathcal{S}}} \in F'(T)$. La propriété de localité est donc vérifiée. □

9.2.4 Contribution à la décidabilité

En quoi cette propriété est nécessaire ? Les règles de $\mathcal{S}$ de la figure 9.1 page 72 ne vérifient pas cette propriété. Il s'agit donc d'un premier pas vers la décidabilité. En effet, la fonction F doit être calculable. Considérons que c'est le cas.

Supposons qu'il existe une solution à une instance donnée du problème de correspondance de Post. Dans ce cas, on a vu auparavant que l'on pouvait construire une preuve de $T \vdash f(s,s)\ [\![]\!]$. La propriété de localité nous indique qu'il est possible de transformer cette preuve (qui n'utilise que des règles de $\mathcal{S}$) de façon à ce que tous les termes soient dans $F(T, f(s,s))$ ou $F(T)$ (selon le type de la dernière règle de la preuve). Dans les deux cas, F étant calculable et son domaine étant fini, il suffit d'énumérer $F(T, f(s,s))$ ou $F(T)$ pour trouver la solution du problème de correspondance de Post. On a donc une procédure de décision pour Post, ce qui est contradictoire. La fonction F, si elle existe, n'est donc pas calculable. Les règles de la figure 9.1 ne vérifient donc pas la propriété de localité.

9.2.5 Indécidabilité

Cette propriété n'est toutefois pas suffisante pour rendre la recherche du secret dans $\overline{\mathcal{S}}$ décidable. En effet, il est possible de remplacer la seconde règle de $\mathcal{S}$ fournissant le secret par une règle de protocole qui ne sera utilisée qu'une fois. L'indécidabilité est donc toujours d'actualité pour un nombre borné de sessions. Les règles utilisées pour réduire Post sont indiquées dans la figure 9.2.

$$\frac{T \vdash f(u_i(x), v_i(y))\ [\![E]\!]}{T \vdash f(x,y)\ [\![E]\!]}\ \mathcal{S}$$

$$\frac{T \vdash u\ [\![E]\!]}{T \vdash f(x,x)\ [\![E \wedge x = u \wedge x_{R,1,s} = 1]\!]}\ \mathcal{P}$$

$$\frac{T \vdash x\ [\![E]\!]}{T \vdash a(x)\ [\![E]\!]}\ \mathcal{S}$$

$$\frac{T \vdash x\ [\![E]\!]}{T \vdash b(x)\ [\![E]\!]}\ \mathcal{S}$$

FIG. 9.2 - *Règles de $\overline{\mathcal{S}}$ pour la réduction de Post*

De plus, nous avons remplacé les deux constructeurs par une forme plus simple, lettre par lettre plutôt que mot par mot. Ces deux règles n'assuraient de toute façon pas de contrôle sur la forme des mots construits. Elles permettaient simplement d'obtenir une preuve plus courte.

La preuve d'indécidabilité de la recherche du secret avec de telles règles est identique à la preuve utilisant les règles de la figure 9.1. Ces règles respectent la propriété de localité. Il suffit de décider que F est la fonction sous-termes et de la clore par la relation suivante : $f(u_i(x), v_i(y)) \in F(T) \implies f(x, y) \in F(T)$. Autrement dit, $F(T)$ est le plongement des termes dans T.

Les règles de $\mathcal{S}$ sont considérées comme des règles de composition et il est alors facile de montrer par récurrence que la propriété de localité est respectée : en effet, si la conclusion commence par le symbole f, alors, seule la première règle a pu être appliquée. Cela conduit à montrer que si la conclusion commence par un symbole f, alors la preuve est une tour d'application de la première règle (ou alors, une simple application de la règle d'axiome) dont l'hypothèse est $f(m_1(x), m_2(y))$ avec m_1 et m_2 des compositions de a et b. F étant clôturée de telle façon que $f(u_i(x), v_i(y)) \in F(T) \implies f(x, y) \in F(T)$, tous les termes sont bien dans $F(T)$. Si la conclusion de la preuve ne commence pas par f, il s'agit soit de l'application de la règle d'axiome, soit de l'application d'une des deux dernières règles. Dans ce cas, celles-ci sont considérées comme des compositions et la clôture de F par les sous-termes permet de conclure.

9.3 Forme des règles de construction et de décomposition

La section 9.2.5 nous apporte une preuve de l'indécidabilité du système $\overline{\mathcal{S}}$, même soumis à la propriété de localité. Nous allons donc poser des hypothèses supplémentaires sur la forme des règles de construction et de décomposition.

9.3.1 Forme des règles de construction

Ainsi, les règles de construction consisteront uniquement en l'ajout d'un symbole en tête :

$$\frac{T \vdash u_1 \ \llbracket E_1 \rrbracket \ldots T \vdash u_n \ \llbracket E_n \rrbracket}{T \vdash f(u_1, \ldots, u_n) \ \llbracket E_1 \wedge \ldots \wedge E_n \rrbracket}\,\mathcal{C}$$

Cette hypothèse écarte malheureusement des théories comme le « ou exclusif » ou les groupes abéliens. Il est possible de l'alléger de façon à accepter de telles théories, au prix d'une complexification de la preuve du théorème de normalisation de preuves : avec une telle hypothèse, les règles de constructions y sont traitées de manière très simple comparativement aux règles de décomposition qui nécessitent une certaine gymnastique. Toutefois, en examinant le cas des décompositions, on pourra généraliser graduellement le cas des constructions. Nous y reviendrons par la suite.

La première règle de la figure 9.2 page précédente ne vérifie pas cette hypothèse alors que les deux dernières la vérifient. Nous considérerons donc la première règle comme une règle de décomposition et les deux dernières comme une règle de construction.

9.3.2 Forme des règles de décomposition

La preuve d'indécidabilité étant toujours de mise, nous devons désormais faire des hypothèses supplémentaires sur les décompositions.

$$\frac{T \vdash t_1 \ \llbracket E_1 \rrbracket \ldots T \vdash t_n \ \llbracket E_n \rrbracket}{T \vdash t \ \llbracket E_1 \wedge \ldots \wedge E_n \rrbracket}\,\mathcal{D}$$

Une règle de décomposition, définie comme ci-dessus, doit vérifier l'une des deux conditions suivantes :

1. chaque t_i est de profondeur au plus 1.
2. chaque t_i est de profondeur au plus 2 et pour chaque t_i de profondeur 2,
   - soit t est une variable de t_i,
   - soit t est un sous-terme de t_i et t est de profondeur 1,
   - soit $t_i = C[f(u_1, \ldots, u_m)]$ et $t = C[u_i]$, où C est un contexte et f un symbole de construction.

De plus, la conclusion d'une règle ne peut être égale à l'une des prémisses. Ce type de règle de décomposition est de toutes les façons inutile.

La première règle de la figure 9.2 ne vérifie pas la première condition, ni la seconde. $f(x, y)$ est un terme de profondeur 1 mais n'est pas un sous-terme de $f(u_i(x), v_i(y))$. Il n'y a pas de contexte C tel que le dernier point soit vérifié.

Par contre, Dolev-Yao vérifie les conditions demandées, ainsi que la théorie des clefs inverses ou encore la théorie « préfixe » qui est définie par la règle de réécriture $f(\langle x, y\rangle) \to f(x)$. Elle correspond au cas du chiffrement symétrique en mode CBC. Dans ce cas, f est l'opérateur de chiffrement en question. Le cas du chiffrement symétrique en mode ECB est par contre plus délicat à traiter et est de toute façon rarement utilisé en raison de ses défauts intrinsèques.

Nous prétendons ensuite que tout système $\overline{\mathcal{S}}$ vérifiant à la fois les restrictions de cette section et la propriété de localité de la section 9.2 page 73 est décidable. C'est le rôle du théorème présenté dans le chapitre suivant.

9.3.3 Indécidabilité des systèmes à profondeur 2

La forme des règles de décomposition présentée ci-dessus peut sembler arbitraire. Nous allons montrer dans cette section que dans un système admettant des règles de décomposition arbitraires à profondeur 2, le secret est indécidable. Pour ce faire, nous allons introduire un certain nombre de symboles de fonctions supplémentaires dans le système.

Par convention, nous notons u_{ij} le j-ième caractère du mot u_i. Si u_i est de longueur strictement inférieure à j, u_{ij} est vide. Le codage de Post est présenté dans la figure 9.3 page suivante. Si u_{ij} est vide, $u_{ij}(x) = x$. On note n_j le maximum entre la longueur de u_j et de v_j.

La première règle de la figure 9.2 page 75 a été décomposée en les trois premières règles de la figure 9.3. La première règle permet d'initier la lecture du couple u_i, v_i, la seconde règle permet de lire lettre par lettre le couple choisi tandis que la troisième règle marque la fin de la lecture, avec succès, du couple.

Cette réduction au problème de Post justifie ainsi la limitation aux décompositions de profondeur 2 et le fait d'imposer des restrictions supplémentaires sur celles-ci.

9.3.4 Retardement des instanciations

Les hypothèses supplémentaires invoquées ici vont nous permettre de préciser davantage à quel point il est possible de retarder les instanciations dans les preuves de $\overline{\mathcal{S}}$.

Avec l'hypothèse sur les règles de construction, on peut démontrer qu'il est possible de repousser les instanciations au-delà des constructions :

$$\frac{T \vdash f(x,y) \ \llbracket E \rrbracket}{T \vdash f_{i0}(x,y) \ \llbracket E \rrbracket} \mathcal{S}$$

$$\frac{T \vdash f_{ij}(u_{ij}(x), v_{ij}(y)) \ \llbracket E \rrbracket}{T \vdash f_{i,j+1}(x,y) \ \llbracket E \rrbracket} \mathcal{S}$$

$$\frac{T \vdash f_{in_j}(x,y) \ \llbracket E \rrbracket}{T \vdash f(x,y) \ \llbracket E \rrbracket} \mathcal{S}$$

$$\frac{T \vdash u \ \llbracket E \rrbracket}{T \vdash f(x,x) \ \llbracket E \wedge x = u \wedge x_{R,1,s} = 1 \rrbracket} \mathcal{P}$$

$$\frac{T \vdash x \ \llbracket E \rrbracket}{T \vdash a(x) \ \llbracket E \rrbracket} \mathcal{S}$$

$$\frac{T \vdash x \ \llbracket E \rrbracket}{T \vdash b(x) \ \llbracket E \rrbracket} \mathcal{S}$$

FIG. 9.3 - *Règles de $\overline{S}$ pour la réduction de Post*

Lemme 9.4 *Pour toute preuve de $\overline{S}$ dont les règles de construction consistent en l'ajout d'un symbole en tête, toute règle de construction peut retarder une instanciation (c'est à dire que la dernière règle de la figure 8.1 page 68 peut toujours être appliquée pour une règle de construction).*

Preuve. La dernière règle de la figure 8.1 est remplacée par celle de la figure 9.4.

$$\frac{T \vdash u_1 \ \llbracket E_1 \rrbracket \ldots \ \dfrac{T \vdash C[x]_p \ \llbracket E_i \rrbracket}{T \vdash C[u_i]_p \ \llbracket E_i \rrbracket} \mathcal{I} \ \ldots T \vdash u_n \ \llbracket E_n \rrbracket}{T \vdash f(u_1, \ldots, C[u_i]_p, \ldots u_n) \ \llbracket E \rrbracket} \mathcal{S} \qquad \Rightarrow$$

$$\frac{\dfrac{T \vdash u_1 \ \llbracket E_1 \rrbracket \ldots T \vdash C[x]_p \ \llbracket E_i \rrbracket \ldots T \vdash u_n \ \llbracket E_n \rrbracket}{T \vdash f(u_1, \ldots, C[x]_p, \ldots u_n) \ \llbracket E \rrbracket} \mathcal{S}}{T \vdash f(u_1, \ldots, C[u_i]_p, \ldots u_n) \ \llbracket E \rrbracket} \mathcal{I}$$

FIG. 9.4 - *Règle de réécriture pour retarder les instanciations vis-à-vis des règles de construction simplifiées*

□

Les hypothèses émises sur les règles de décomposition permettent de repousser toute instanciation trop profonde.

Lemme 9.5 *Pour toute preuve de $\overline{S}$ dont les règles de décomposition respectent les hypothèses de la section 9.3.2 page 76, toute règle de décomposition peut retarder les règles d'instanciation de la forme suivante :*

$$\frac{T \vdash C[x] \;\; [\![E \wedge x = u]\!]}{T \vdash C[u] \;\; [\![E \wedge x = u]\!]}\,\mathcal{I}$$

où C n'est pas une abstraction des prémises d'une règle de décomposition, c'est-à-dire qu'il n'existe pas de prémise u_i d'une règle de décomposition et de substitution σ portant sur les variables telles que $C[x]\sigma = u_i$.

Ce lemme permet de dégager deux types d'instanciations : celles qui sont utiles pour les règles de décomposition car elles instancient un terme utilisé ensuite dans la décomposition et celles qui ne le sont pas et que l'on peut donc appliquer après la décomposition. Il ne nécessite pas directement les hypothèses émises sur les règles de décomposition et pourrait également être étendu aux règles de construction si celles-ci ne commutaient pas déjà avec les règles d'instanciation. Cependant, grâce aux hypothèses sur les décompositions, l'instanciation est limitée à la profondeur 1 (ou 0).

Imaginons que l'on a la règle suivante comme règle de décomposition :

$$\frac{T \vdash \{x\}_{\mathrm{pub}(y)} \;\; [\![E_1]\!] \quad T \vdash \mathrm{priv}(y) \;\; [\![E_2]\!]}{T \vdash x \;\; [\![E_1 \wedge E_2]\!]}\,\mathcal{D}$$

Il est alors possible d'en déduire une règle d'instanciation qui appartient à la première catégorie :

$$\frac{T \vdash \{x\}_z \;\; [\![E \wedge z = u]\!]}{T \vdash \{x\}_u \;\; [\![E \wedge z = u]\!]}\,\mathcal{I}$$

De manière plus formelle, si une règle de décomposition a une prémise de la forme $g(t_1, \ldots, t_n)$, alors pour chaque i tel que t_i est de profondeur 1 ou plus, on peut construire l'instanciation suivante qui appartient à la première catégorie :

$$\frac{T \vdash g(u_1, \ldots, u_n) \;\; [\![E' \wedge E]\!]}{T \vdash g(v_1, \ldots, v_n) \;\; [\![E' \wedge E]\!]}\,\mathcal{I}$$

où $u_i = x_i$, $v_i = w_i$, $u_j = t_j$ et $v_j = t_j$ pour $j \neq i$; E est $x_i = w_i$.

Preuve.

On considère la dernière règle de la figure 8.1 page 68 dans le cas où C n'est pas le résultat de l'abstraction d'une prémise, en particulier, C n'est pas une abstraction de la prémise à la position i. Comme $C[u_i]_p$ est tout de même unifiable avec la prémise, cela signifie que $C[x]_p$ est également unifiable avec la prémise (pour tout x, donc en particulier pour celui de la règle de réécriture). La linéarisation nous permet donc d'appliquer la règle de décomposition directement sur $C[x]_p$ donnant un terme u' tel qu'il existe une suite d'instanciation (éventuellement vide) conduisant à u et dont E_i (et donc E) est le modèle.

□

CHAPITRE 10

Théorème

Sommaire

Nous allons énoncer dans ce chapitre le principal théorème de cette thèse. Ce dernier permet de transformer un arbre de preuve en un arbre de preuve dont les éléments sont suffisamment petits. Ce théorème servira ensuite de base pour la partie III et permettra de concevoir un algorithme de décision pour un nombre borné de sessions et un algorithme de recherche d'attaque dans le cas d'un nombre non borné de sessions.

10.1 Énoncé du théorème principal

10.1.1 Définition préliminaire

Un terme suffisamment petit est intuitivement un sous-terme des termes initiaux et des termes contenus dans les règles. En pratique, on utilisera la fonction F de la propriété de localité de la section 9.2 page 73 : on lui demandera d'être $F(V,T,\Sigma,s,\xi),E$-admissible où s est le secret.

Voici la définition de U,E-admissible pour un séquent contraint :

Définition 10.1 *Un séquent contraint $T \vdash s\ [\![G]\!]$ est dit U,E-*admissible, *U étant un ensemble de termes et G un ensemble de contraintes sous forme résolue, si et seulement si :*

1. *pour tout $x = t$ dans G, si x est minimal dans l'ordre d'occurence de G, alors $T \vdash t\ [\![G \setminus \{x = t\}]\!]$ est U,E-admissible,*
2. *$s \in U$ ou bien il existe t tel que $s =_E t$ et $t \in U$.*

Posons :

$$\begin{array}{rcl} U & = & \{a, b, \{a\}_b, c, d, \langle c, d\rangle\} \\ E & = & \{x = \{a\}_b, y = \langle c, d\rangle, z = c\} \end{array}$$

Les séquents suivant sont alors $U,E-$admissibles :

$$\begin{array}{ll} T \vdash a\ [\![]\!] & T \vdash \{a\}_b\ [\![]\!] \\ T \vdash x\ [\![]\!] & T \vdash \langle z, d\rangle\ [\![]\!] \\ T \vdash x\ [\![w = \langle c, d\rangle \wedge x = \{a\}_b]\!] & T \vdash \{a\}_b\ [\![w = \langle z, d\rangle]\!] \end{array}$$

Cette définition nous permet alors d'énoncer le théorème. On considère que le système $\overline{S}$ vérifie les propriétés indiquées dans le chapitre 9 page 71.

10.1.2 Énoncé

Voici l'énoncé du théorème principal :

Théorème 10.1 *S'il existe une preuve Π de $T \vdash s\ [\![E]\!]$ dans $\overline{S}$, défini dans la figure 6.1 page 56 et auquel on impose la propriété de localité 9.2 page 74 et les restrictions sur les règles de composition et de décomposition telles que définies dans les sections 9.3.1 page 76 et 9.3.2 page 76, telle que E est un ensemble d'équations satisfaisable et sous forme résolue, utilisant l'ensemble des règles de protocole V dont Σ est l'ensemble des affectations initiales, alors il existe une preuve Π' de $T \vdash s'\ [\![E']\!]$ dans $\overline{S}$ telle que $s'\sigma_{E'} = s\sigma_{E'}$ et telle que tout séquent $T \vdash t\ [\![E'']\!]$ dans Π' soit $F(V,T,\Sigma,s,\xi),E'$-admissible.*

Pourquoi ne pas se contenter d'imposer aux séquents d'être dans $F(V,T,\Sigma,s,\xi)$? Imaginons que $T = \{a\}$ et que l'on dispose d'une règle de protocole $f(x) \Rightarrow g(x)$ et d'une règle de composition permettant de construire $f(y)$ à partir de y. Pour obtenir le secret $g(x)$, la preuve sera la suivante :

$$\cfrac{\cfrac{\cfrac{}{T \vdash a\ [\![]\!]}\ \mathcal{A}}{T \vdash f(a)\ [\![]\!]}\ \mathcal{C}}{T \vdash g(x)\ [\![x = a \wedge x_{R,1,s} = 1]\!]}\ \mathcal{P}$$

Or dans cette preuve, le terme $f(a)$ apparaît et si on considère F comme la fonction sous-terme, il ne se trouve pas dans $F(V,T,\Sigma,s,\xi)$. Il paraît cependant raisonnable de le considérer comme suffisamment petit pour être considéré. Cet exemple explique l'utilisation de contraintes $F(V, T, \Sigma, s, \xi), E'$-admissible. Ajoutons que la « taille » d'un terme est relatif à la contrainte finale et non à la contrainte du séquent auquel il appartient car, dans notre exemple, $f(a)$ apparaît avec une contrainte vide.

10.2 Lemmes additionnels

Avant d'entreprendre la preuve du théorème principal, nous introduisons quelques lemmes supplémentaires.

Le premier lemme va nous permettre de montrer que si un terme est prouvable sous la contrainte E, il en est de même pour ses prémisses.

Lemme 10.1 *S'il existe une preuve Π de $T \vdash u\ \llbracket E \rrbracket$ dans $\overline{\mathcal{S}}$ dont les hypothèses sont $T \vdash u_i\ \llbracket E_i \rrbracket$, contenant uniquement des règles de $\mathcal{S}$, alors il existe une preuve Π' dans $\overline{\mathcal{S}}$ de $T \vdash u_1\ \llbracket E \rrbracket$ utilisant les mêmes hypothèses.*

Preuve. Chaque règle de $\mathcal{S}$ est transformée (à l'exception des règles d'axiome) en utilisant la règle de réécriture de la figure 10.1.

$$\frac{T \vdash u_1\ \llbracket E_1 \rrbracket \quad \cdots \quad T \vdash u_n\ \llbracket E_n \rrbracket}{T \vdash u\ \llbracket E_1 \wedge \ldots \wedge E_n \rrbracket}\,\mathcal{S}$$

$$\Longrightarrow \quad \cfrac{\cfrac{\cfrac{T \vdash u_1\ \llbracket E_1 \rrbracket \quad T \vdash u_2\ \llbracket E_2 \rrbracket}{T \vdash u_1\ \llbracket E_1 \wedge E_2 \rrbracket}\,\mathcal{W} \qquad T \vdash u_3\ \llbracket E_3 \rrbracket}{\vdots}\,\mathcal{W}}{T \vdash u_1\ \llbracket E_1 \wedge \ldots \wedge E_n \rrbracket}\,\mathcal{W}$$

FIG. 10.1 - *Règle de réécriture du lemme 10.1*

Le système de réécriture termine en raison du fait que le nombre de règles de $\mathcal{S}$ décroît strictement. Les règles de $\mathcal{S}$ sont transformées en des règles d'affaiblissement valides et le fait que l'on a une preuve de $\overline{\mathcal{S}}$ est un invariant. Les preuves sur lesquelles le système de réécriture n'a pas d'effet satisfont les conditions du lemme. $\square$

Le second lemme consiste à ordonner les règles d'une preuve contenant des affaiblissements et des règles de $\mathcal{S}$ de façon à regrouper ces dernières dans le bas de la preuve comme indiqué dans la figure 10.2 page suivante.

Lemme 10.2 *Toute preuve Π utilisant uniquement des règles de $\mathcal{S}$ et des règles d'affaiblissement peut être réécrite en une preuve Π' utilisant les mêmes hypothèses et la même conclusion de telle façon à ce qu'aucune règle de $\mathcal{S}$ ne soit suivie par une règle d'affaiblissement.*

Preuve.

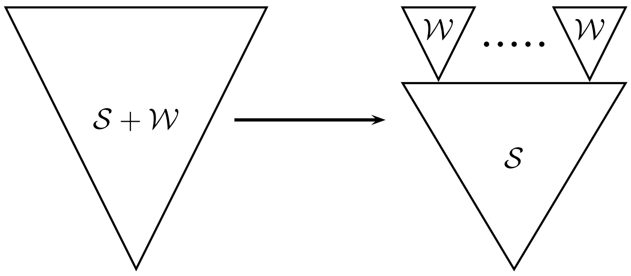

FIG. 10.2 - *Transformation effectuée par le lemme 10.2 page précédente*

$$\dfrac{T \vdash u_1 \;\; [\![E_1]\!] \qquad \dfrac{T \vdash u_{21} \;\; [\![E_{21}]\!] \quad \cdots \quad T \vdash u_{2k} \;\; [\![E_{2k}]\!]}{T \vdash u_2 \;\; [\![E_{21} \wedge \ldots \wedge E_{2k}]\!]}\,\mathcal{S}}{T \vdash u_1 \;\; [\![E_1 \wedge E_{21} \wedge \ldots \wedge E_{2k}]\!]}\,\mathcal{W}$$

$$\Longrightarrow \qquad \dfrac{T \vdash u_1 \;\; [\![E_1]\!] \qquad \dfrac{\begin{array}{c} \dfrac{\dfrac{T \vdash u_{21} \;\; [\![E_{21}]\!] \quad T \vdash u_{22} \;\; [\![E_{22}]\!]}{T \vdash u_{21} \;\; [\![E_{21} \wedge E_{22}]\!]}\,\mathcal{W} \qquad T \vdash u_{23} \;\; [\![E_{23}]\!]}{}\,\mathcal{W} \\ \vdots \end{array}}{T \vdash u_{21} \;\; [\![E_{21} \wedge \ldots \wedge E_{2k}]\!]}\,\mathcal{W}}{T \vdash u_1 \;\; [\![E_1 \wedge E_{21} \wedge \ldots \wedge E_{2k}]\!]}\,\mathcal{W}$$

FIG. 10.3 - *Première règle de réécriture pour le lemme 10.2 page précédente*

$$\dfrac{\dfrac{T \vdash u_{11} \;\; [\![E_{11}]\!] \quad \cdots \quad T \vdash u_{1k} \;\; [\![E_{1k}]\!]}{T \vdash u_1 \;\; [\![E_{11} \wedge \ldots \wedge E_{1k}]\!]}\,\mathcal{S} \qquad T \vdash u_2 \;\; [\![E_2]\!]}{T \vdash u_1 \;\; [\![E_{11} \wedge \ldots \wedge E_{1k} \wedge E_2]\!]}\,\mathcal{W}$$

$$\Longrightarrow \qquad \dfrac{\dfrac{T \vdash u_{11} \;\; [\![E_{11}]\!] \quad T \vdash u_2 \;\; [\![E_2]\!]}{T \vdash u_{11} \;\; [\![E_{11} \wedge E_2]\!]}\,\mathcal{W} \qquad T \vdash u_{12} \;\; [\![E_{12}]\!] \quad \cdots \quad T \vdash u_{1k} \;\; [\![E_{1k}]\!]}{T \vdash u_1 \;\; [\![E_{11} \wedge \ldots \wedge E_{1k} \wedge E_2]\!]}\,\mathcal{S}$$

FIG. 10.4 - *Seconde règle de réécriture pour le lemme 10.2 page précédente*

On applique le système de réécriture présenté dans les figures 10.3 et 10.4.

Ce système de réécriture termine. En effet, si l'on caractérise une preuve par le couple (N, M) où N est le nombre de règles de $\mathcal{S}$ et M est le multi-ensemble du nombre de règles de $\mathcal{S}$ dans les sous-preuves se terminant par des règles d'affaiblissement, l'application de la première règle permet de faire décroître strictement N et la seconde règle laisse N constant tandis que M décroît de manière stricte.

À chaque application de l'une des deux règles de réécriture, on transforme une preuve de $\overline{\mathcal{S}}$ en une preuve de $\overline{\mathcal{S}}$. Les hypothèses et la conclusion sont également préservés. Enfin, une preuve irréductible pour ce système de réécriture vérifie les conditions du lemme. □

10.3 Cas de Dolev-Yao

Afin de mieux comprendre la preuve qui va suivre dans les sections suivantes, nous allons démontrer dans un premier temps le théorème dans le cas de Dolev-Yao.

10.3.1 Définition de $\overline{\mathcal{S}}$ dans le cas de Dolev-Yao

La figure 10.5 page suivante indique les règles que l'on utilise dans le cas de Dolev-Yao.

Dans le cas de Dolev-Yao, nous pouvons nous limiter aux instanciations à profondeur 0. En effet, toutes les autres instanciations commutent, à l'aide du lemme 8.3 page 68, avec les règles de composition et de décomposition mises en œuvre. Il est donc possible de repousser toutes les instanciations à profondeur autre que 0 en bas de la preuve. Ainsi, dans le cas Dolev-Yao, on se contentera de normaliser la preuve obtenue sans ces instanciations (et donc il n'y aura pas d'instanciations autres qu'à profondeur 0).

Les compositions sont les règles « paire » et « chiffrement » et consistent bien en l'ajout d'un symbole en tête, comme imposé dans la section 9.3.1 page 76. Les règles de déchiffrement respectent les contraintes de la section 9.3.2 page 76 car la conclusion est dans les deux cas une variable de l'une des prémisses. Enfin, la propriété de localité 9.2 page 74 est bien respectée par ce système. La fonction F est alors la fonction « sous-terme ».

10.3.2 Hypothèse de récurrence

L'hypothèse de récurrence est paramétrée par les éléments suivants :
- E_f une contrainte sous forme factorisée et
- s_f un terme.

On définit la fonction ρ dont le domaine est le l'ensemble des termes et dont l'image est $F(V, T, \Sigma, s_f, \xi)$ et telle que pour tout terme t, s'il existe $u \in F(V, T, \Sigma, s_f, \xi)$ tel que u n'est pas une variable et $t =_{E_f} u$, alors $\rho(t) = t$; sinon, $\rho(t) = 0$.

Nous optons pour l'hypothèse de récurrence sur la taille de Π suivante : soit Π une preuve de $T \vdash s$ $[\![E]\!]$ telle que $E_f \models E$. Il existe alors une preuve Π_χ de $T \vdash s$ $[\![E']\!]$ avec $E' = \bigwedge_{x=t \in E} x = \rho(t)$ telle que Π_χ est une preuve de $\overline{\mathcal{S}}$ et, en notant $E'_f = \bigwedge_{x=t \in E_f} x = \rho(t)$:

1. pour tout $x = t \in E'$, si $\rho(t) = 0$ alors il existe une preuve terminant par une construction de $T \vdash t$ $[\![E]\!]$ plus petite que Π et dans $\overline{\mathcal{S}}$.
2. pour toute sous-preuve π de Π_χ dont la conclusion est $T \vdash v$ $[\![G]\!]$, tout séquent $T \vdash u$ $[\![G']\!]$ de π est $F(V, T, \Sigma, v, \xi), E'_f$-admissible si la dernière règle de π la plus

On se reportera aux remarques concernant la figure 6.1 page 56 en ce qui concerne la forme exacte des contraintes.

Déchiffrement

$$\frac{T \vdash \{x\}_y \;\; [\![E_1]\!] \quad T \vdash z \;\; [\![E_2]\!]}{T \vdash x \;\; [\![E_1 \wedge E_2]\!]} \text{ si } y = z$$

Projections

$$\frac{T \vdash \langle x, y \rangle \;\; [\![E]\!]}{T \vdash x \;\; [\![E]\!]} \qquad \frac{T \vdash \langle x, y \rangle \;\; [\![E]\!]}{T \vdash y \;\; [\![E]\!]}$$

Chiffrement, paire

$$\frac{T \vdash x \;\; [\![E_1]\!] \quad T \vdash y \;\; [\![E_2]\!]}{T \vdash \{x\}_y \;\; [\![E_1 \wedge E_2]\!]} \qquad \frac{T \vdash x \;\; [\![E_1]\!] \quad T \vdash y \;\; [\![E_2]\!]}{T \vdash \langle x, y \rangle \;\; [\![E_1 \wedge E_2]\!]}$$

Ouverture de session . Si $v \Rightarrow w$ est la première règle du rôle R, σ_s est l'ensemble des affectations initiales propres à la session s et $E \not\models_{\mathcal{E}} x_{R,1,s} = 1$, alors :

$$\frac{T \vdash u \;\; [\![E]\!]}{T \vdash w \;\; [\![u = v \wedge E \wedge \sigma_s \wedge x_{R,1,s} = 1]\!]} \mathcal{P}$$

Avancement de session . Si $v \Rightarrow w$ est la règle numérotée $k > 1$ du rôle R, $E \not\models_{\mathcal{E}} x_{R,k,s} = 1$ et $E \models_{\mathcal{E}} x_{R,k-1,s} = 1$, alors :

$$\frac{T \vdash u \;\; [\![E]\!]}{T \vdash w \;\; [\![u = v \wedge E \wedge x_{R,k,s} = 1]\!]}$$

Agent compromis . Si $E \models_{\mathcal{E}} x_{R,1,s} = 1$ et l'acteur principal de la session s est compromis, alors :

$$\frac{T \vdash u \;\; [\![E]\!]}{T \vdash N_i(s) \;\; [\![E]\!]}$$

Règle d'instanciation

$$\frac{T \vdash x \;\; [\![E \wedge x = u]\!]}{T \vdash u \;\; [\![E \wedge x = u]\!]}$$

Règle d'affaiblissement

$$\frac{T \vdash u_1 \;\; [\![E_1]\!] \quad T \vdash u_2 \;\; [\![E_2]\!]}{T \vdash u_1 \;\; [\![E_1 \wedge E_2]\!]}$$

FIG. 10.5 - *Règles de définition du système $\overline{\mathcal{S}}$ dans le cas de Dolev-Yao*

à gauche qui n'est pas une règle d'affaiblissement est une règle de composition ou $F(V, T, \Sigma, \xi), E'_f$-admissible dans le cas contraire.

Prouver le théorème principal dans le cas de Dolev-Yao revient à prouver l'hypothèse de récurrence avec E_f et s_f tels que $T \vdash s_f \ \llbracket E_f \rrbracket$ soit la contrainte finale de la preuve.

10.3.3 Preuve

Nous procédons donc par récurrence sur la taille de Π. Le cas de base est immédiat car on conserve la preuve initiale Π qui satisfait déjà aux conditions demandées. Considérons maintenant la dernière règle de Π.

Règle d'affaiblissement . La preuve Π est obtenue par affaiblissement de deux preuves Π_1 et Π_2 :

$$\dfrac{\dfrac{\Pi_1}{T \vdash u_1 \ \llbracket E_1 \rrbracket} \quad \dfrac{\Pi_2}{T \vdash u_2 \ \llbracket E_2 \rrbracket}}{T \vdash u_1 \ \llbracket E_1 \wedge E_2 \rrbracket} \mathcal{W}$$

On applique l'hypothèse de récurrence sur Π_1 et Π_2 pour obtenir deux preuves Π^1_χ et Π^2_χ de, respectivement, $T \vdash u_1 \ \llbracket E'_1 \rrbracket$ et $T \vdash u_2 \ \llbracket E'_2 \rrbracket$ vérifiant l'hypothèse de récurrence. On note $E' = E'_1 \wedge E'_2$. Soit $x = t \in E'$. Si $\rho(t) = 0$, l'hypothèse de récurrence sur Π^1_χ ou Π^2_χ (selon que $x = t$ provienne de E'_1 ou E'_2, respectivement) permet de conclure qu'il existe une preive plus petite de $T \vdash t \ \llbracket E_1 \rrbracket$ (respectivement $T \vdash t \ \llbracket E_2 \rrbracket$). Cette preuve se termine par une construction. On affaiblit l'une des hypothèses avec la preuve de $T \vdash u_2 \ \llbracket E_2 \rrbracket$ (respectivement $T \vdash u_1 \ \llbracket E_1 \rrbracket$) puis on applique la règle de construction pour obtenir une preuve de $T \vdash u \ \llbracket E_1 \wedge E_2 \rrbracket$ se terminant par une règle de construction. Le premier point de l'hypothèse de récurrence est donc vérifiée si l'on utilise E' comme contrainte finale.

Nous allons maintenant construire une preuve vérifiant l'hypothèse de récurrence et dont la contrainte finale est E'.

Soit π le chapeau de preuve maximal de $T \vdash u_2 \ \llbracket E'_2 \rrbracket$ ne contenant que des règles de composition de $\mathcal{S}$ et des règles d'affaiblissement. On y applique le lemme 10.2 pour obtenir π'. Sur la partie de π' ne contenant que des règles de $\mathcal{S}$, nous appliquons le lemme 10.1 page 83 pour obtenir la preuve π''. Nous pouvons alors obtenir une preuve Π^3_χ de $T \vdash u_3 \ \llbracket E'_2 \rrbracket$ où u_3 est un terme issu de l'application du lemme 10.1. On applique la règle d'affaiblissement sur les preuves Π^1_χ et Π^3_χ :

$$\Pi_\chi = \dfrac{\dfrac{\Pi^1_\chi}{T \vdash u_1 \ \llbracket E'_1 \rrbracket} \quad \dfrac{\Pi^3_\chi}{T \vdash u_2 \ \llbracket E'_2 \rrbracket}}{T \vdash u_1 \ \llbracket E' \rrbracket} \mathcal{W}$$

$E' = E'_1 \wedge E'_2$ est bien égal à $\bigwedge_{x=t \in E_1 \wedge E_2} x = \rho(x,t)$ car $E'_1 = \bigwedge_{x=t \in E_1} x = \rho(x,t)$ et $E'_2 = \bigwedge_{x=t \in E_2} x = \rho(x,t)$ d'après l'hypothèse de récurrence.

Considérons ensuite le dernier point. Les notations sont rappelées dans la figure 10.6 page suivante. Soit P_1 une sous preuve de Π_χ dont la conclusion est $T \vdash v \ \llbracket c \rrbracket$ et $T \vdash u \ \llbracket c' \rrbracket$ un séquent de P_1. Si P_1 est enraciné dans Π^1_χ ou si elle est enracinée dans l'une des hypothèses de π'' (qui est un chapeau de Π^3_χ), on applique juste l'hypothèse de récurrence. Si P_1 est enracinée dans π'', par maximalité de π et suite à l'application

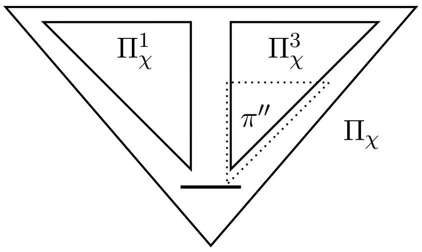

FIG. 10.6 - *Notations utilisées dans le cas d'une règle d'affaiblissement (cas Dolev-Yao)*

du lemme 10.1, la règle la plus à gauche qui n'est pas une règle d'affaiblissement n'est pas non plus une règle de composition. Si le séquent $T \vdash u\ [\![c']\!]$ n'est pas dans π'', il est $F(V,T,\Sigma,\xi), E'_f$-admissible grâce à l'hypothèse de récurrence. S'il est dans π'', il est composé d'un terme qui est celui de l'hypothèse la plus à gauche de la sous-preuve π'' et d'une contrainte qui est la combinaison des contraintes de certaines hypothèses de π''. Ces deux éléments réunis montrent que le séquent est $F(V,T,\Sigma,\xi), E'_f$-admissible par hypothèse de récurrence.

Enfin, si $P_1 = \Pi_\chi$, ou bien le séquent est dans Π^3_χ et vu ce qui a été dit ci-dessus, il est $F(V,T,\Sigma,\xi), E'_f$-admissible. Ou bien, il est dans Π^1_χ et si la règle la plus à gauche qui n'est pas une règle d'affaiblissement est une règle de composition, c'est aussi le cas pour la preuve Π_χ et vice-versa. Le séquent est donc $F(V,T,\Sigma,\xi), E'_f$-admissible (dans le premier cas), ou bien $F(V,T,\Sigma,s,\xi), E'_f$-admissible (dans le second cas) en utilisant l'hypothèse de récurrence. Si le séquent est la conclusion de Π_χ, il est la combinaison du terme constituant la conclusion s de Π^1_χ et de la contrainte E'. On sait déjà que le séquent $T \vdash s\ [\![E'_1]\!]$ est $F(V,T,\Sigma,s,\xi), E'_f$-admissible ou $F(V,T,\Sigma,\xi), E'_f$-admissible selon la nature de la règle la plus à gauche. Sachant que le séquent $T \vdash u_2\ [\![E'_2]\!]$ est $F(V,T,\Sigma,\xi), E'_f$-admissible, le séquent $T \vdash s\ [\![E']\!]$ est alors $F(V,T,\Sigma,\xi), E'_f$-admissible ou $F(V,T,\Sigma,s,\xi), E'_f$ selon la nature de la règle la plus à gauche. Dans tous les cas, le dernier point de l'hypothèse de récurrence est vérifié.

Règle de protocole . On considère la preuve suivante :

$$\Pi = \cfrac{\cfrac{\Pi_1}{T \vdash u\ [\![E]\!]}}{T \vdash w\ [\![u = v \wedge E \wedge x_{R,k,s} = 1 \wedge \sigma_S]\!]}\ \mathcal{P}$$

Dans le cas où la règle correspond à une progression de session, c'est-à-dire quand $k > 1$, σ_S est vide. Nous appliquons l'hypothèse de récurrence. Nous obtenons une preuve Π^1_χ de $T \vdash u\ [\![E']\!]$.

Soit u_i l'un des membres droits de la forme résolue de $u = v$, si $T \vdash u_i\ [\![]\!]$ n'est pas $F(V,T,\Sigma,\xi), E'_f$ admissible, nous allons remplacer u_i par 0. Pour ce faire, nous considérons le chapeau maximal π_1 de Π^1_χ ne contenant que des règles de composition et des règles d'affaiblissement. Nous appliquons le lemme 10.2 page 83 sur cette preuve pour obtenir la preuve π_2. Appelons π_3 la partie ne contenant que des règles de composition. Il existe un séquent de π_3 tel que u_i en soit le terme : en effet, les règles de composition consistent simplement en l'ajout d'un symbole en tête (cf section 9.3.1 page 76) et u_i est le résultat des compositions car les hypothèses de π_3 sont $F(V,T,\Sigma,\xi), E'_f$-admissibles grâce à l'hypothèse de récurrence. La preuve de $T \vdash u_i\ \ [\![E'']\!]$ peut être remplacée par la preuve de $T \vdash 0\ \ [\![]\!]$ car $0 \in T$. Toutefois, nous avons perdu l'ensemble des contraintes. E'' est la conjonction d'un certain nombre de contraintes de séquents que constituent les hypothèses de π_3. En appliquant la règle d'affaiblissement à la preuve de $T \vdash 0\ \ [\![]\!]$ et à chacun de ces séquents, nous obtenons une preuve de $T \vdash 0\ \ [\![E'']\!]$.

En itérant cette transformation sur chacune des positions contenant un u_i, nous obtenons alors une preuve de $T \vdash u'\ \ [\![E']\!]$ avec $u' = u[\bigcup_{(u_i,x_i)} \{u_{ip(u_i,x_i)} \leftarrow 0\}]$ où $p(u_i, x_i)$ est position de u_i dans u correspondant à la variable x_i dans v.

L'application de la règle de protocole sur cette preuve est toujours possible : u' s'unifie toujours avec v car nous avons remplacé des termes qui étaient liés aux variables de v et ceci, de manière uniforme : si $v = f(x,x)$, les deux termes t liés aux deux variables x ont été remplacés tous les deux par 0.

Nous obtenons la preuve Π_χ de $T \vdash w\ \ [\![\bigwedge_i x_i = \rho(u_i) \wedge E' \wedge x_{R,k,s} = 1 \wedge \sigma_S]\!]$.

Nous devons vérifier le premier point de la récurrence. Pour les variables autres que celles remplacées ici par 0, il suffit d'appliquer l'hypothèse de récurrence pour conclure. Prenons un $x = \rho(u_i)$ apparaissant dans la contrainte telle que $\rho(u_i) = 0$. Nous avons vu ci-dessus que nous avions une preuve de $T \vdash u_i\ \ [\![E'']\!]$ que nous pouvions transformer par affaiblissement en une preuve de $T \vdash u_i\ \ [\![E]\!]$. L'affaiblissement avec Π nous donne alors la preuve voulue pour vérifier le premier point de l'hypothèse de récurrence.

Passons désormais au dernier point de l'hypothèse de récurrence. Soit $T \vdash t\ \ [\![c]\!]$ un séquent de Π_χ. Si celui-ci est dans l'une des hypothèses enracinées dans π_3, par maximalité de π_1, la dernière règle est une composition et donc la récurrence nous garantit que le séquent considéré est $F(V,T,\Sigma,\xi), E'_f$-admissible. Si le séquent se trouve dans π_3, de par la forme des règles de composition, le terme du séquent est un sous-terme de u'. Or, ce terme est égal modulo E'_f à v qui est dans $F(V,T,\Sigma,\xi)$. La contrainte est la combinaison de contraintes apparaissant dans certains des séquents utilisés comme hypothèses de π_3. Ces séquents étant $F(V,T,\Sigma,\xi), E'_f$-admissible, le séquent que l'on considère l'est donc aussi. Enfin, si le séquent considéré est la conclusion de Π_χ, il s'agit de la combinaison du terme w qui est dans V et d'une contrainte qui est combinaison de :

- E' qui est la contrainte du séquent $T \vdash u'\ \ [\![E']\!]$ dont nous venons de voir qu'il est $F(V,T,\Sigma,\xi), E'_f$-admissible ;
- $\bigwedge x_i = \rho(u_i)$ dont les membres droits sont bien des termes égaux modulo E'_f à des termes de $F(V,T,\Sigma,\xi)$, par construction de ρ ;
- $x_{R,k,s} = 1$ avec $1 \in T$;
- σ_S dont les membre droits sont dans Σ par définition.

Ce séquent est donc également $F(V,T,\Sigma,\xi), E'_f$-admissible.

Les conditions de l'hypothèse de récurrence sont donc également vérifiées dans le cas où la dernière règle est une règle de protocole.

Règle de composition . Dans ce cas, aucune des prémisses ne se termine par une règle d'instanciation dont le contexte n'est pas vide. La preuve Π est de la forme suivante, avec $s = f(u_1, u_2)$ avec $f = \{\cdot\}_.$ ou $f = \langle\cdot,\cdot\rangle$:

$$\dfrac{\dfrac{\Pi_1}{T \vdash u_1 \ \llbracket E_1 \rrbracket} \quad \dfrac{\Pi_2}{T \vdash u_2 \ \llbracket E_2 \rrbracket}}{T \vdash s \ \llbracket E_1 \wedge E_2 \rrbracket}\ \mathcal{C}$$

On applique l'hypothèse de récurrence sur les preuves Π_1 et Π_2.

Nous obtenons alors les preuves Π_χ^1 et Π_χ^2 de $T \vdash u_1 \ \llbracket E_1' \rrbracket$ et $T \vdash u_2 \ \llbracket E_2' \rrbracket$. L'application de la règle de $\mathcal{S}$ sur ces preuves permet d'obtenir une preuve Π' du séquent contraint $T \vdash f(u_1, u_2) \ \llbracket E_1' \wedge E_2' \rrbracket$. Toutefois, la preuve obtenue n'est pas celle que nous cherchons. En effet, si l'une des preuves se termine par une composition ou une règle d'affaiblissement, nous devons appliquer la propriété de localité pour que le dernier point de l'hypothèse de récurrence soit vérifié.

Nous appliquons d'abord le lemme 10.2 page 83 sur la dernière règle de la preuve Π'. Le chapeau de preuve maximal π_1 ne contenant que des règles de $\mathcal{S}$ et des règles d'affaiblissement est transformée en un nouveau chapeau dont les règles sont réarrangées comme indiqué sur la figure 10.2 page 84. On considère la partie ne contenant que des règles de $\mathcal{S}$ et on y applique la propriété de localité pour obtenir la preuve π_3. Cette preuve prend la place de la sous-preuve ne contenant que des règles de $\mathcal{S}$ dans π_2 pour obtenir π_4. Cette dernière prend la place de π_1 dans Π'. On obtient ainsi la preuve Π_χ. La figure 10.7 illustre les transformations effectuées dans ce paragraphe.

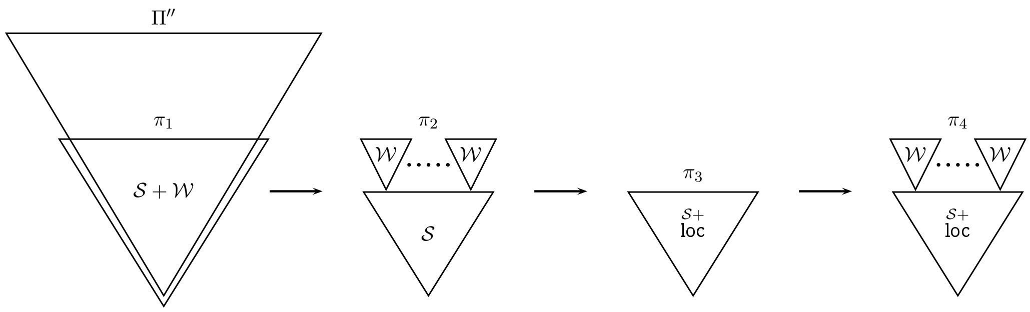

FIG. 10.7 - *Transformations effectuées sur la preuve dans le cas où la dernière règle est une règle de $\mathcal{S}$*

En ce qui concerne le premier point de l'hypothèse de récurrence, l'argumentaire est exactement le même que pour le cas où la dernière règle est une règle d'affaiblissement. Reste à considérer le dernier point.

Soit P_1 une sous-preuve de Π_χ dont la conclusion est $T \vdash v \ \llbracket c \rrbracket$ et $T \vdash u \ \llbracket c' \rrbracket$ un séquent de P_1. Si P_1 est enracinée en dehors de π_4, on utilise simplement l'hypothèse de récurrence pour conclure. Dans le cas contraire, si le séquent est en dehors de π_4, il fait partie d'une preuve enracinée dans l'une des hypothèses de π_4. Or, par maximalité de π_1, la conclusion des preuves enracinées dans les hypothèses de π_4 proviennent d'une règle d'axiome, d'une instanciation ou d'une règle de protocole. Le séquent est donc $F(V, T, \Sigma, \xi), E_f'$-admissible en utilisant l'hypothèse de récurrence.

Si le séquent est la conclusion d'une règle d'affaiblissement de π_4 (et donc, il n'est pas dans π_3), on se retrouve dans un cas similaire à celui où la dernière règle de Π est une règle d'affaiblissement : le séquent est la composition du terme d'un séquent utilisé en hypothèse de π_4 et de la conjonction d'un certain nombre de contraintes utilisées en hypothèse de π_4. Les séquents en question sont $F(V,T,\Sigma,\xi), E'_f$-admissible et donc ce séquent l'est également.

Si le séquent est dans π_3, nous devons exploiter la propriété de localité pour conclure. Par maximalité de π_1, les hypothèses de π_3 sont $F(V,T,\Sigma,\xi), E'_f$-admissible, notamment, les termes sont égaux modulo E'_f à des termes dans $F(V,T,\Sigma,\xi)$. La propriété de localité nous indique alors que tous les termes dans π_3 sont égaux modulo E'_f à des termes de $F(V,T,\Sigma,s,\xi)$ ou de $F(V,T,\Sigma,\xi)$ selon que la dernière règle la plus à gauche qui n'est pas une règle d'affaiblissement est ou n'est pas (respectivement) une règle de composition.

Les contraintes des séquents situés dans π_3 sont des combinaisons des contraintes des séquents des hypothèses qui étaient $F(V,T,\Sigma,\xi), E'_f$-admissible et donc les séquents de π_3 sont $F(V,T,\Sigma,s,\xi), E'_f$-admissible ou $F(V,T,\Sigma,\xi), E'_f$-admissible.

L'hypothèse de récurrence est alors vérifiée.

Règle de décomposition . Pour des raisons de simplicité, nous allons traiter uniquement le cas de la règle de déchiffrement. Le cas des règles de projection constituent un sous-cas plus simple (il n'y a pas de non-linéarités à traiter). La preuve que nous considérons est alors de la forme :

$$\dfrac{\dfrac{\Pi_1}{T \vdash \{u\}_v \ \ [\![E_1]\!]} \quad \dfrac{\Pi_2}{T \vdash w \ \ [\![E_2]\!]}}{T \vdash u \ \ [\![E_1 \wedge E_2]\!]} \; v = w$$

Nous appliquons l'hypothèse de récurrence sur Π_1 et Π_2 pour obtenir des preuves de $T \vdash \{u\}_v \ \ [\![E'_1]\!]$ et $T \vdash w \ \ [\![E'_2]\!]$. Nous voudrions simplement appliquer de nouveau la règle de déchiffrement pour obtenir une preuve de $T \vdash u \ \ [\![E'_1 \wedge E'_2]\!]$. Cependant, nous devons nous assurer des conditions d'appliquabilité de la règle de déchiffrement, notamment en raison de la condition de bord $v = w$ requise.

Si $E'_1 \wedge E'_2 \models v = w$, nous appliquons la règle de déchiffrement pour obtenir une preuve sur laquelle nous effectuons les mêmes transformations que dans le cas d'une règle de compositon. Le premier point de l'hypothèse de récurrence est vérifiée comme dans le cas d'une règle d'affaiblissement. Le second point est similaire au cas des règles de compositon, sauf que l'application de la propriété de localité qui a mené à la preuve π_3 (cf figure 10.7 page précédente) nous assure que nous travaillons qu'avec des séquents $F(V,T,\Sigma,\xi), E'_f$-admissibles au niveau de π_3 (et non, éventuellement, des séquents $F(V,T,\Sigma,s,\xi), E'_f$-admissibles).

Considérons maintenant le cas où $E'_1 \wedge E'_2 \not\models v = w$. Cela est dû à des remplacements par 0 de certains termes dans les contraintes E_1 et E_2. Traitons de manière exhaustive les cas possibles :

1. $v = x$ et $w = y$; dans ce cas, comme $E_1 \wedge E_2 \models x = y$ et que nous utilisons des contraintes sous forme factorisée, il existe une suite de variables $x_1, \ldots, x_n$ telle que $x = x_1$ et $y = x_n$ et $x_i = x_{i+1} \in E_1 \wedge E_2$. Aucune variable x_{i+1} ne peut être mise à 0 par l'application de l'hypothèse de récurrence (car ce sont des sous-termes de V). On a donc $E'_1 \wedge E'_2 \models x = y$. Ce cas n'est donc pas possible.

2. $v = t$ et $w = y$ où t est un terme clôs et y une variable. Comme $E'_1 \wedge E'_2 \not\models t = y$, y est égal modulo $E'_1 \wedge E'_2$ à un terme clôs où des sous-termes ont été remplacés par 0 par la fonction ρ, suite à l'application de l'hypothèse de récurrence. Cela signifie alors que $T \vdash t\ [\![]\!]$ n'est pas $F(V, T, \Sigma, s_f, \xi), E'_f$-admissible et il en est donc de même pour $T \vdash \{u\}_t\ [\![E'_1]\!]$. Ce séquent est donc issu d'une composition (il est $F(V, T, \Sigma, \{u\}_t, \xi), E'_f$-admissible).

   On note π le chapeau de preuve maximal ne contenant que des règles de $\mathcal{S}$ et des règles d'affaiblissement et donc la conclusion est le séquent $T \vdash \{u\}_t\ [\![E'_1]\!]$. Par maximalité de π et selon l'hypothèse de récurrence, les séquents utilisés comme hypothèses dans π sont $F(V, T, \Sigma, s_f, \xi), E'_f$-admissibles et ne contiennent donc pas le terme t. Celui-ci se trouve don construit dans π à l'aide de règles de chiffrement et de construction de paires. Il y a donc un séquent $T \vdash t\ [\![E'']\!]$ dans π. Nous le substituons par le séquent $T \vdash 0\ [\![]\!]$ ce qui permet d'obtenir une preuve de $T \vdash \{u\}_0\ [\![E''_1]\!]$ avec $E''_1 \wedge E'' = E'_1$. Nous appliquons alors le lemme 10.1 page 83 à la preuve de $T \vdash t\ [\![E'']\!]$ afin d'affaiblir la preuve de $T \vdash \{u\}_0\ [\![E''_1]\!]$ en $T \vdash \{u\}_0\ [\![E'_1]\!]$.

   Après cette transformation, l'égalité nécessaire à l'application de la règle de décomposition est retrouvée.

3. $v = y$ et $w = t$ où t est un terme clôs et y une variable. C'est le cas symétrique au précédent. Il se traite de manière identique, sauf qu'il n'est pas utile de chercher le terme t dans un chapeau de preuve car il se trouve directement en conclusion de la preuve de $T \vdash t\ [\![E'_2]\!]$.

4. $v = t$ et $w = y$ où t est un terme non clôs et y une variable. Si $\rho(t) = t$, il n'y a rien à faire car l'égalité $y = t$ est aussi vérifiée dans $E'_1 \wedge E'_2$. Supposons alors que $\rho(t) = 0$. Dans ce cas, par définition de la $F(V, T, \Sigma, \xi), E'_f$-admissibilité, tout terme égal modulo E'_f à t est également non $F(V, T, \Sigma, \xi), E'_f$-admissible. Comme $E'_f \models E'_1 \wedge E'_2$, l'hypothèse de récurrence nous indique que $E'_1 \wedge E'_2 \models y = 0$. Il nous suffit donc, comme dans le cas 2, de remplacer t par 0 dans la preuve pour s'assurer de l'égalité voulue.

5. $v = t$ et $w = y$ où t est un terme clôs et y une variable. Ce cas est identique au précédent.

6. Pour les autres cas, on procède en itérant sur la forme factorisée de $v = w$. Il y a trois cas possibles pour les égalités $v' = w'$ rencontrées :
   - Il s'agit d'égalités entre variables et on se ramène au premier point évoqué ci-dessus : $E'_1 \wedge E'_2 \models v' = w'$. Il n'y a donc pas à modifier la preuve pour ces égalités.
   - Il s'agit d'égalités entre une variable et un terme clôs. On se ramène au second ou troisième points évoqués ci-dessus qui transforme la preuve de façon à ce que l'égalité soit respectée.
   - Il s'agit d'une égalité entre une variable et un terme non variable et non clôs. On se ramène alors au quatrième et cinquième points évoqués ci-dessus.

   Au terme de l'itération, l'égalité entre v et w est assurée dans $E'_1 \wedge E'_2$.

Maintenant que nous avons $E'_1 \wedge E'_2 \models v = w$, nous pouvons appliquer la règle de déchiffrement pour obtenir une preuve de $T \vdash u\ [\![E'_1 \wedge E'_2]\!]$. Le premier point de l'hypothèse de récurrence est vérifiée de la même façon que pour une règle d'affaiblissement. La preuve du second point est semblable à celui de la règle d'affaiblissement ou à celle de composition mais nous ne la détaillerons pas ici afin de ne pas alourdir cette preuve dans le cas simple de Dolev-Yao.

Dans le cas général, nous procéderons de manière plus générale en traitant différemment les non-linéarités afin d'éviter la surmultiplication des cas.

Règle d'instanciation . La seule règle d'instanciation dans le cas de Dolev Yao est l'instanciation dont le contexte est vide. La preuve que nous considérons est donc de la forme :

$$\dfrac{\dfrac{\Pi_1}{T \vdash x \ \llbracket E \wedge x = u \rrbracket}}{T \vdash u \ \llbracket E \wedge x = u \rrbracket}\,\mathcal{I}$$

Nous appliquons l'hypothèse de récurrence sur la preuve Π_1 pour obtenir une preuve Π_χ. Deux cas peuvent alors se présenter. Ou bien la contrainte de Π_χ est de la forme $E' \wedge x = u$, ou bien elle est de la forme $E' \wedge x = 0$.

Dans le premier cas, il nous suffit d'appliquer la règle d'instanciation pour obtenir une preuve de $T \vdash u \ \llbracket E' \wedge x = u \rrbracket$. L'admissibilité des séquents de cette preuve provient alors directement de l'hypothèse de récurrence en ce qui concerne les séquents autres que le séquent final et du fait que le séquent $T \vdash u \ \llbracket E' \wedge x = u \rrbracket$ est la composition de la contrainte du séquent $T \vdash x \ \llbracket E' \wedge x = u \rrbracket$ et du terme u. Le séquent $T \vdash u \ \llbracket E' \wedge x = u \rrbracket$ est $F(V, T, \Sigma, \xi), E'_f$-admissible via l'hypothèse de récurrence et donc u est égal, modulo E'_f, à un terme qui est dans $F(V, T, \Sigma, \xi)$. $T \vdash u \ \llbracket E' \wedge x = u \rrbracket$ est donc également $F(V, T, \Sigma, \xi), E'_f$-admissible.

Dans le second cas, l'hypothèse de récurrence nous fournit une preuve du séquent $T \vdash u \ \llbracket E \wedge x = u \rrbracket$. On peut y appliquer l'hypothèse de récurrence pour obtenir une preuve de $T \vdash u \ \llbracket E' \wedge x = 0 \rrbracket$ vérifiant les propriétés voulues.

Règle de divulgation de nonce . On considère la preuve suivante :

$$\Pi = \dfrac{\dfrac{\Pi_1}{T \vdash u \ \llbracket E \rrbracket}}{T \vdash N_i(s) \ \llbracket E \rrbracket}\,\mathcal{ND}$$

On utilise l'hypothèse de récurrence sur la preuve Π_1 pour obtenir une preuve de $T \vdash u \ \llbracket E' \rrbracket$. Soit π_1 la sous-preuve maximale ne contenant que des règles d'affaiblissement et des règles de composition. On y applique le lemme 10.1 page 83 pour obtenir la preuve π_2. La règle de divulgation de nonce peut toujours être appliquée car les conditions d'application n'ont pas changé (le terme de la conclusion de π_2 n'a pas d'importance). On obtient alors la preuve Π_χ de $T \vdash N_i(s) \ \llbracket E' \rrbracket$.

Soit $T \vdash t \ \llbracket c \rrbracket$ un séquent de Π_χ. Si celui-ci est en dehors de π_2 et n'est pas le séquent final, l'hypothèse de récurrence permet de conclure. S'il est dans π_2, il est la composition du terme d'un des séquents $F(V, T, \Sigma, \xi), E'_f$-admissible utilisé en tant qu'hypothèse de π_2 et de la combinaison des contraintes de certains séquents également $F(V, T, \Sigma, \xi), E'_f$-admissible et hypothèses de π_2. Le séquent qui nous intéresse est donc aussi $F(V, T, \Sigma, \xi), E'_f$-admissible.

Enfin, si on considère le séquent conclusion de Π_χ, il s'agit de la combinaison de la contrainte E' du séquent $T \vdash u'$ $[\![E']\!]$ qui est $F(V,T,\Sigma,\xi), E'_f$-admissible comme on vient de le voir et du terme $N_i(s)$ qui appartient à $F(V,T,\Sigma,\xi)$. Dans ce cas, il est $F(V,T,\Sigma,\xi), E'_f$-admissible.

10.4 Hypothèses supplémentaires

Afin de prouver le théorème dans le cas général, nous allons introduire des hypothèses supplémentaires sur le système $\overline{\mathcal{S}}$. Contrairement aux hypothèses du chapitre 9 page 71, celles-ci ne sont pas motivées par la décidabilité. Il s'agit d'hypothèses afin de simplifier la preuve. Il est possible que celles-ci ne soient pas nécessaires. Elles sont utilisées dans le cas où l'on traite une règle de décomposition où au moins une des prémisses est à profondeur 2.

1. Pour toute règle de décomposition, pour tout couple de variables liées x et y (c'est-à-dire tels que $x = y$ est une condition nécessaire d'application de la règle de décomposition), si x est à profondeur 1 et y à profondeur 2, alors :
   - le symbole parent f de x est unaire,
   - le symbole parent g de y est unaire,
   - pour tout terme u, $f(u) \in F(T) \implies g(u) \in F(T)$.
2. Pour toute règle de décomposition, pour tout couple de variables liées x et y, si x et y sont à profondeur 2 alors :
   - le symbole parent f de x est unaire,
   - le symbole parent g de y est unaire,
   - pour tout terme u, $f(u) \in F(T) \Leftrightarrow g(u) \in F(T)$.
3. Pour les règles de décomposition où l'une des prémisses est $C[f(u_1,\ldots,u_m)]$ et la conclusion est $t = C[u_i]$ (cf section 9.3.2 page 76), f ne peut apparaître à profondeur 1 que dans cette règle de décomposition.

10.5 Pertinence des hypothèses restrictives

Par rapport au modèle général décrit dans le chapitre 4 page 33, nous avons pris un certain nombre d'hypothèses supplémentaires destinées à aider à démontrer le théorème principal :

- la propriété de localité 9.2 page 74,
- la forme des règles de composition dans la section 9.3.1 page 76,
- la forme des règles de décomposition dans la section 9.3.2 page 76,
- la clôture de la fonction F dans la section 10.4 et
- les symboles admis à profondeur 1 pour les règles de décomposition (dernier point de la section 10.4).

Les trois premières restrictions ont été introduites dans le chapitre 9 page 71 dédié à la décidabilité. Elles se justifient donc en partie par leur participation à la décidabilité du problème de l'existence d'une attaque. Les deux dernières hypothèses seront exploitées par la suite dans la preuve du théorème. Le lecteur peut cependant, légitimement, se demander si les hypothèses en question ne sont pas trop fortes.

Dans [Tur03], Mathieu Turuani définit des propriétés que doit respecter un intrus afin de permettre de décider de l'insécurité d'un protocole en temps NP. Ces propriétés sont définies ainsi :

Définition 10.2 (Règles d'oracle) *Soit un intrus normalisé $\mathcal{L}$ disposant des règles d'intrus normalisé $L_c \cup L_d \cup L_{oc} \cup L_{od}$ avec $L_c \cup L_d$, L_{oc} et L_{od} disjoints, L_{oc} des règles d'intrus normalisé de composition et L_{od} des règles d'intrus normalisé de décomposition. Alors $\mathcal{L}$ est un oracle (ou dispose des règles d'oracle) si et seulement si :*

1. *Pour tous E et t, si $t \in forge_{\mathcal{L}}(E)$, alors il existe une dérivation bien formée (construite sur $\mathcal{L}$) partant de E et de but t.*
2. *Pour tous E, a et t, si $E \rightarrow_{L_{oc}} E, t$ et $E, t \rightarrow_{L_d(t)} E, t, a$, alors il existe une dérivation D partant de E et de but a telle que $L_d(t) \notin D$.*
3. *Il existe une fonction ϵ : $Terme \setminus Atome \rightarrow \lceil Terme \rceil$ telle que pour tout message non atomique u, $|\epsilon(u)|_{dag} < |\lceil u \rceil|_{dag}$, et pour tout ensemble fini de message F et un message t, si $F \rightarrow_{L_c \cup L_{oc}} F, u$ (i.e u peut être composé en un pas) et si $F, u \rightarrow_{L_{oc} \cup L_{od}} F, u, t$ (i.e t n'est pas créé par DY), alors $\lceil t[u \leftarrow \epsilon(u)] \rceil \in Forge_{\mathcal{L}}(\lceil F[u \leftarrow \epsilon(u)] \rceil)$ et $\epsilon(u) \in Forge_{\mathcal{L}}(F)$.*

L_c constitue l'ensemble des règles de composition de Dolev-Yao. L_d est l'ensemble des règles de décomposition. $L_d(t)$ désigne l'ensemble des règles de décomposition pouvant s'appliquer sur le terme t. L'ensemble $\mathrm{Forge}_{\mathcal{L}}(E)$ est l'ensemble des termes que l'intrus peut déduire en utilisant les règles $\mathcal{L}$ à partir de l'ensemble de termes E. $E_1 \rightarrow_L E_2$ signifie que l'intrus peut déduire l'ensemble E_2 à partir de l'ensemble E_1 en appliquant une règle de L (en un pas). $\lceil u \rceil$ est la forme normale du terme u : la définition prend en compte les théories équationnelles. Pour de plus amples informations sur les notations, on se référera à [Tur03] d'où provient cette définition.

Nous disposons donc d'un point de comparaison pour juger de la pertinence des restrictions que nous avons imposées sur le modèle du chapitre 4 page 33.

Notons d'abord qu'un intrus disposant de règles d'oracle peut exploiter une théorie telle que le « ou exclusif », ce que nous ne pouvons pas faire actuellement. Considérons Dolev-Yao avec cette règle de composition supplémentaire :

$$\frac{a \quad b}{\langle \langle a, b \rangle, a \rangle}$$

Cette règle est un raccourci pour la construction suivante :

$$\frac{\dfrac{a \quad b}{\langle a, b \rangle} \quad a}{\langle \langle a, b \rangle, a \rangle}$$

Un tel intrus dispose des règles d'oracle mais nos restrictions sur les règles de composition sont trop importantes (ajout de symbole en tête uniquement) pour accepter de telles règles. La définition d'oracle semble donc plus générale de ce côté, même sans prendre en compte des théories équationnelles.

Inversement, voici un exemple qui respecte nos conditions mais qui ne constitue pas des règles d'oracle. Nous ajoutons à Dolev-Yao la règle suivante :

$$\frac{T \vdash g(x) \quad T \vdash f(g(x))}{T \vdash f(x)} \mathcal{D}$$

Nous ajoutons aussi une règle permettant de construire $g(x)$ à partir de x (mais pas de règle pour construire $f(x)$ à partir de x). Si l'on considère la fonction F comme étant la plus petite fonction incluant les sous-termes et close par la relation $f(g(x)) \in F(T) \implies f(x) \in F(T)$. Cette fonction permet à notre ensemble de règles de vérifier la propriété de localité 9.2. Par contre, le premier point de la définition des règles d'oracle n'est pas respecté : une dérivation bien formée n'exploite que des sous-termes. Les autres restrictions imposées sur notre modèle sont vérifiées.

Considérons la preuve suivante :

$$\dfrac{\dfrac{T \vdash x}{T \vdash g(x)}\,\mathcal{C} \qquad T \vdash f(g(x))}{T \vdash f(x)}\,\mathcal{D}$$

Nous prenons $T = \{x, f(g(x))\}$. Avec les notations de la définition 10.2, on choisit $u = g(x)$ et $t = f(x)$. Nous sommes dans les conditions du troisième point de la définition et nous supposons l'existence d'une fonction ϵ ayant les propriétés voulues. On note θ la substitution $u \leftarrow \epsilon(u)$. $t\theta = t$ et $F\theta = \{x, f(\epsilon(g(x)))\}$. Il n'est cependant pas possible d'avoir $t\theta \in \mathrm{Forge}_{\mathcal{L}}(F\theta)$ en respectant la condition de taille sur u et $\epsilon(u)$. La règle ci-dessus ne permet donc pas non plus de vérifier le troisième point des règles d'oracle.

Les deux modèles sont donc incomparables. Les règles d'oracle ont l'avantage de permettre l'utilisation des théories équationnelles. En contre-partie, trouver la fonction ϵ qui convient est plus complexe que la vérification de nos propriétés purement syntaxiques.

Tentons tout de même de comparer informellement les deux approches. Le premier point des règles d'oracle correspond à notre propriété de localité. Toutefois, nous pouvons utiliser une fonction F plus générale que les sous-termes. Le second point est une seconde propriété de localité que nous avons traduit (de manière plus drastique) en une limitation sur la forme des règles de composition.

Le troisième point correspondrait aux restrictions restantes : pour chaque règle de décomposition telle que l'une des prémisses soit $t_j = C[f(u_1, \ldots, u_m)]$ et la conclusion $t = C[u_i]$, où C est un contexte et f un symbole de construction, on définit $\epsilon(f(u_1, \ldots, u_m)) = u_i$. Sur les autres termes, on impose $\epsilon(u) = 0$. Une telle fonction ϵ est correctement définie en raison de la dernière restriction de la section 10.4 page 94 qui interdit au symbole f d'apparaître dans des conditions similaires dans d'autres règles de décomposition. Nous respectons de plus la condition sur la taille des termes. Il ne nous semble cependant pas possible de prouver que les autres conditions sont strictement respectées, ni même qu'une forme proche puisse être démontrée. On appréciera toutefois leurs proximités avec les résultats obtenus dans la preuve du théorème principal quand on traite les règles de décomposition. C'est d'ailleurs à cette occasion que les conditions de clôture de la fonction F sont utilisées.

10.6 Preuve

On applique en premier lieu le lemme 8.2 page 66.

On utilise ensuite le lemme 8.3 page 68 : les instanciations sont repoussées le plus possible vers la racine. Regardons la preuve Π_0 de $T \vdash s' \; [\![E]\!]$ telles que les instanciations repoussées permettent d'obtenir une preuve de $T \vdash s \; [\![E]\!]$. On peut alors prouver le théorème pour Π_0 à condition que la contrainte que l'on obtiendra E' ait le même domaine que la contrainte

E pour les variables qui sont utilisées lors des instanciations finales. Dans ce cas, on aura effectivement $s\sigma_{E'} = s'\sigma_{E'}$.

Par la suite, on travaille directement sur Π_0 et on considère donc sans perte de généralité qu'il s'agit d'une preuve de $T \vdash s \ \llbracket E \rrbracket$ ne se terminant pas par une instanciation.

Nous allons prouver le théorème en procédant par récurrence sur la taille de la preuve Π.

10.6.1 Hypothèse de récurrence

Nous allons énoncer ici l'hypothèse de récurrence utilisée pour prouver le théorème. Celle-ci est singulièrement plus compliquée que le théorème et nous allons donc également justifier brièvement les hypothèses mises en jeu. Avant d'entamer la preuve de cette hypothèse, nous prouverons également sa pertinence, c'est-à-dire le fait qu'elle prouve bien le théorème principal.

10.6.1.1 Énoncé de l'hypothèse de récurrence

La récurrence est paramétrée par l'ensemble de contraintes E_f sous forme factorisée. Il s'agira en fait de la contrainte finale E de la preuve Π du théorème. Elle est également paramétrée par le terme s_f qui correspondra à la conclusion de la preuve Π du théorème.

Soit Π une preuve de $T \vdash s \ \llbracket E \rrbracket$ ne se terminant pas par une instanciation dont le contexte n'est pas vide et telle que $E_f \models E$. On définit les éléments suivants (indépendamment de l'hypothèse de récurrence) :

1. Un ensemble de variables χ distinctes de tout terme existant ; chaque variable est associée à un terme $t \notin F(V, T, \Sigma, s_f, \xi)$ et à une variable $x \in V$ ou à un entier i. Elle est notée χ_t^x ou χ_t^i selon les cas.
2. Une fonction ρ dont le domaine est le produit de l'ensemble des variables de V et de l'ensemble des termes et l'image $F(V, T, \Sigma, s_f, \xi) \cup \chi$ telle que pour tout terme t tel que $t =_{E_f} u$ et $u \in F(V, T, \Sigma, s_f, \xi)$ mais u n'est pas une variable, $\rho(x, t) = t$; sinon, $\rho(x, t) = \chi_t^x$.
3. Une substitution A_χ telle que pour tout terme $t \notin F(V, T, \Sigma, s_f, \xi)$, tout variable $x \in V$ et tout entier i, $\chi_t^i A_\chi = \chi_t^x A_\chi = t$.

Il existe alors un ensemble de contraintes F_χ sur les variables de χ tel que chaque contrainte est de l'une des deux formes suivantes, avec $t, t_1, \ldots, t_n$ des termes quelconques, x une variable de V apparaissant dans E et $i, i_1, \ldots, i_n$ des entiers :

1. $\chi_t^i = \chi_t^j$ avec $i < j$,
2. $\chi_t^i = \chi_t^x$,
3. $\chi_t^x = \chi_t^y$ avec $x < y$ pour l'ordre lexicographique sur les noms des symboles,
4. $\chi_{f(t_1,\ldots,t_n)}^x = f(\chi_{t_1}^{i_1}, \ldots, \chi_{t_n}^{i_n})$ où f est un symbole de construction.

On a automatiquement les deux propriétés suivantes :

- $A_\chi \models F_\chi$ et
- $E \models \bigwedge_{x=t \in E} x = \rho(x, t) \wedge A_\chi$.

Il existe un arbre de preuve Π_χ de $T, \chi \vdash s \ \llbracket E' \rrbracket$ avec $E' = \bigwedge_{x=t \in E} x = \rho(x, t)$ et tel que pour toute substitution $\theta_\chi \models F_\chi$ dont le domaine est l'ensemble des variables χ, $\Pi_{\theta_\chi} = \Pi_\chi \theta_\chi$ est une preuve de $T, \chi\theta_\chi \vdash s \ \llbracket E'\theta_\chi \rrbracket$ dans $\overline{\mathcal{S}}$ telle que, en notant $E'_f = \bigwedge_{x=t \in E_f} x = \rho(x, t)\theta_\chi$:

1. pour toute variable χ_u^x apparaissant en membre droit de E', il existe une preuve terminant par une construction de $T \vdash u \ \llbracket E \rrbracket$ plus petite que Π et dans $\overline{\mathcal{S}}$.
2. pour toute contrainte de F_χ de la forme $\chi^x_{f(t_1,\ldots,t_n)} = f(\chi^{i_1}_{t_1}, \ldots, \chi^{i_n}_{t_n})$:
   - ou bien $\chi^{i_1}_{t_1}, \ldots, \chi^{i_n}_{t_n}$ sont minimales pour l'ordre d'occurrence dans F_χ,
   - ou bien ces variables ne le sont pas et sont égales, modulo F_χ à $\chi^{z_1}_{t_1}, \ldots, \chi^{z_n}_{t_n}$ et $f(z_1, \ldots, z_n) \in F(V, T, \Sigma, s_f, \xi)$.
3. Pour toute sous-preuve π de Π_{θ_χ} dont la conclusion est $T, \chi\theta_\chi \vdash v \ \llbracket c\theta_\chi \rrbracket$, tout séquent $T, \chi\theta_\chi \vdash u \ \llbracket c'\theta_\chi \rrbracket$ de π est $F(V, T, \Sigma, v, \xi, \chi\theta_\chi), E'_f$-admissible si la dernière règle de π la plus à gauche qui n'est pas une règle d'affaiblissement est une règle de composition ou $F(V, T, \Sigma, \xi, \chi\theta_\chi), E'_f$-admissible dans le cas contraire.

10.6.1.2 Justification

L'application du théorème de normalisation va permettre d'obtenir une preuve Π' dont le résultat est sensiblement équivalent à celui de la preuve Π mais dont des termes ont été remplacés par des termes plus petits.

La principale nouveauté de l'hypothèse de récurrence par rapport à l'énoncé du théorème est l'introduction des variables χ. Celles-ci ont pour but de modéliser l'incertitude sur la forme correcte du terme plus petit destiné à remplacer le terme trop gros. En effet, au cours de la récurrence, on ne connaît pas encore la nature exacte du terme qui pourra remplacer un terme trop gros : peut-on remplacer celui-ci directement par 0 ? Par la suite, il sera peut-être nécessaire d'avoir accès au symbole de tête de ce terme pour pouvoir appliquer une décomposition. Il est même possible qu'il soit nécessaire d'accéder au symbole de tête de l'un des sous-termes.

Voici un exemple de preuve où le choix du petit terme n'est pas immédiat et qui justifie donc l'introduction des variables χ. Dans cet exemple, on dispose des règles de $\mathcal{S}$ suivantes :

$$\frac{T \vdash t \ \llbracket E \rrbracket}{T \vdash p_1(t) \ \llbracket E \rrbracket}\,\mathcal{S} \qquad \frac{T \vdash t \ \llbracket E \rrbracket}{T \vdash p_2(t) \ \llbracket E \rrbracket}\,\mathcal{S}$$

$$\frac{T \vdash t \ \llbracket E \rrbracket}{T \vdash g(t) \ \llbracket E \rrbracket}\,\mathcal{S} \qquad \frac{T \vdash \{t_1\}_{p_1(t_2)} \ \llbracket E_1 \rrbracket \quad T \vdash p_2(t_2) \ \llbracket E_2 \rrbracket}{T \vdash t_1 \ \llbracket E_1 \wedge E_2 \rrbracket}\,\mathcal{S}$$

$$\frac{T \vdash h(t_1, g(t_2)) \ \llbracket E \rrbracket}{T \vdash t_1 \ \llbracket E \rrbracket}\,\mathcal{S}$$

On pose $T = \{t\}$. Le but est d'obtenir le terme $\langle s1, s2 \rangle$. On peut s'aider des règles de protocole suivantes (il n'y a qu'un seul rôle) :

1. $y \to \{s_1\}_y$
2. $z \to p_2(z)$
3. $z \to h(s_2, z)$

La preuve de $\langle s1, s2 \rangle$ est donnée de manière schématique dans la figure 10.8 page ci-contre. Les séquents y sont écrits de manière incomplète pour des raisons de simplicité. Il manque de plus l'utilisation des règles d'affaiblissement nécessaires à l'application des règles de protocole 2 et 3. On note toutefois qu'il est possible d'établir un ordre sur la preuve : d'abord la branche de gauche, puis la branche de droite et enfin la preuve de s_2.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{t}{g(t)}}{p_1(g(t))}}{\{s_1\}_y \ \ \llbracket y = p_1(g(t)) \rrbracket}\, 1}{\{s_1\}_{p_1(g(t))}}\, \mathcal{I} \qquad \dfrac{\dfrac{\dfrac{t}{g(t)}}{p_2(z) \ \ \llbracket z = g(t) \rrbracket}\, 2}{p_2(g(t))}\, \mathcal{I}}{s_1}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{t}{g(t)}}{h(s_2, z) \ \ \llbracket z = g(t) \rrbracket}\, 3}{h(s_2, g(t))}\, \mathcal{I}}{s_2}$$

FIG. 10.8 - *Preuve de* $\langle s1, s2 \rangle$

Le terme $g(t)$ est trop gros. Le terme $p_2(g(t))$ est donc également trop gros. Imaginons que l'on procède par récurrence. Si on considère la preuve de s_1, on serait tenté de remplacer $g(t)$ par 0. Toutefois, dans la preuve de s_2, le symbole de tête de g est nécessaire pour appliquer l'une des règles de $\mathcal{S}$. Il aurait donc fallu remplacer $g(t)$ par $g(0)$.

Étant donné que pendant la preuve de s_1, il ne nous est pas possible de savoir que le symbole de tête de $g(t)$ est important pour la suite, on introduit les variables χ. $g(t)$ est dans un premier temps remplacé par $\chi^z_{g(t)}$, ce qui signifie que la preuve peut fonctionner en remplaçant $g(t)$ par n'importe quel terme. Par la suite, on remplacera $p_1(g(t))$ par $\chi^y_{p_1(g(t))}$ puis on ajoutera la contrainte $\chi^y_{p_1(g(t))} = p_1(\chi^1_{g(t)})$ à F_χ car le fait que le symbole de tête du terme que remplace $\chi^y_{p_1(g(t))}$ nous importe pour l'application de la règle de décomposition.

Intuitivement, à la fin de la récurrence, une fois F_χ entièrement construit, on remplacera les variables de χ qui sont minimales pour l'ordre d'occurrence dans F_χ par 0 pour se ramener au théorème principal. Nous verrons par la suite que les hypothèses sur la contrainte F_χ permettent d'assurer l'admissibilité de tous les séquents du moment que θ_χ est choisit relativement minimal (ce qui est le cas si l'on procède comme indiqué ci-dessus).

Enfin, la restriction aux preuves ne se terminant pas par des instanciations nous est nécessaire car au moment de l'instanciation, nous n'avons pas les informations nécessaires pour effectuer la substitution par un terme plus petit : celle-ci dépend grandement de la règle de décomposition qui suit. Par exemple, cette règle a-t-elle besoin du symbole de tête ?

10.6.1.3 Pertinence

L'hypothèse de récurrence étant beaucoup plus complexe que l'énoncé du théorème, il faut s'assurer en premier lieu qu'elle implique le théorème.

Rappelons d'abord que la preuve Π de $T \vdash s \ \llbracket E \rrbracket$ que l'on tente de normaliser est le résultat de l'application des lemmes 8.2 et 8.3 et que celle-ci ne se termine pas par une instanciation. On recherche alors une preuve Π' de $T \vdash s\ \llbracket E' \rrbracket$ dans $\overline{\mathcal{S}}$ telle que $E \models\!\!\dashv E'$. En effet, le fait d'avoir retardé les instanciations permet de conserver le même terme s entre la preuve Π et la preuve Π'. On choisit $E_f = E$ et $s_f = s$ pour les deux paramètres de la récurrence. On se trouve alors dans les conditions d'application de l'hypothèse de récurrence.

On dispose donc d'un ensemble de variables χ et d'un ensemble de contraintes F_χ sur celles-ci. Il est possible d'ordonner les variables de χ selon leurs dépendances dans F_χ : les variables minimales pour l'ordre d'occurrence dans F_χ sont indicées par 0. Si $\chi^x_{f(u_1,\ldots,u_k)}$ est membre gauche dans F_χ, son indice est $n+1$ où n est le plus grand indice parmi les indices de $\chi^{i_1}_{u_1},\ldots,\chi^{i_k}_{u_k}$. Enfin si $\chi^i_u = \chi^x_u$ est dans F_χ, alors l'indice de $\chi^{i_k}_{u_k}$ est celui de χ^x_u. Il en est de même pour les contraintes $\chi^i_u = \chi^j_u$ et $\chi^x_u = \chi^y_u$. Nous choisissons θ_χ tel que les variables qui sont minimales pour l'ordre d'occurrence dans F_χ sont affectées à 0 et tel que $\theta_\chi \models F_\chi$. Il existe donc une preuve Π_{θ_χ} de $T, \chi\theta_\chi \vdash s\ \llbracket E'\theta_\chi \rrbracket$ dans $\overline{\mathcal{S}}$. On note que $E'\theta_\chi = E'_f$.

Nous devons alors résoudre deux problèmes sur cette preuve :

① les éléments de $\chi\theta_\chi$ doivent être constructibles depuis T, pour obtenir une preuve de $T \vdash s\ \llbracket E'\theta_\chi \rrbracket$ en éliminant la coupure,

② un séquent $F(V,T,\Sigma,s,\xi,\chi\theta_\chi), E'_f$-admissible doit être également $F(V,T,\Sigma,s,\xi), E'_f$-admissible.

Une fois ces deux conditions réunies, le théorème est vérifié à partir du moment où l'hypothèse de récurrence est vérifiée. En effet, tout séquent de la preuve sera $F(V,T,\Sigma,s,\chi), E'_f$-admissible grâce au second point. Le premier point fourni une preuve de $T \vdash s\ \llbracket E'\theta_\chi \rrbracket$ et comme le domaine de $\sigma_{E'\theta_\chi}$ est le même que celui de σ_E, la première partie du théorème est vérifiée.

En ce qui concerne le point ①, nous allons montrer par récurrence sur l'ordre d'occurrence que chacun des termes dans $\chi\theta_\chi$ est constructible. Pour le cas de base, les variables minimales pour l'ordre d'occurrence sont substituées par 0 qui est dans T. Pour le cas récursif, on considère la variable $\chi^x_{f(u_1,\ldots,u_k)}$. Si elle est égale à $\chi^y_{f(u_1,\ldots,u_k)}$, on applique l'hypothèse de récurrence. Sinon, $\chi^{i_1}_{u_1},\ldots,\chi^{i_k}_{u_k}$ sont constructibles depuis T. Le symbole f est un symbole de construction. Il est donc possible de construire $\chi_{f(u_1,\ldots,u_k)}\theta_\chi$ depuis T. La construction de la variable χ^i_u revient à la construction de la variable χ^x_u ou χ^j_u qui lui correspond dans F_χ.

Regardons le point ②. Considérons $T \vdash r\ \llbracket G \rrbracket$ un séquent contraint issu de la preuve Π_{θ_χ} et $F(V,T,\Sigma,s,\xi,\chi\theta_\chi), E'_f$-admissible. Pour que celui-ci soit également $F(V,T,\Sigma,s,\xi), E'_f$-admissible, nous devons montrer deux choses :

❶ pour tout $x = \rho(x,t)\theta_\chi \in G$ (on rappelle que E' est $\bigwedge_{x=t\in E} x = \rho(x,t)$), $\rho(x,t)\theta_\chi$ est dans $F(V,T,\Sigma,s,\xi)$ ou bien il existe un terme t tel que $t =_{E'_f} \rho(x,t)\theta_\chi$ et $t \in F(V,T,\Sigma,s,\xi) \setminus V$;

❷ $r \in F(V,T,\Sigma,s,\xi)$ ou alors il existe $t =_{E'_f} r$ tel que $t \in F(V,T,\Sigma,s,\xi)$.

Pour le point ❷, on sait que $r \in F(V,T,\Sigma,s,\xi,\chi\theta_\chi)$, mais l'hypothèse de récurrence nous garantit que, en dehors des cas où la règle d'instanciations est appliquée, r ne contient pas d'instances de $\chi\theta_\chi$ et donc $r \in F(V,T,\Sigma,s,\xi)$, ou bien il existe $t =_E r$ tel que $t \in F(V,T,\Sigma,s,\xi,\chi\theta_\chi)$ et on peut faire la même remarque que pour r, sachant que E ne contient pas non plus de références aux variables χ et donc $t \in F(V,T,\Sigma,s,\xi)$.

Regardons le point ❶. Supposons que $\rho(x,t)$ est une variable de χ. Si celle-ci est minimale pour l'ordre d'occurrence dans F_χ, $\rho(x,t)\theta_\chi = 0$ et $0 \in T$ permettent de conclure. Sinon, la variable en question est de la forme $\chi^x_{f(u_1,\ldots,u_k)}$. Si les variables $\chi^{i_1}_{u_1},\ldots,\chi^{i_k}_{u_k}$ qui y sont associées dans F_χ sont minimales pour l'ordre d'occurrence dans F_χ, alors $\rho(x,t)\theta_\chi = f(0,\ldots,0)$

qui est aussi par hypothèse dans T car f est un symbole de construction. Sinon, on sait que $\rho(x,t)\theta_\chi =_{E'\theta_\chi} f(z_1,\ldots,z_k) \in F(V,T,\Sigma,s,\xi)$ où $z_1,\ldots,z_k$ sont des variables de V. Or $E'\theta_\chi = E'_f$, on a donc l'égalité voulue.

Supposons maintenant que $\rho(x,t)$ n'est pas une variable de χ. Cela signifie qu'il existe un terme u tel que $t =_E u$ et $u \in F(V,T,\Sigma,s,\xi)$. Ce terme ne peut pas présenter de variables de χ et est donc insensible à l'application de θ_χ. Donc, $t = \rho(x,t)\theta_\chi =_{E'_f} \rho(x,u)\theta_\chi = u$.

Ainsi, prouver l'hypothèse de récurrence permet de prouver le théorème.

10.6.2 Preuve

Nous procédons donc par récurrence sur la taille de la preuve Π. Les deux paramètres de la récurrence sont E_f et s_f. Si Π est réduite à l'application de la règle d'axiome, on choisit l'ensemble F_χ vide. Pour toute substitution $\theta_\chi \models F_\chi$, nous associons la preuve Π_{θ_χ} identique à Π. Toutes les conditions requises sont vérifiées.

Intéressons-nous désormais à la dernière règle de Π. Pour des raisons de concision, nous allons manipuler par la suite directement les preuves « Π_{θ_χ} » plutôt que des preuves « Π_χ ». On notera toutefois que la preuve Π_χ correspondant peut être obtenue en retirant syntaxiquement les occurrences de θ_χ et donc que cette preuve ne dépend pas de θ_χ.

Règle d'affaiblissement . La preuve Π est obtenue par affaiblissement de deux preuves Π_1 et Π_2 :

$$\dfrac{\dfrac{\Pi_1}{T \vdash u_1 \ \llbracket E_1 \rrbracket} \quad \dfrac{\Pi_2}{T \vdash u_2 \ \llbracket E_2 \rrbracket}}{T \vdash u_1 \ \llbracket E_1 \wedge E_2 \rrbracket}\ \mathcal{W}$$

On applique l'hypothèse de récurrence sur Π_1 et Π_2 pour obtenir deux ensembles de contraintes F^1_χ et F^2_χ de χ. Ni Π_1, ni Π_2 ne peut terminer par une règle d'instanciation car une règle d'affaiblissement ne peut pas « retenir » une instanciation lors de l'application des règles de la figure 8.1 du lemme 8.3 page 68.

On note F_χ l'union de F^1_χ et de F^2_χ. On effectue les renommages nécessaires des χ^i_u dans l'une des deux sous-preuves pour ne pas avoir le même χ^i_u comme membre gauche de F^1_χ et de F^2_χ. Le cas des variables du type χ^x_u ne pose pas de problème. F_χ est alors de la forme attendue.

L'hypothèse de récurrence fournit également deux arbres de preuve Π^1_χ et Π^2_χ de, respectivement, $T,\chi \vdash u_1 \ \llbracket E'_1 \rrbracket$ et $T \vdash u_2 \ \llbracket E'_2 \rrbracket$. On note $E' = E'_1 \wedge E'_2$.

Soit χ^x_u apparaissant en tant que membre droit de E'. Si χ^x_u était membre droit dans E'_1, alors on utilise l'hypothèse de récurrence pour conclure qu'il existe une preuve de $T \vdash u \ \llbracket E_1 \rrbracket$ plus petite que Π dans $\overline{\mathcal{S}}$. Sinon, χ^x_u est membre droit dans E'_2 et on applique le même raisonnement pour obtenir une preuve de $T \vdash u \ \llbracket E_2 \rrbracket$. Dans les deux cas, ces preuves se terminent par des constructions. On affaiblit l'une des hypothèses avec la preuve de $T \vdash u_2 \ \llbracket E_2 \rrbracket$ (respectivement $T \vdash u_1 \ \llbracket E_1 \rrbracket$) puis on applique la règle de construction pour obtenir une preuve de $T \vdash u \ \llbracket E_1 \wedge E_2 \rrbracket$ se terminant par une règle de construction. Le premier point de l'hypothèse de récurrence est donc vérifié.

Passons aux deux points suivants. Soit une substitution $\theta_\chi \models F_\chi$. On a donc $\theta_\chi \models F^1_\chi$ et $\theta_\chi \models F^2_\chi$. On dispose donc de preuves $\Pi^1_{\theta_\chi}$ et $\Pi^2_{\theta_\chi}$ de $T,\chi\theta_\chi \vdash u_1 \ \llbracket E'_1\theta_\chi \rrbracket$ et de $T,\chi\theta_\chi \vdash u_2 \ \llbracket E'_2\theta_\chi \rrbracket$.

Soit π le chapeau de preuve maximal de $T, \chi\theta_\chi \vdash u_2 \ [\![E_2'\theta_\chi]\!]$ ne contenant que des règles de composition de $\mathcal{S}$ et des règles d'affaiblissement. On y applique le lemme 10.2 pour obtenir π'. Sur la partie de π' ne contenant que des règles de $\mathcal{S}$, nous appliquons le lemme 10.1 page 83 pour obtenir la preuve π''. Nous pouvons alors obtenir une preuve de $T, \chi\theta_\chi \vdash u_3 \ [\![E_2'\theta_\chi]\!]$ où u_3 est un terme issu de l'application du lemme 10.1. On applique la règle d'affaiblissement sur les preuves $\Pi^1_{\theta_\chi}$ et $\Pi^3_{\theta_\chi}$:

$$\Pi_{\theta_\chi} = \cfrac{\cfrac{\Pi^1_{\theta_\chi}}{T, \chi\theta_\chi \vdash u_1 \ [\![E_1'\theta_\chi]\!]} \quad \cfrac{\Pi^3_{\theta_\chi}}{T, \chi\theta_\chi \vdash u_2 \ [\![E_2'\theta_\chi]\!]}}{T, \chi\theta_\chi \vdash u_1 \ [\![E_1'\theta_\chi \wedge E_2'\theta_\chi]\!]}\ \mathcal{W}$$

$E' = E_1' \wedge E_2'$ est bien égal à $\bigwedge_{x=t\in E_1\wedge E_2} x = \rho(x,t)$ car $E_1' = \bigwedge_{x=t\in E_1} x = \rho(x,t)$ et $E_2' = \bigwedge_{x=t\in E_2} x = \rho(x,t)$ d'après l'hypothèse de récurrence. Aucune nouvelle variable de χ n'a été introduite dans F_χ, il suffit donc d'appliquer l'hypothèse de récurrence en notant que $E_1' \wedge E_2' \models E'$. Le second point est donc vérifié.

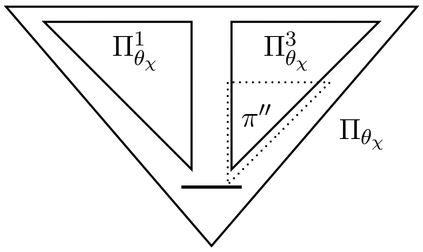

FIG. 10.9 - *Notations utilisées dans le cas d'une règle d'affaiblissement*

Considérons ensuite le dernier point. Les notations sont rappelées dans la figure 10.9. Soit P_1 une sous preuve de Π_{θ_χ} dont la conclusion est le séquent $T, \chi\theta_\chi \vdash v \ [\![c\theta_\chi]\!]$ et $T, \chi\theta_\chi \vdash u \ [\![c'\theta_\chi]\!]$ un séquent de P_1. Si P_1 est enraciné dans $\Pi^1_{\theta_\chi}$ ou si elle est enracinée dans l'une des hypothèses de π'' (qui est un chapeau de $\Pi^3_{\theta_\chi}$), on applique juste l'hypothèse de récurrence. Si P_1 est enracinée dans π'', par maximalité de π et suite à l'application du lemme 10.1, la règle la plus à gauche qui n'est pas une règle d'affaiblissement n'est pas non plus une règle de composition. Si le séquent $T, \chi\theta_\chi \vdash u \ [\![c'\theta_\chi]\!]$ n'est pas dans π'', il est $F(V, T, \Sigma, \xi, \chi\theta_\chi), E_f'$-admissible grâce à l'hypothèse de récurrence. S'il est dans π'', il est composé d'un terme qui est celui de l'hypothèse la plus à gauche de la sous-preuve π'' et d'une contrainte qui est la combinaison des contraintes de certaines hypothèses de π''. Ces deux éléments réunis montrent que le séquent est $F(V, T, \Sigma, \xi, \chi\theta_\chi), E_f'$-admissible par hypothèse de récurrence.

Enfin, si $P_1 = \Pi_{\theta_\chi}$, ou bien le séquent est dans $\Pi^3_{\theta_\chi}$ et vu ce qui a été dit ci-dessus, il est $F(V, T, \Sigma, \xi, \chi\theta_\chi), E_f'$-admissible. Ou bien, il est dans $\Pi^1_{\theta_\chi}$ et si la règle la plus à gauche qui n'est pas une règle d'affaiblissement est une règle de composition, c'est aussi le cas pour la preuve Π_{θ_χ} et vice-versa. Le séquent est donc $F(V, T, \Sigma, \xi, \chi\theta_\chi), E_f'$-admissible (dans le premier cas), ou bien $F(V, T, \Sigma, s, \xi, \chi\theta_\chi), E_f'$-admissible (dans le second cas) en utilisant l'hypothèse de récurrence. Si le séquent est la conclusion de Π_{θ_χ}, il est la combinaison du terme constituant la conclusion s de $\Pi^1_{\theta_\chi}$ et de la contrainte $E_1'\theta_\chi \wedge E_2'\theta_\chi$.

On sait déjà que le séquent $T, \chi\theta_\chi \vdash s \; [\![E'_1\theta_\chi]\!]$ est $F(V, T, \Sigma, s, \xi, \chi\theta_\chi), E'_f$-admissible ou $F(V, T, \Sigma, \xi, \chi\theta_\chi), E'_f$-admissible selon la nature de la règle la plus à gauche. Sachant que le séquent $T, \chi\theta_\chi \vdash u_2 \; [\![E'_2\theta_\chi]\!]$ est $F(V, T, \Sigma, \xi, \chi\theta_\chi), E'_f$-admissible, le séquent $T, \chi\theta_\chi \vdash s \; [\![E'_1\theta_\chi \wedge E'_2\theta_\chi]\!]$ est alors $F(V, T, \Sigma, s, \xi, \chi\theta_\chi), E'_f$-admissible ou alors $F(V, T, \Sigma, s, \xi, \chi\theta_\chi), E'_f$ selon la nature de la règle la plus à gauche. Dans tous les cas, le dernier point de l'hypothèse de récurrence est vérifié.

Règle de protocole . On considère la preuve suivante :

$$\Pi = \cfrac{\cfrac{\Pi_1}{T \vdash u \;\; [\![E]\!]}}{T \vdash w \;\; [\![u = v \wedge E \wedge x_{R,k,s} = 1 \wedge \sigma_S]\!]} \mathcal{P}$$

Dans le cas où la règle correspond à une progression de session, c'est-à-dire quand $k > 1$, σ_S est vide. La preuve Π_1 ne se termine pas par une instanciation étant donné que l'on a appliqué le lemme 8.3 page 68. Nous pouvons donc utiliser l'hypothèse de récurrence. Nous obtenons un ensemble de contraintes F_χ de χ.

$u = v$ est sous forme résolue dans la contrainte finale de Π. La forme résolue de $u = v \wedge E$ est $E \wedge x_1 = u_1 \wedge \ldots \wedge x_n = u_n$ où x_i est une variable de v et u_i un sous-terme de u.

Soit $\theta_\chi \models F_\chi$. L'hypothèse de récurrence nous fournit une preuve $\Pi^1_{\theta_\chi}$ du séquent $T, \chi\theta_\chi \vdash u \; [\![E'\theta_\chi]\!]$.

Soit u_i l'un des membres droits de la forme résolue de $u = v$, si $T, \chi\theta_\chi \vdash u_i \; [\![]\!]$ n'est pas $F(V, T, \Sigma, \xi, \theta_\chi), E'_f$ admissible, nous allons remplacer u_i par $\chi^x_{u_i}\theta_\chi$ dans u où x est le membre gauche de la contrainte ayant u_i comme membre droit. Pour ce faire, nous considérons le chapeau maximal π_1 de $\Pi^1_{\theta_\chi}$ ne contenant que des règles de composition et des règles d'affaiblissement. Nous appliquons le lemme 10.2 page 83 sur cette preuve pour obtenir la preuve π_2. Appelons π_3 la partie ne contenant que des règles de composition. Il existe un séquent de π_3 tel que u_i en soit le terme : en effet, les règles de composition consistent simplement en l'ajout d'un symbole en tête (cf section 9.3.1 page 76) et u_i est le résultat des compositions car les hypothèses de π_3 sont $F(V, T, \Sigma, \xi, \chi\theta_\chi), E'_f$-admissibles grâce à l'hypothèse de récurrence. La preuve de $T, \chi\theta_\chi \vdash u_i \; [\![E'']\!]$ peut être remplacée par la preuve de $T, \chi\theta_\chi \vdash \chi^x_{u_i}\theta_\chi \; [\![]\!]$. Toutefois, nous avons perdu l'ensemble des contraintes. E'' est la conjonction d'un certain nombre de contraintes de séquents que constituent les hypothèses de π_3. En appliquant la règle d'affaiblissement à la preuve de $T, \chi\theta_\chi \vdash \chi^x_{u_i}\theta_\chi \; [\![]\!]$ et à chacun de ces séquents, nous obtenons une preuve de $T, \chi\theta_\chi \vdash \chi^x_{u_i}\theta_\chi \; [\![E'']\!]$.

En itérant cette transformation sur chacune des positions contenant un u_i, nous obtenons alors une preuve de $T, \chi\theta_\chi \vdash u' \; [\![E'\theta_\chi]\!]$ avec $u' = u[\bigcup_{(u_i, x_i)} \{u_{ip(u_i, x_i)} \leftarrow \chi^{x_i}_{u_i}\theta_\chi\}]$ où $p(u_i, x_i)$ est position de u_i dans u correspondant à la variable x_i dans v.

L'application de la règle de protocole sur cette preuve est toujours possible : u' s'unifie toujours avec v car nous avons remplacé des termes qui étaient liés aux variables de v et ceci, de manière uniforme : si $v = f(x, x)$, les deux termes t liés aux deux variables x ont été remplacés tous les deux par χ^x_t.

Nous obtenons la preuve Π_{θ_χ} de $T, \chi\theta_\chi \vdash w \; [\![\bigwedge_i x_i = u'_i\theta_\chi \wedge E'\theta_\chi \wedge x_{R,k,s} = 1 \wedge \sigma_S]\!]$ où $u'_i = u_i$ si u_i n'a pas été remplacé par une variable de χ et $u'_i = \chi^{x_i}_{u_i}$ dans le cas contraire.

Nous devons vérifier le premier point de la récurrence. Pour les variables autres que $\chi^{x_i}_{u_i}$, il suffit d'appliquer l'hypothèse de récurrence pour conclure. Prenons un $\chi^{x_i}_{u_i}$. Nous avons vu ci-dessus que nous avions une preuve de $\chi^x_{u_i}\theta_\chi$ $[\![E'']\!]$ que nous pouvions transformer par affaiblissement en une preuve de $T \vdash \chi^x_{u_i}\theta_\chi$ $[\![E'\theta_\chi]\!]$. Le squelette de cette preuve étant indépendant de θ_χ, on peut la transformer en une preuve de $T \vdash u_i$ $[\![E]\!]$ en posant $\theta_\chi = A_\chi$. L'affaiblissement avec Π nous donne alors la preuve voulue pour vérifier le premier point de l'hypothèse de récurrence.

Nous n'avons pas introduit de contrainte supplémentaire dans F_χ, le second point se prouve donc de la même manière que dans le cas de la règle d'affaiblissement.

Passons au dernier point de l'hypothèse de récurrence. Soit $T, \chi\theta_\chi \vdash t$ $[\![c\theta_\chi]\!]$ un séquent de Π_{θ_χ}. Si celui-ci est dans l'une des hypothèses enracinées dans π_3, par maximalité de π_1, la dernière règle est une composition et donc la récurrence nous garantit que le séquent considéré est $F(V, T, \Sigma, \xi, \chi\theta_\chi), E'_f$-admissible. Si le séquent se trouve dans π_3, de par la forme des règles de composition, le terme du séquent est un sous-terme de u'. Or, ce terme est égal modulo E'_f à v qui est dans $F(V, T, \Sigma, \xi, \chi\theta_\chi)$. La contrainte est la combinaison de contraintes apparaissant dans certains des séquents utilisés comme hypothèses de π_3. Ces séquents étant $F(V, T, \Sigma, \xi, \chi\theta_\chi), E'_f$-admissible, le séquent que l'on considère l'est donc aussi. Enfin, si le séquent considéré est la conclusion de Π_{θ_χ}, il s'agit de la combinaison du terme w qui est dans V et d'une contrainte qui est combinaison de :
- $E'\theta_\chi$ qui est la contrainte du séquent $T, \chi\theta_\chi \vdash u'$ $[\![E'\theta_\chi]\!]$ dont nous venons de voir qu'il est $F(V, T, \Sigma, \xi, \chi\theta_\chi), E'_f$-admissible ;
- $\bigwedge x_i = u'_i$ dont les membres droits sont ou bien des variables de χ, ou bien des termes égaux modulo E'_f à des termes de $F(V, T, \Sigma, \xi, \chi\theta_\chi)$, par construction ;
- $x_{R,k,s} = 1$ avec $1 \in T$;
- σ_S dont les membre droits sont dans Σ par définition.

Ce séquent est donc également $F(V, T, \Sigma, \xi, \chi\theta_\chi), E'_f$-admissible.

Les conditions de l'hypothèse de récurrence sont donc également vérifiées dans le cas où la dernière règle est une règle de protocole.

Règle de composition . Dans ce cas, aucune des prémisses ne se termine par une règle d'instanciation dont le contexte n'est pas vide. La preuve Π est de la forme suivante, avec $s = f(u_1, \ldots, u_k)$:

$$\dfrac{\dfrac{\Pi_1}{T \vdash u_1 \ [\![E_1]\!]} \quad \ldots \quad \dfrac{\Pi_k}{T \vdash u_k \ [\![E_k]\!]}}{T \vdash s \ [\![E_1 \wedge \ldots \wedge E_k]\!]}\,\mathcal{C}$$

On applique l'hypothèse de récurrence sur les preuves $\Pi_1, \ldots, \Pi_k$. Ceci est possible du fait que, par hypothèse sur ce point, ces preuves ne se terminent pas par des instanciations dont le contexte est non vide. À partir des différentes contraintes F^i_χ obtenues, on construit l'ensemble F_χ par union des F^i_χ. Il suffit pour cela de généraliser le cas de la règle d'affaiblissement à un nombre arbitraire de prémisses. Le premier point de l'hypothèse de récurrence est alors validé (on procède comme pour le cas de la règle d'affaiblissement).

Soit θ_χ une substitution modèle de F_χ. Elle est également modèle des F^i_χ. Nous avons, via l'hypothèse de récurrence, un ensemble de preuves $\Pi^i_{\theta_\chi}$ de $T, \chi\theta_\chi \vdash u_i$ $[\![E'_i\theta_\chi]\!]$. L'application de la règle de $\mathcal{S}$ sur ces preuves permet d'obtenir une preuve Π' de $T, \chi\theta_\chi \vdash u$ $[\![\bigwedge_i E'_i\theta_\chi]\!]$. En effet, contrairement au cas des règles de décomposition que nous verrons par la suite, il n'y a pas de problème de linéarité à traiter : les contraintes n'ont pas d'importance pour l'application d'une règle de composition. Toutefois, la preuve obtenue n'est pas celle que nous cherchons. En effet, si l'une des preuves se termine par une composition ou une règle d'affaiblissement, nous devons appliquer la propriété de localité pour que le dernier point de l'hypothèse de récurrence soit vérifié.

Nous appliquons d'abord le lemme 10.2 page 83 sur la dernière règle de la preuve Π'. Le chapeau de preuve maximal π_1 ne contenant que des règles de $\mathcal{S}$ et des règles d'affaiblissement est transformée en un nouveau chapeau dont les règles sont réarrangées comme indiqué sur la figure 10.2 page 84. On considère la partie ne contenant que des règles de $\mathcal{S}$ et on y applique la propriété de localité pour obtenir la preuve π_3. Cette preuve prend la place de la sous-preuve ne contenant que des règles de $\mathcal{S}$ dans π_2 pour obtenir π_4. Cette dernière prend la place de π_1 dans Π'. On obtient ainsi la preuve Π_{θ_χ}. La figure 10.7 page 90 illustre les transformations effectuées dans ce paragraphe.

En ce qui concerne les deux premiers points de l'hypothèse de récurrence, l'argumentaire est exactement le même que pour le cas où la dernière règle est une règle d'affaiblissement. Reste à considérer le dernier point.

Soit P_1 une sous-preuve de Π_{θ_χ} dont la conclusion est le séquent $T, \chi\theta_\chi \vdash v \ [\![c\theta_\chi]\!]$ et $T, \chi\theta_\chi \vdash u \ [\![c'\theta_\chi]\!]$ un séquent de P_1. Si P_1 est enracinée en dehors de π_4, on utilise simplement l'hypothèse de récurrence pour conclure. Dans le cas contraire, si le séquent est en dehors de π_4, il fait partie d'une preuve enracinée dans l'une des hypothèses de π_4. Or, par maximalité de π_1, la conclusion des preuves enracinées dans les hypothèses de π_4 proviennent d'une règle d'axiome, d'une instanciation ou d'une règle de protocole. Le séquent est donc $F(V, T, \Sigma, \xi, \chi\theta_\chi), E'_f$-admissible en utilisant l'hypothèse de récurrence.

Si le séquent est la conclusion d'une règle d'affaiblissement de π_4 (et donc, il n'est pas dans π_3), on se retrouve dans un cas similaire à celui où la dernière règle de Π est une règle d'affaiblissement : le séquent est la composition du terme d'un séquent utilisé en hypothèse de π_4 et de la conjonction d'un certain nombre de contraintes utilisées en hypothèse de π_4. Les séquents en question sont $F(V, T, \Sigma, \xi, \chi\theta_\chi), E'_f$-admissible et donc ce séquent l'est également.

Si le séquent est dans π_3, nous devons exploiter la propriété de localité pour conclure. Par maximalité de π_1, les hypothèses de π_3 sont $F(V, T, \Sigma, \xi, \chi\theta_\chi), E'_f$-admissible, notamment, les termes sont égaux modulo E'_f à des termes dans $F(V, T, \Sigma, \xi, \chi\theta_\chi)$. La propriété de localité nous indique alors que tous les termes dans π_3 sont égaux modulo E'_f à des termes de $F(V, T, \Sigma, s, \xi, \chi\theta_\chi)$ ou de $F(V, T, \Sigma, \xi, \chi\theta_\chi)$ selon que la dernière règle la plus à gauche qui n'est pas une règle d'affaiblissement est ou n'est pas (respectivement) une règle de composition.

On notera que l'utilisation de la coupure de π_3 à cet endroit pour conclure nous impose d'utiliser la propriété de clôture de F : $F(F(T)) = F(T)$.

Les contraintes des séquents situés dans π_3 sont des combinaisons des contraintes des séquents des hypothèses qui étaient $F(V, T, \Sigma, \xi, \chi\theta_\chi), E'_f$admissible et donc les séquents de π_3 sont $F(V, T, \Sigma, s, \xi, \chi\theta_\chi), E'_f$admissible ou $F(V, T, \Sigma, \xi, \chi\theta_\chi), E'_f$admissible.

L'hypothèse de récurrence est alors vérifiée.

Règle de décomposition . Dans ce cas, il est possible que l'une au moins des prémisses termine par une règle d'instanciation dont le contexte n'est pas vide, d'après le lemme 9.4 page 78.

Contrairement aux autres cas, nous allons bien faire la distinction entre Π_χ et Π_{θ_χ} : cette distinction est nécessaire pour mener proprement la récurrence.
Pour chaque i, on considère la preuve Π_i de $T \vdash v_{1i}\ [\![E_i]\!]$ qui mène à la preuve de $T \vdash v_{n_i i}\ [\![E_i]\!]$ en utilisant uniquement des instanciations dont le contexte n'est pas vide.

$$\Pi = \cfrac{\begin{array}{ccc} \cfrac{\Pi_1}{T \vdash v_{11}\ [\![E_1]\!]} & & \cfrac{\Pi_n}{T \vdash v_{1n}\ [\![E_n]\!]} \\ \vdots\ \mathcal{I} & & \vdots\ \mathcal{I} \\ T \vdash v_{n_1 1}\ [\![E_1]\!] & \ldots & T \vdash v_{n_n n}\ [\![E_n]\!] \end{array}}{T \vdash u\ [\![E_1 \wedge \ldots \wedge E_n]\!]}\ \mathcal{D}$$

Il est alors possible d'appliquer l'hypothèse de récurrence sur les preuves $\Pi_1, \ldots, \Pi_n$. On obtient alors les contraintes $F_\chi^1, \ldots, F_\chi^n$ et comme pour les règles d'affaiblissement, il est possible de construire à partir de ceux-ci F_χ tel que le second point de l'hypothèse de récurrence soit vérifié. Nous allons toutefois enrichir par la suite F_χ de contraintes supplémentaires. Nous devrons donc nous assurer que cet ensemble vérifie toujours le second point de l'hypothèse de récurrence.
L'hypothèse de récurrence nous fournit aussi les arbres de preuve $\Pi'_1, \ldots, \Pi'_n$ de, respectivement, $T, \chi \vdash v_{11}\ [\![E'_1]\!], \ldots, T, \chi \vdash v_{1n}\ [\![E'_n]\!]$ où $E'_i = \bigwedge_{x=t \in E_i} x = \rho(x, t)$.
Appliquons sur chacune des preuves Π'_i les instanciations qui étaient appliquées sur les preuves Π_i, dans le même ordre. Nous obtenons les arbres de preuve suivants :

$$\begin{array}{c} \cfrac{\cfrac{\Pi'_i}{T, \chi \vdash v_{1i}\ [\![E'_i]\!]}}{T, \chi \vdash v'_{2i}\ [\![E'_i]\!]}\ \mathcal{I} \\ \vdots\ \mathcal{I} \\ T, \chi \vdash v'_{n_i i}\ [\![E'_i]\!] \end{array}$$

En effet, on peut montrer par récurrence que si la première instanciation peut être appliquée, les suivantes aussi. Toutes les instanciations sont des instanciations à profondeur 1, d'après le lemme 9.5 page 79. Si, dans la preuve originale, l'instanciation instancie une variable en une variable, l'instanciation suivante peut toujours être appliquée dans la preuve transformée car ρ ne transforme pas les variables de V. Par contre, si l'instanciation instancie une variable en un terme, l'instanciation suivante ne peut agir au niveau de ce terme : les variables qui en dépendent se retrouvent au niveau 2. Si une instanciation suit, elle s'applique sur une autre partie du terme, non touché par l'instanciation précédente et peut donc toujours s'appliquer.
La première instanciation peut toujours s'appliquer car elle se fait sur le terme v_{1i}, que ce soit dans la preuve originale ou dans la preuve modifiée.
Lors de ces instanciations, il est possible que l'un des $v'_{n_i i}$ se retrouve avec une variable de χ à profondeur 1. Cela nous pose problème si la règle de décomposition nécessite un sous-terme de la forme $f(x_1, \ldots, x_k)$ à cette profondeur. Dans ce cas, on sait que la variable χ est du type $\chi^x_{f(t_1,\ldots,t_k)}$. On ajoute alors la contrainte suivante dans F_χ (si elle n'existe pas, sinon il n'y a rien à faire) :

$$\chi^x_{f(t_1,\ldots,t_k)} = f(\chi^{i_1}_{t_1}, \ldots, \chi^{i_k}_{t_k})$$

Les entiers $i_1, \ldots, i_k$ n'ont jamais été utilisés en indice de variables de χ dans la preuve. La contrainte ajoutée est bien de la forme souhaitée (quatrième forme) : en effet, grâce à l'hypothèse de récurrence, nous savons que la preuve de $T \vdash f(t_1, \ldots, t_k)\ [\![E'_i]\!]$ termine par une règle de construction.

Nous devons vérifier qu'avec une telle contrainte supplémentaire, le second point de l'hypothèse de récurrence est toujours vérifié. Étant donné que les variables $\chi_{t_1}^{i_1}, \ldots, \chi_{t_k}^{i_k}$ viennent d'être introduites, elles sont minimales dans l'ordre d'occurence dans F_χ. Le second point est donc vérifié. Nous pourrons être amenés par la suite à modifier cette nouvelle contrainte pour satisfaire aux non-linéarités de la règle de décomposition : nous serons alors amenés à utiliser le deuxième cas possible.

Ainsi, séparément, chaque $v'_{n_i i}$ peut s'unifier avec le terme t_i correspondant de la règle de décomposition. Toutefois, l'application de la règle de décomposition nécessite de vérifier si les non linéarités sont toujours satisfaites. Choisissons alors une non-linéarité, c'est-à-dire deux positions distinctes x et y correspondant à la même variable dans deux termes t_i^1 et t_j^2 (i peut être égal à j) de la règle de décomposition. Ces deux termes s'unifient respectivement avec $v_{n_i i}$ et $v_{n_j j}$. Chaque variable peut être à profondeur 0, 1 ou 2. Il y a donc 6 cas à traiter en prenant en compte les symétries.

Les deux variables sont à profondeur 0 . Dans la preuve originale, ces deux variables s'unifiaient donc avec le même terme. Si on a $v'_{n_i i} = v'_{n_j j}$, inutile d'aller plus loin. Si ce n'est pas le cas, disons que $i < j$ et appliquons la règle de réécriture de la figure 10.10.

$$\dfrac{\cdots \quad \begin{array}{c} \dfrac{\dfrac{\Pi'_i}{T,\chi \vdash v_{1i}\ [\![E'_i]\!]}}{T,\chi \vdash v'_{2i}\ [\![E'_i]\!]}\mathcal{I} \\ \vdots\,\mathcal{I} \\ T,\chi \vdash v'_{n_i i}\ [\![E'_i]\!] \end{array} \quad \cdots \quad \begin{array}{c} \dfrac{\dfrac{\Pi'_j}{T,\chi \vdash v_{1j}\ [\![E'_j]\!]}}{T,\chi \vdash v'_{2j}\ [\![E'_j]\!]}\mathcal{I} \\ \vdots\,\mathcal{I} \\ T,\chi \vdash v'_{n_j j}\ [\![E'_j]\!] \end{array} \quad \cdots}{T,\chi \vdash \cdots\ [\![E'_1 \wedge \ldots \wedge E'_n]\!]}\mathcal{D}$$

$$\Longrightarrow \quad \dfrac{\cdots \quad \begin{array}{c} \dfrac{\dfrac{\Pi'_i}{T,\chi \vdash v_{1i}\ [\![E'_i]\!]}}{T,\chi \vdash v'_{2i}\ [\![E'_i]\!]}\mathcal{I} \\ \vdots\,\mathcal{I} \\ T,\chi \vdash v'_{n_i i}\ [\![E'_i]\!] \end{array} \quad \cdots \quad \begin{array}{c} \dfrac{\dfrac{\dfrac{\Pi'_i}{T,\chi \vdash v_{1i}\ [\![E'_i]\!]} \quad \dfrac{\Pi'_j}{T,\chi \vdash v_{1j}\ [\![E'_j]\!]}}{T,\chi \vdash v_{1i}\ [\![E'_i \wedge E'_j]\!]}\mathcal{W}}{T,\chi \vdash v'_{2i}\ [\![E'_i \wedge E'_j]\!]}\mathcal{I} \\ \vdots\,\mathcal{I} \\ T,\chi \vdash v'_{n_i i}\ [\![E'_i \wedge E'_j]\!] \end{array} \quad \cdots}{T,\chi \vdash \cdots\ [\![E'_1 \wedge \ldots \wedge E'_n]\!]}\mathcal{D}$$

FIG. 10.10 - *Règle de réécriture pour les décompositions dans le cas « 0-0 »*

La preuve Π'_j se voit de plus appliquer le lemme 10.1 afin de satisfaire au dernier point de l'hypothèse de récurrence.

Profondeur 0 et profondeur 1 . x est à profondeur 0 et y est à profondeur 1. La figure 10.11 présente les éléments de notation pour ce cas. On note p la position de s_1 dans $v_{n_j j}$ ($s_1 = v_{n_j j_{|p}}$).

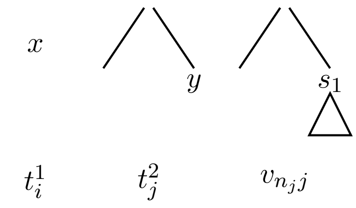

FIG. 10.11 - *Notations pour les décompositions dans le cas « 0-1 »*

Deux cas peuvent se présenter :
① $s_1 \neq v'_{n_j j_{|p}}$
② $s_1 = v'_{n_j j_{|p}}$
Dans le cas ①, le terme s_1 provient d'une instanciation de profondeur 1 d'une variable que l'on notera z. On a alors $v'_{n_j j_{|p}} = \chi^z_{s_1}$. L'hypothèse de récurrence nous fournit une preuve Π_{s_1} de $T \vdash s_1 \;\; \llbracket E_j \rrbracket$ qui se termine par une construction. On y applique l'hypothèse de récurrence pour obtenir une preuve Π'_{s_1} de $T, \chi \vdash s_1 \;\; \llbracket E'_j \rrbracket$. On utilise alors la règle de réécriture de la figure 10.12.

$$\dfrac{\cdots \quad \begin{array}{c} \dfrac{\dfrac{\Pi'_i}{T,\chi \vdash v_{1i} \;\; \llbracket E'_i \rrbracket}}{T,\chi \vdash v'_{2i} \;\; \llbracket E'_i \rrbracket}\mathcal{I} \\ \vdots\, \mathcal{I} \\ T,\chi \vdash v'_{n_i i} \;\; \llbracket E'_i \rrbracket \end{array} \quad \cdots \quad \begin{array}{c} \dfrac{\dfrac{\Pi'_j}{T,\chi \vdash v_{1j} \;\; \llbracket E'_j \rrbracket}}{T,\chi \vdash v'_{2j} \;\; \llbracket E'_j \rrbracket}\mathcal{I} \\ \vdots\, \mathcal{I} \\ T,\chi \vdash v'_{n_j j} \;\; \llbracket E'_j \rrbracket \end{array} \quad \cdots}{T,\chi \vdash \cdots \;\; \llbracket E'_1 \wedge \ldots \wedge E'_n \rrbracket}\mathcal{D}$$

$$\Longrightarrow \quad \dfrac{\cdots \quad \dfrac{\dfrac{\Pi'_{s_1}}{T,\chi \vdash \chi^z_{s_1} \;\; \llbracket E'_j \rrbracket} \quad \dfrac{\Pi'_i}{T,\chi \vdash v_{1i} \;\; \llbracket E'_i \rrbracket}}{T,\chi \vdash \chi^z_{s_1} \;\; \llbracket E'_j \wedge E'_i \rrbracket}\mathcal{W} \quad \cdots \quad \begin{array}{c} \dfrac{\dfrac{\Pi'_j}{T,\chi \vdash v_{1j} \;\; \llbracket E'_j \rrbracket}}{T,\chi \vdash v'_{2j} \;\; \llbracket E'_j \rrbracket}\mathcal{I} \\ \vdots\, \mathcal{I} \\ T,\chi \vdash v'_{n_j j} \;\; \llbracket E'_j \rrbracket \end{array} \quad \cdots}{T,\chi \vdash \cdots \;\; \llbracket E'_1 \wedge \ldots \wedge E'_n \rrbracket}\mathcal{D}$$

FIG. 10.12 - *Règle de réécriture pour les décompositions dans le cas « 0-1 »*

La preuve Π_i' se voit de plus appliquer le lemme 10.1 afin de satisfaire au dernier point de l'hypothèse de récurrence.

Dans le cas ②, deux nouveaux cas peuvent se présenter. S'il est le résultat de l'instanciation d'une variable z à profondeur 1, il existe un terme $s_2 =_{E_f'} s_1$ et tel que $s_2 \in F(V,T,\Sigma,s_f,\xi)$. Dans le cas contraire, z serait affecté à $\chi^z_{s_1}$ au lieu de s_1, comme ci-dessus. $v_{n_i i} = s_1 =_{E_f'} s_2 \in F(V,T,\Sigma,s_f,\xi)$ et du fait de la clôture de F par le sous-termes, tous les sous-termes de $v_{n_i i}$ sont dans $F(V,T,\Sigma,s_f,\xi)$ et donc $v'_{n_i i} = v_{n_i i} = s_1$ (car chacun des v_{ji} sont dans $F(V,T,\Sigma,s_f,\xi)$ et donc la fonction ρ correspond à l'identité sur ces termes). On a bien $x = y$.

Si s_1 n'est pas le résultat d'une instanciation, il est présent au même endroit dans v_{1j}. Si $v'_{n_i i} \neq v_{n_i i}$, on considère les instanciations $z = s_3$ qui ont abouti à $v_{n_i i}$ et telles que $\rho(z,s_3) = \chi^z_{s_3}$. Ces instanciations sont responsables de la différence entre $v_{n_i i}$ et $v'_{n_i i}$. Par clôture de la fonction F par sous-termes, nous savons donc qu'il n'existe pas de terme $s_2 =_{E_f'} s_1$ tel que $s_2 \in F(V,T,\Sigma,s_f,\xi)$ (les s_3 sont des sous-termes de s_1). L'hypothèse de récurrence nous permet donc d'affirmer que v_{1j} est le résultat de règles de compositions.

Considérons alors la preuve Π_j'. Comme dans le cas d'une règle de protocole, nous pouvons construire une preuve Π_j'' telle que s_1 se trouve remplacé par un terme arbitraire grâce au fait que les compositions sont le simple ajout d'un symbole en tête. Dans notre cas, nous choisissons le terme $v'_{n_i i}$. Nous obtenons alors une preuve de $T,\chi \vdash v_{1j}[s_1 \leftarrow v'_{n_i i}]\ \ [\![E_j' \wedge E_i']\!]$. Nous appliquons l'ensemble des instanciations qui ne portaient pas sur la position p de s_1 pour obtenir une preuve de $T,\chi \vdash v'_{n_j j}[s_1 \leftarrow v'_{n_i i}]\ \ [\![E_j' \wedge E_i']\!]$. Cette transformation est présentée dans la figure 10.13.

$$
\dfrac{\cdots\ \ \dfrac{\dfrac{\dfrac{\Pi_i'}{T,\chi \vdash v_{1i}\ \ [\![E_i']\!]}}{T,\chi \vdash v'_{2i}\ \ [\![E_i']\!]}\mathcal{I}}{\vdots\,\mathcal{I}} \ \ T,\chi \vdash v'_{n_i i}\ \ [\![E_i']\!] \quad \cdots \quad \dfrac{\dfrac{\dfrac{\Pi_j'}{T,\chi \vdash v_{1j}\ \ [\![E_j']\!]}}{T,\chi \vdash v'_{2j}\ \ [\![E_j']\!]}\mathcal{I}}{\vdots\,\mathcal{I}}\ \ T,\chi \vdash v'_{n_j j}\ \ [\![E_j']\!] \quad \cdots}{T,\chi \vdash \cdots\ \ [\![E_1' \wedge \ldots \wedge E_n']\!]}\mathcal{D}
$$

$$
\Longrightarrow\quad \dfrac{\cdots\ \ \dfrac{\dfrac{\dfrac{\Pi_i'}{T,\chi \vdash v_{1i}\ \ [\![E_i']\!]}}{T,\chi \vdash v'_{2i}\ \ [\![E_i']\!]}\mathcal{I}}{\vdots\,\mathcal{I}}\ \ T,\chi \vdash v'_{n_i i}\ \ [\![E_i']\!] \quad \cdots \quad \dfrac{\dfrac{\dfrac{\Pi_j''}{T,\chi \vdash v_{1j}[s_1 \leftarrow v'_{n_i i}]\ \ [\![E_j' \wedge E_i']\!]}}{T,\chi \vdash v'_{2j}[s_1 \leftarrow v'_{n_i i}]\ \ [\![E_j' \wedge E_i']\!]}\mathcal{I}}{\vdots\,\mathcal{I}}\ \ T,\chi \vdash v'_{n_j j}[s_1 \leftarrow v'_{n_i i}]\ \ [\![E_j' \wedge E_i']\!] \quad \cdots}{T,\chi \vdash \cdots\ \ [\![E_1' \wedge \ldots \wedge E_n']\!]}\mathcal{D}
$$

FIG. 10.13 - *Seconde règle de réécriture pour les décompositions dans le cas « 0-1 »*

L'application d'une telle règle de réécriture nous permet d'assurer que $x = y$.

Profondeur 0 et profondeur 2 . On considère que la variable x est à profondeur 0 et correspond au terme $v_{n_i i}$. Les notations sont présentées dans la figure 10.14.

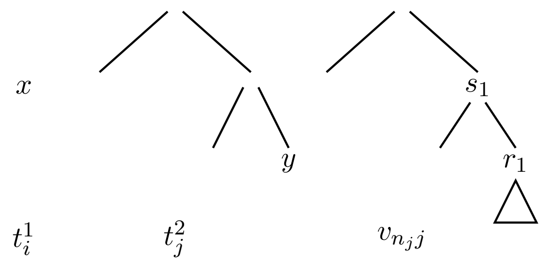

FIG. 10.14 - *Notations pour les décompositions dans le cas « 0-2 »*

Nous procédons de manière très similaire au cas précédent. Le premier cas est identique mot pour mot. Passons au cas où s_1 est présent à la même position dans $v_{n_j j}$ et $'_{n_j j}$.
S'il est le résultat de l'instanciation d'une variable z à profondeur 1, on procède de la même façon que le cas « 0-1 » à l'exception que $v_{n_i i}$ n'est plus égal à s_1 mais à s_2. Il est cependant sous-terme de s_1. Le cas contraire est de nouveau identique au cas correspondant dans « 0-1 ». La même règle de réécriture s'applique.

Profondeur 1 et profondeur 2 . On considère que la variable x est à profondeur 1. Les notations employées sont reportées sur la figure 10.15. On note p la position de s_1 dans $v_{n_j j}$: $s_1 = v_{n_j j_{|p}}$.

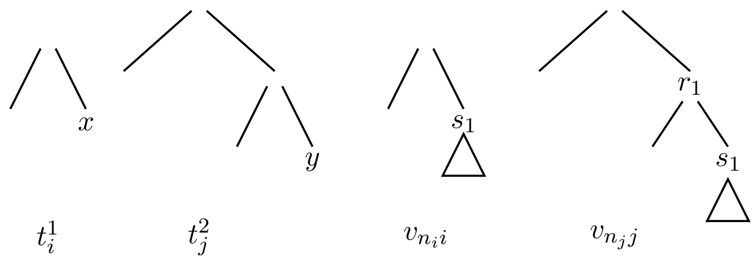

FIG. 10.15 - *Notations pour les décompositions dans le cas « 1-2 »*

Deux cas peuvent se présenter :
① $s_1 = v'_{n_j j_{|p}}$
② $s_1 \neq v'_{n_j j_{|p}}$
Considérons le cas ①. On note p' la position de s_1 dans $v_{n_i i}$: $s_1 = v_{n_i i_{|p'}}$. Si $s_1 \neq v'_{n_i i_{|p'}}$, s_1 dans la branche menant à $v_{n_i i}$ est le résultat d'une instanciation d'une variable z à profondeur 1 et qu'il n'est pas égal, modulo E'_f à un terme de $F(V, T, \Sigma, s_f, \xi)$, dans $v'_{n_i i}$, il se retrouve remplacé par $\chi^z_{s_1}$: $v'_{n_i i_{|p'}} = \chi^z_{s_1}$. Comme s_1 se retrouve dans $v_{n_j j}$ et à la même position p dans $v'_{n_j j}$, il n'est pas le résultat d'une instanciation et se trouve également, à la position p dans v_{1j}. On emploie alors le même raisonnement que dans le cas « 0-1 », à savoir que ce terme est le

résultat d'une composition et que l'on peut substituer s_1 par $\chi^z_{s_1}$ (dont on a une preuve car c'est une variable de χ) et ainsi obtenir l'égalité $x = y$. La règle de réécriture correspondante est présentée dans la figure 10.16.

$$
\dfrac{\cdots\quad \begin{array}{c} \dfrac{\dfrac{\Pi'_i}{T,\chi \vdash v_{1i}\ \ [\![E'_i]\!]}}{T,\chi \vdash v'_{2i}\ \ [\![E'_i]\!]}\,\mathcal{I} \\ \vdots\,\mathcal{I} \\ T,\chi \vdash v'_{n_i i}\ \ [\![E'_i]\!] \end{array} \quad\cdots\quad \begin{array}{c} \dfrac{\dfrac{\Pi'_j}{T,\chi \vdash v_{1j}\ \ [\![E'_j]\!]}}{T,\chi \vdash v'_{2j}\ \ [\![E'_j]\!]}\,\mathcal{I} \\ \vdots\,\mathcal{I} \\ T,\chi \vdash v'_{n_j j}\ \ [\![E'_j]\!] \end{array} \quad\cdots}{T,\chi \vdash \cdots\ \ [\![E'_1 \wedge \ldots \wedge E'_n]\!]}\,\mathcal{D}
$$

$$
\Longrightarrow\quad \dfrac{\cdots\quad \begin{array}{c} \dfrac{\dfrac{\Pi'_i}{T,\chi \vdash v_{1i}\ \ [\![E'_i]\!]}}{T,\chi \vdash v'_{2i}\ \ [\![E'_i]\!]}\,\mathcal{I} \\ \vdots\,\mathcal{I} \\ T,\chi \vdash v'_{n_i i}\ \ [\![E'_i]\!] \end{array} \quad\cdots\quad \begin{array}{c} \dfrac{\dfrac{\Pi''_j}{T,\chi \vdash v_{1j}[s_1 \leftarrow \chi^z_{s_1}]\ \ [\![E'_j \wedge E'_i]\!]}}{T,\chi \vdash v'_{2j}[s_1 \leftarrow \chi^z_{s_1}]\ \ [\![E'_j \wedge E'_i]\!]}\,\mathcal{I} \\ \vdots\,\mathcal{I} \\ T,\chi \vdash v'_{n_j j}[s_1 \leftarrow \chi^z_{s_1}]\ \ [\![E'_j \wedge E'_i]\!] \end{array} \quad\cdots}{T,\chi \vdash \cdots\ \ [\![E'_1 \wedge \ldots \wedge E'_n]\!]}\,\mathcal{D}
$$

FIG. 10.16 - *Règle de réécriture pour les décompositions dans le cas « 1-2 »*

Considérons maintenant le cas ②. r_1 est le résultat de l'instanciation à profondeur 1 d'une variable z. Dans $v'_{n_j j}$, r_1 se retrouve remplacé par $\chi^z_{r_1}$. Cela signifie que r_1 n'est pas égal, modulo E'_f à un terme de $F(V,T,s_f,\Sigma,\xi)$. Nous utilisons alors la première hypothèse de la section 10.4 page 94 pour conclure que $v_{n_i i}$ n'est pas non plus égal, modulo E'_f à un terme de $F(V,T,s_f,\Sigma,\xi)$. On sait également que le symbole de tête, g, de r_1 est unaire. Afin de pouvoir appliquer la règle de décomposition, on avait introduit dans F_χ une contrainte $\chi^z_{r_1} = g(\chi^k_{s_1})$.
Si v_{1i} est le résultat de compositions, on effectue les substitutions de s_1 par $\chi^k_{s_1}$ pour avoir l'égalité $x = y$. La règle de réécriture est donnée dans la figure 10.17 page suivante. Π''_i est obtenue de façon similaire à Π''_j.
Dans le cas contraire où v_{1i} n'est pas le résultat de compositions et donc il est égal (ainsi que ses sous-termes), modulo E'_f à un terme de $F(V,T,\Sigma,s_f,\xi)$. Dans la suite de la branche, s_1 est introduit par l'instanciation d'une variable w à profondeur 1. Dans E'_i, la contrainte correspondante est devenue $w = \chi^w_{s_1}$ via l'application de la fonction ρ sur la contrainte $w = s_1$. On ajoute alors dans F_χ la contrainte $\chi^k_{s_1} = \chi^w_{s_1}$. Nous devons alors vérifier le second point de la récurrence vis-à-vis de la variable $\chi^z_{r_1} = g(\chi^k_{s_1})$. Lors de l'ajout de cette contrainte, la variable $\chi^k_{s_1}$ venait d'être introduite et se trouvait donc minimale. Il est possible que cela ne soit plus le cas. Dans ce cas, le second point de l'hypothèse de récurrence est tout de même vérifié : $f(w)$ est dans $F(V,T,\Sigma,s_f,\chi)$ (dans le cas contraire, v_{1i} serait le résultat de compositions). On utilise alors la première hypothèse de la section 10.4 page 94

$$\dfrac{\cdots \quad \begin{array}{c}\Pi'_i\\ \hline T,\chi\vdash v_{1i}\ [\![E'_i]\!] \\ \hline T,\chi\vdash v'_{2i}\ [\![E'_i]\!] \\ \vdots\,\mathcal{I} \\ T,\chi \vdash v'_{n_i i}\ [\![E'_i]\!]\end{array}\ \mathcal{I} \quad \cdots \quad \begin{array}{c}\Pi'_j\\ \hline T,\chi\vdash v_{1j}\ [\![E'_j]\!] \\ \hline T,\chi\vdash v'_{2j}\ [\![E'_j]\!] \\ \vdots\,\mathcal{I} \\ T,\chi \vdash v'_{n_j j}\ [\![E'_j]\!]\end{array}\ \mathcal{I} \quad \cdots}{T,\chi\vdash\cdots\ [\![E'_1\wedge\ldots\wedge E'_n]\!]}\ \mathcal{D}$$

$$\Longrightarrow \quad \dfrac{\cdots \quad \begin{array}{c}\Pi''_i\\ \hline T,\chi\vdash v_{1i}[s_1\leftarrow\chi^k_{s_1}]\ [\![E'_i\wedge E'_j]\!] \\ \hline T,\chi\vdash v'_{2i}[s_1\leftarrow\chi^k_{s_1}]\ [\![E'_i\wedge E'_j]\!] \\ \vdots\,\mathcal{I} \\ T,\chi \vdash v'_{n_i i}[s_1\leftarrow\chi^k_{s_1}]\ [\![E'_i\wedge E'_j]\!]\end{array}\ \mathcal{I} \quad \cdots \quad \begin{array}{c}\Pi'_j\\ \hline T,\chi\vdash v_{1j}\ [\![E'_j]\!] \\ \hline T,\chi\vdash v'_{2j}\ [\![E'_j]\!] \\ \vdots\,\mathcal{I} \\ T,\chi \vdash v'_{n_j j}\ [\![E'_j]\!]\end{array}\ \mathcal{I} \quad \cdots}{T,\chi\vdash\cdots\ [\![E'_1\wedge\ldots\wedge E'_n]\!]}\ \mathcal{D}$$

FIG. 10.17 - *Seconde règle de réécriture pour les décompositions dans le cas « 1-2 »*

pour conclure que $g(w)$ est également dans $F(V,T,\Sigma,s_f,\chi)$.

Les deux variables sont à profondeur 1 . La figure 10.18 indique les notations utilisées dans ce cas. On note p (respectivement p') la position de s_1 dans v_{n_ii} (respectivement v_{n_jj}) : $s_1 = v_{n_ii|p}$ (respectivement $s_1 = v_{n_jj|p'}$).

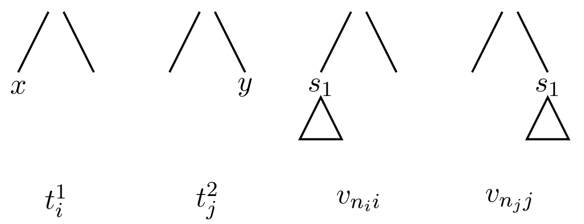

FIG. 10.18 - *Notations pour les décompositions dans le cas « 1-1 »*

Supposons d'abord que $s_1 \neq v'_{n_ii|p}$. Il s'agit du résultat de l'instanciation à profondeur 1 d'une variable z. Cette variable se retrouve liée à $\chi^z_{s_1}$ dans E'_i.

Si $s_1 = v'_{n_jj|p'}$, s_1 est aussi présent dans v_{1j} à la position p' et il est alors le résultat de compositions. On applique alors la même règle de réécriture que celle de la figure 10.16 page précédente.

Si $s_1 \neq v'_{n_jj|p'}$, il s'agit aussi du résultat de l'instanciation à profondeur 1 d'une variable z'. Si $z \neq z'$, on ajoute simplement à F_χ la contrainte $\chi^z_{s_1} = \chi^{z'}_{s_1}$. Cette opération est sans importance pour le second point de l'hypothèse de récurrence. Le cas $s_1 = v'_{n_i i|p}$ et $s_1 \neq v'_{n_j j|p'}$ est symétrique au cas que l'on vient de traiter. Le cas $s_1 = v'_{n_i i|p}$ et $s_1 = v'_{n_j j|p'}$ est trivial.

Les deux variables sont à profondeur 2 . Les notations utilisées sont indiquées sur la figure 10.19. On note p (respectivement p') la position de s_1 dans $v_{n_i i}$ (respectivement $v_{n_j j}$) : $s_1 = v_{n_i i|p}$ (respectivement $s_1 = v_{n_j j|p'}$).

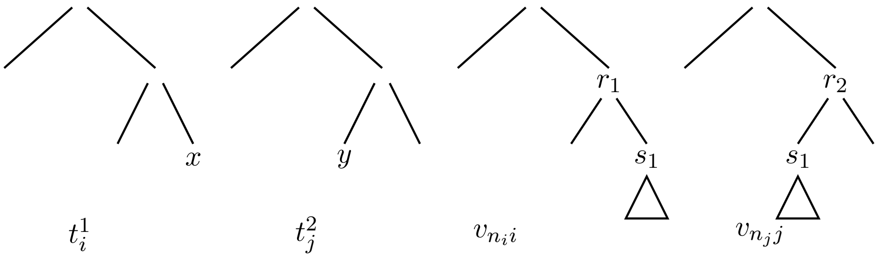

FIG. 10.19 - *Notations pour les décompositions dans le cas « 2-2 »*

Supposons que $s_1 \neq v'_{n_i i|p}$. Le terme r_1 est donc le résultat d'une instanciation à profondeur 1 de la variable z. La seconde propriété de clôture de F de la section 10.4 page 94 nous indique que son symbole de tête f est unaire. $r_1 = f(s_1)$ est donc remplacé dans $v'_{n_i i}$ par $\chi^z_{f(s_1)}$ et la contrainte $\chi^z_{f(s_1)} = f(\chi^k_{s_1})$ a été ajoutée dans F_χ.

La seconde propriété de clôture de F de la section 10.4 page 94 nous indique également que $r_2 = g(s_1)$ n'est pas égal, modulo E'_f, à un terme de $F(V, T, \Sigma, s_f, \xi)$ (car r_1 ne l'est pas).

Si $s_1 \neq v'_{n_j j|p'}$, r_2 est le résultat d'une instanciation à profondeur 1 d'une variable z'. Comme pour r_1, il se trouve remplacé par $\chi^{z'}_{g(s_1)}$ et la contrainte $\chi^{z'}_{g(s_1)} = g(\chi^{k'}_{s_1})$ a été ajoutée à F_χ. Pour satisfaire l'égalité $x = y$, on ajoute la nouvelle contrainte $\chi^k_{s_1} = \chi^{k'}_{s_1}$. Cet ajout peut perturber l'ordre d'occurrence dans F_χ de $\chi^k_{s_1}$ ou de $\chi^{k'}_{s_1}$: ces deux variables deviennent égales dans ce pré-ordre d'occurrence. Supposons que $\chi^k_{s_1}$ était auparavant minimale mais que $\chi^{k'}_{s_1}$ ne l'était pas. Cela signifie qu'il existe une variable w telle que $\chi^{k'}_{s_1} =_{F_\chi} \chi^w_{s_1}$ et $g(w)$ est dans $F(V, T, \Sigma, s_f, \xi)$. La propriété de clôture de F nous indique alors que $f(w)$ l'est aussi. Les autres cas sont symétriques ou triviaux. Le second point de l'hypothèse de récurrence est donc bien vérifié.

Si $s_1 = v'_{n_j j|p'}$ (r_2 n'a pas été modifié), on utilise une règle de réécriture proche de celle de la figure 10.16 page 111 : le terme provient d'une série de constructions et il est possible de modifier celles-ci pour remplacer s_1 par $\chi^k_{s_1}$ (et non par $\chi^z_{s_1}$ comme indiqué sur la figure).

Nous avons traité les règles de décomposition comme si elles n'étaient pas linéaires alors que nous avons justement linéarisé celles-ci. Cette approche ignorant la linéarisation est cependant valide. En effet, l'application de l'hypothèse de récurrence respecte les non linéarités dans les contraintes. Considérons la preuve suivante où l'on a $x = \alpha \wedge y = \beta \wedge E_1 \wedge \ldots \wedge E_n \models x = y$:

$$\dfrac{\cdots \quad \dfrac{\pi_i}{T \vdash C_1[x] \ \llbracket x = \alpha \wedge E_i \rrbracket} \quad \cdots \quad \dfrac{\pi_j}{T \vdash C_2[y] \ \llbracket y = \beta \wedge E_j \rrbracket} \quad \cdots}{T \vdash u \ \llbracket E_1 \wedge \ldots \wedge E_n \wedge x = \alpha \wedge y = \beta \rrbracket} \mathcal{D}$$

Rappelons que la contrainte $E_1 \wedge \ldots \wedge E_n \wedge x = \alpha \wedge y = \beta$ est sous forme factorisée. Comme elle est modèle de $x = y$, cela signifie que $x = y \in E_1 \wedge \ldots \wedge E_n \wedge x = \alpha \wedge y = \beta$ (se reporter à la section 8.1 page 65 pour des détails à ce sujet). La fonction ρ ne peut pas transformer des variables de V, cela signifie que $x = y$ apparaît toujours dans la contrainte résultant de l'application de l'hypothèse de récurrence.

Maintenant que les non linéarités sont respectées, il est possible d'appliquer la règle de décomposition pour obtenir un arbre de preuve $T, \chi \vdash u' \ \llbracket E'_1 \wedge \ldots \wedge E'_n \rrbracket$. Deux cas peuvent se présenter :

① $u = u'$ et tout va bien ;

② $u' \neq u$ et dans ce cas, il faut trouver une autre preuve.

Dans le cas ①, on doit cependant encore s'assurer que le dernier point de l'hypothèse de récurrence est vérifié. Soit $\theta_\chi \models F_\chi$. Le principal problème se situe si l'un des v_{1i} est le résultat d'une règle de composition. Si $T, \chi\theta_\chi \vdash v_{1i} \ \llbracket E'_i\theta_\chi \rrbracket$ est $F(V, T, \Sigma, \xi, \chi\theta_\chi), E'_f$-admissible, $T, \chi\theta_\chi \vdash v_{n_i i} \ \llbracket E'_i\theta_\chi \rrbracket$ l'est aussi. En effet, $v_{n_i i} =_{E'_F} v_{1i}$. Dans le cas où la branche actuelle ne contient pas de colonne d'instanciations à profondeur 1, l'utilisation de la propriété de localité permet de conclure (la dernière règle est une règle de décomposition).

S'il y a une colonne d'instanciations, on applique le lemme 10.1 page 83 au chapeau de preuve maximal ne contenant que des règles de composition et des règles d'affaiblissement. On utilise alors la règle de réécriture de la figure 10.20 pour repousser les compositions après les instanciations à profondeur 1, mais avant la règle de décomposition. θ est une instanciation à profondeur 1 et s'applique donc à l'un des u_j : $\theta(f(u_1, \ldots, u_m)) = f(u_1, \ldots, \theta(u_j), \ldots, u_m)$.

$$\dfrac{\dfrac{T, \chi \vdash u_1 \ \llbracket E_{1i} \rrbracket \ldots T, \chi \vdash u_m \ \llbracket E_{1m} \rrbracket}{T, \chi \vdash f(u_1, \ldots, u_m) \ \llbracket E_i \rrbracket} \mathcal{C}}{T, \chi \vdash \theta(f(u_1, \ldots, u_m)) \ \llbracket E_i \rrbracket} \mathcal{I}$$

$$\Longrightarrow \dfrac{T, \chi \vdash u_1 \ \llbracket E_{1i} \rrbracket \ldots \quad \dfrac{T, \chi \vdash u_j \ \llbracket E_{ji} \rrbracket}{T, \chi \vdash \theta(u_j) \ \llbracket E_{ji} \rrbracket} \mathcal{I} \quad \ldots T, \chi \vdash u_m \ \llbracket E_{1m} \rrbracket}{T, \chi \vdash f(u_1, \ldots, \theta(u_j), \ldots, u_m) \ \llbracket E_i \rrbracket} \mathcal{C}$$

FIG. 10.20 - *Règle de réécriture pour faire remonter les instanciations*

Une fois cette règle appliquée tant que possible, on se ramène au cas où la localité s'applique.

Considérons maintenant le cas ② où $u \neq u'$. Nous devons trouver une autre façon d'obtenir un arbre de preuve de $T, \chi \vdash u \ \llbracket E' \rrbracket$. La section 9.3.2 page 76 restreignait les formes possibles pour une règle de décomposition. Nous allons raisonner suivant la forme de celle-ci :

La conclusion est une variable de l'une des prémisses (ou un sous-terme à profondeur 1 de l'une des prémisses). Cette variable ne peut se trouver à profondeur 0 (la preuve ne serait pas minimale). On note p sa position dans $v_{n_i i}$: $v_{n_i i|p} = u$ et $v'_{n_i i|p} = u'$. Deux cas peuvent se présenter :

❶ la variable est à profondeur 1,

❷ la variable est à profondeur 2 (dans ce cas, la conclusion n'est pas un sous-terme à profondeur 1 de l'une des prémisses).

Considérons le cas ❶. u et u' sont le résultat d'une instanciation à profondeur 1 par une variable z. On a alors $u' = \chi^z_u$. L'hypothèse de récurrence nous donne une preuve de $T \vdash u \;\; [\![E_i]\!]$ car χ^z_u apparaît en membre droit de E'_i.

On applique l'hypothèse de récurrence sur cette preuve qui se termine par une composition ce qui nous permet d'obtenir la preuve Π''_i de $T, \chi \vdash u \;\; [\![E'_i]\!]$. On utilise des règles d'affaiblissement pour obtenir un arbre de preuve du séquent $T, \chi \vdash u \;\; [\![E'_1 \wedge \ldots \wedge E'_n]\!]$ en exploitant les preuves de $T, \chi \vdash v_{1j} \;\; [\![E'_j]\!]$. Éventuellement, les règles de composition qui terminent ces preuves sont transformées en règles d'affaiblissement grâce au lemme 10.1 page 83 (se reporter au cas de la règle d'affaiblissement qui détaille cette transformation).

Regardons maintenant le cas ❷ où la variable est à profondeur 2. C'est aussi suite à une instanciation à profondeur 1 d'une variable z par $f(u_1, \ldots, u, \ldots u_k)$ qui est le sur-terme immédiat de u dans la prémisse $v_{n_i i}$. On note p' la position de ce sur-terme dans $v_{n_i i}$. On a alors $v'_{n_i i|p'} = \chi^z_{f(u_1,\ldots,u,\ldots u_k)}$. L'hypothèse de récurrence nous fournit alors une preuve de $T \vdash f(u_1, \ldots, u, \ldots u_k) \;\; [\![E_i]\!]$ car $\chi^z_{f(u_1,\ldots,u,\ldots u_k)}$ apparaît en membre droit de E'_i. Cette preuve se termine par une composition. On utilise la règle de réécriture de la figure 10.21 page suivante.

On applique l'hypothèse de récurrence sur chacune des preuves π^l_i pour obtenir des preuves π'^l_i de $T, \chi \vdash u_l \;\; [\![E'^l_i]\!]$ avec $u_l = u$. On applique sur ces preuves les règles d'affaiblissement appliquées précédemment sur les preuves π^l_i pour obtenir une preuve de $T, \chi \vdash u \;\; [\![E'_i]\!]$.

On utilise ensuite les preuves Π'_l avec $l \neq i$ pour obtenir une preuve du séquent $T, \chi \vdash u \;\; [\![E']\!]$ en utilisant plusieurs fois la règle d'affaiblissement. On se reportera au cas de la règle d'affaiblissement pour les détails sur cette opération.

La conclusion est $C[u_l]$ et l'une de ses prémisses est $C[f(u_1, \ldots, u_m)]$ avec f un symbole de composition et C un contexte de profondeur 1 (sinon, on se ramène au cas où la conclusion est une variable de l'une des prémisses).

Deux cas sont alors possibles :

❶ $f(u_1, \ldots, u_m)$ n'est pas le résultat d'une instanciation à profondeur 1

❷ $f(u_1, \ldots, u_m)$ est le résultat d'une instanciation à profondeur 1

Dans le cas ❶, cela signifie que $f(u_1, \ldots, u_m)$ est sous-terme de v_{1i}. Dans ce cas, le terme u_l se trouvent dans le chapeau de preuve contenant uniquement des compositions et des règles d'affaiblissement et dont la conclusion est v_{1i}. On peut alors appliquer la règle de réécriture de la figure 10.22 page 117. Sur cette figure, la règle ① est en réalité une suite de règles de composition et d'affaiblissement, ce qui justifie la possibilité de l'appliquer sur u_l au lieu de $f(u_1, \ldots, u_m)$.

$$\dfrac{\dfrac{P_1}{T \vdash v_{n_1 1}\ [\![E_1]\!]} \quad \cdots \quad \dfrac{P_i}{T \vdash C[f(u_1,\dots,u,\dots,u_k)]\ [\![E_i]\!]} \quad \cdots \quad \dfrac{P_n}{T \vdash v_{n_n n}\ [\![E_n]\!]}}{T \vdash u\ [\![E_1 \wedge \dots \wedge E_n]\!]}\,\mathcal{D}$$

$$\Longrightarrow \dfrac{\dfrac{\begin{array}{c}\dfrac{\dfrac{\pi_i^j}{T\vdash u\ [\![E_i^j]\!]} \quad \dfrac{\pi_i^1}{T\vdash u_1\ [\![E_i^1]\!]}}{T\vdash u\ [\![E_i^j\wedge E_1^1]\!]}\,\mathcal{W}\\ \vdots \\ T\vdash u\ [\![E_i^j\wedge E_1^1\wedge\dots\wedge E_1^{k-1}]\!]\end{array} \quad \dfrac{\pi_i^k}{T\vdash u_k\ [\![E_i^k]\!]}}{T\vdash u\ [\![E_i]\!]}\,\mathcal{W} \quad \dfrac{P_1}{T\vdash v_{n_1 1}\ [\![E_1]\!]}}{\begin{array}{c}T\vdash u\ [\![E_1\wedge E_i]\!]\\ \vdots\,\mathcal{W}\\ T\vdash u\ [\![E_1\wedge\dots\wedge E_n]\!]\end{array}}\,\mathcal{W}$$

s'il existe une preuve

$$\dfrac{\dfrac{\pi_i^1}{T\vdash u_1\ [\![E_i^1]\!]} \quad \dots \quad \dfrac{\pi_i^j}{T\vdash u\ [\![E_i^j]\!]} \quad \dots \quad \dfrac{\pi_i^k}{T\vdash u_k\ [\![E_i^k]\!]}}{T\vdash f(u_1,\dots,u,\dots u_k)\ [\![E_i]\!]}\,\mathcal{C}$$

FIG. 10.21 - *Règle de réécriture pour obtenir une preuve de u*

$$\dfrac{\dfrac{P_1}{T \vdash v_{n_1 1}\ [\![E_1]\!]} \quad \cdots \quad \dfrac{\cdots \quad \dfrac{\dfrac{\pi_1}{T\vdash u_1\ [\![G_1]\!]}\ \ldots\ \dfrac{\pi_l}{T\vdash u_l\ [\![G_l]\!]}\ \ldots\ \dfrac{\pi_m}{T\vdash u_k\ [\![G_m]\!]}}{T\vdash f(u_1,\ldots,u_l,\ldots u_m)\ [\![G_1\wedge\ldots\wedge G_m]\!]}\mathcal{C} \quad \cdots}{T\vdash C[f(u_1,\ldots,u_l,\ldots u_m)]\ [\![E_i]\!]}\textcircled{1} \quad \dfrac{P_n}{T\vdash v_{n_n n}\ [\![E_n]\!]}}{T\vdash C[u_l]\ [\![E_1\wedge\ldots\wedge E_n]\!]}\mathcal{D}$$

$$\Longrightarrow \quad \begin{array}{c} \dfrac{\dfrac{\dfrac{\pi_l}{T\vdash u_l\ [\![G_l]\!]}}{T\vdash C[u_l]\ [\![G_l]\!]}\textcircled{1} \quad \dfrac{\pi_1}{T\vdash u_1\ [\![G_1]\!]}}{T\vdash C[u_l]\ [\![G_l\wedge G_1]\!]}\mathcal{W} \\ \vdots\,\mathcal{W} \\ T\vdash C[u_l]\ [\![G_1\wedge\ldots G_m]\!] \\ \vdots\,\mathcal{W} \\ \dfrac{T\vdash C[u_l]\ [\![E_i]\!] \qquad \cdots\ \dfrac{P_1}{T\vdash v_{n_1 1}\ [\![E_1]\!]}}{T\vdash C[u_l]\ [\![E_1\wedge E_i]\!]}\mathcal{W} \\ \vdots\,\mathcal{W} \\ T\vdash C[u_l]\ [\![E_1\wedge\ldots\wedge E_n]\!]\end{array}$$

FIG. 10.22 - *Règle de réécriture pour obtenir une preuve de* $C[u_l]$

Comme pour le cas où la conclusion est une variable d'une des prémisses, on applique l'hypothèse de récurrence sur les preuves $\pi_1, \ldots, \pi_m, \ldots, P_1, \ldots, P_n$ (qui constituent les feuilles dans le membre droit de la règle de réécriture) et on applique les mêmes règles pour obtenir une preuve de $T, \chi \vdash C[u_l]\ \llbracket E' \rrbracket$.

Dans le cas ❷, on voudrait remplacer $f(u_1, \ldots, u_m)$ par une preuve de u_l et ainsi obtenir une preuve de $C[u_l]$. On remplace dans la contrainte $f(u_1, \ldots, u_m)$ (dans la preuve originale) par u_l et on applique de nouveau l'hypothèse de récurrence. Ce processus termine car la taille de la contrainte diminue de manière stricte (et c'est le seul endroit de la preuve où l'on procède ainsi). Plus précisément, le terme $f(u_1, \ldots, u_m)$ a été inséré dans la contrainte lors d'une règle de protocole. Ce terme n'est pas égal, modulo E'_f à un terme de $F(V, T, \Sigma, \xi)$, il est donc le résultat de compositions et on peut effectuer son remplacement par u_l puis ajuster les contraintes avec des règles d'affaiblissement.

La figure 10.23 page ci-contre présente la règle de réécriture à appliquer. La règle ② est en fait une combinaison de règles de composition et d'affaiblissement. On considère que $G_1 \wedge \ldots \wedge G_m \subset G \subset G' \subset G'' = E_i$.

Il reste cependant à vérifier que la preuve peut toujours être construite malgré ce remplacement. Le seul cas problématique réside dans les règles de décomposition qui sont précédées d'une instanciation profonde de ce terme. La règle de décomposition que l'on examine actuellement n'est pas concernée puisqu'on a réussi à construire une preuve de la conclusion en effectuant ce remplacement. Aucune autre règle de décomposition n'est concernée car, parmi les restrictions imposées sur les règles de décomposition, à profondeur 1, le symbole f ne peut être utilisé que dans la prémisse de la règle que l'on vient de considérer. C'est la troisième hypothèse de la section 10.4 page 94.

Aucun de ces deux cas . Dans ce cas, chaque prémisse est à profondeur 1 et il n'y a pas d'instanciation profondes qui n'ait pas été repoussée par le lemme 8.3 page 68. Ce cas n'est pas possible (on a $u = u'$).

Règle d'instanciation . Nous avons simplement besoin de couvrir le cas où l'instanciation est appliquée avec un contexte vide. La preuve que nous considérons est donc de la forme :

$$\dfrac{\dfrac{\Pi_1}{T \vdash x\ \llbracket E \wedge x = u \rrbracket}}{T \vdash u\ \llbracket E \wedge x = u \rrbracket}\,\mathcal{I}$$

La preuve Π_1 se termine ou bien par une instanciation dont le contexte est vide, ou bien par une règle qui n'est pas une instanciation. On peut donc y appliquer l'hypothèse de récurrence. Nous obtenons un ensemble de contraintes F_χ de χ qui vérifie directement le premier point de l'hypothèse de récurrence. Le second point est également vérifié car aucune nouvelle variable n'a été introduite dans F_χ.

Soit θ_χ une substitution modèle de F_χ. Il existe donc une preuve Π_{θ_χ} obtenue via l'hypothèse de récurrence. Deux cas peuvent alors se présenter. Ou bien la contrainte de Π_{θ_χ} est de la forme $E'\theta_\chi \wedge x = u$, ou bien elle est de la forme $E'\theta_\chi \wedge x = \chi^x_u\theta_\chi$.

Dans le premier cas, il nous suffit d'appliquer la règle d'instanciation pour obtenir une preuve de $T, \chi\theta_\chi \vdash u\ \llbracket E'\theta_\chi \wedge x = u \rrbracket$. L'admissibilité des séquents de cette preuve pro-

$$\dfrac{\dfrac{P_1}{T \vdash v_{n_1 1}\ \llbracket E_1 \rrbracket} \quad \cdots \quad \dfrac{\begin{array}{c}\dfrac{\dfrac{\cdots \quad \dfrac{\dfrac{\pi_1}{T \vdash u_1\ \llbracket G_1 \rrbracket}\ \ldots\ \dfrac{\pi_l}{T \vdash u_l\ \llbracket G_l \rrbracket}\ \ldots\ \dfrac{\pi_m}{T \vdash u_k\ \llbracket G_m \rrbracket}}{T \vdash f(u_1, \ldots, u_l, \ldots u_m)\ \llbracket G_1 \wedge \ldots \wedge G_m \rrbracket}\,\mathcal{C} \quad \cdots}{T \vdash C'[f(u_1, \ldots, u_l, \ldots u_m)]\ \llbracket G \rrbracket}\,\textcircled{2}}{T \vdash C''[z]\ \llbracket G' \wedge z = f(u_1, \ldots, u_m) \rrbracket}\,\mathcal{P} \\ \vdots \\ T \vdash C[z]\ \llbracket G'' \wedge z = f(u_1, \ldots, u_m) \rrbracket\end{array}}{T \vdash C[f(u_1, \ldots, u_m)]\ \llbracket G'' \wedge z = f(u_1, \ldots, u_m) \rrbracket}\,\mathcal{I} \quad \cdots \quad \dfrac{P_n}{T \vdash v_{n_n n}\ \llbracket E_n \rrbracket}}{T \vdash C[u_l]\ \llbracket E_1 \wedge E_{i-1} \wedge G'' \wedge z = f(u_1, \ldots, u_m) \wedge E_{i+1} \wedge \ldots \wedge E_n \rrbracket}\,\mathcal{D}$$

$$\Longrightarrow \quad \begin{array}{c} \dfrac{\dfrac{\begin{array}{c}\dfrac{\dfrac{\pi_l}{T \vdash u_l\ \llbracket G_l \rrbracket} \quad \dfrac{\pi_1}{T \vdash u_1\ \llbracket G_1 \rrbracket}}{T \vdash u_l\ \llbracket G_l \wedge G_1 \rrbracket}\,\mathcal{W} \\ \vdots\,\mathcal{W} \\ \cdots \quad T \vdash u_l\ \llbracket G_1 \wedge \ldots \wedge G_m \rrbracket \quad \cdots \end{array}}{T \vdash C'[u_l]\ \llbracket G \rrbracket}\,\textcircled{2}}{T \vdash C''[z]\ \llbracket G' \wedge z = u_l \rrbracket}\,\mathcal{P} \\ \vdots \\ \dfrac{\dfrac{T \vdash C[z]\ \llbracket G'' \wedge z = u_l \rrbracket}{T \vdash C[u_l]\ \llbracket G'' \wedge z = u_l \rrbracket}\,\mathcal{I} \qquad \dfrac{P_1}{T \vdash v_{n_1 1}\ \llbracket E_1 \rrbracket}}{T \vdash C[u_l]\ \llbracket G'' \wedge z = u_l \wedge E_1 \rrbracket}\,\mathcal{W} \\ \vdots\,\mathcal{W} \\ T \vdash C[u_l]\ \llbracket E_1 \wedge E_{i-1} \wedge G'' \wedge z = u_l \wedge E_{i+1} \wedge \ldots \wedge E_n \rrbracket \end{array}$$

FIG. 10.23 - *Seconde règle de réécriture pour obtenir une preuve de* $C[u_l]$

vient alors directement de l'hypothèse de récurrence en ce qui concerne les séquents autres que le séquent final et du fait que le séquent $T, \chi\theta_\chi \vdash u\ [\![E'\theta_\chi \wedge x = u]\!]$ est la composition de la contrainte du séquent $T, \chi\theta_\chi \vdash x\ [\![E'\theta_\chi \wedge x = u]\!]$ et du terme u. Le séquent $T, \chi\theta_\chi \vdash u\ [\![E'\theta_\chi \wedge x = u]\!]$ est $F(V, T, \Sigma, \xi, \chi\theta_\chi), E'_f$-admissible via l'hypothèse de récurrence et donc u est égal, modulo E'_f, à un terme qui est dans $F(V, T, \Sigma, \xi, \chi\theta_\chi)$. Le séquent $T, \chi\theta_\chi \vdash u\ [\![E'\theta_\chi \wedge x = u]\!]$ est donc également $F(V, T, \Sigma, \xi, \chi\theta_\chi), E'_f$-admissible.

Dans le second cas, l'hypothèse de récurrence nous fournit une preuve du séquent $T \vdash u\ [\![E \wedge x = u]\!]$. On peut y appliquer l'hypothèse de récurrence car cette preuve se termine par une règle de composition. On obtient un ensemble de contraintes F'_χ tel que $F_\chi \models F'_\chi$ et donc $\theta_\chi \models F'_\chi$. On obtient donc une preuve du séquent contraint $T\, \chi\theta_\chi \vdash u\ [\![E'\theta_\chi \wedge x = u'\theta_\chi]\!]$ vérifiant les propriétés voulues.

Règle de divulgation de nonce . On considère la preuve suivante :

$$\Pi = \cfrac{\cfrac{\Pi_1}{T \vdash u\ [\![E]\!]}}{T \vdash N_i(s)\ [\![E]\!]}\,\mathcal{ND}$$

La preuve Π_1 ne termine pas par une règle d'instanciation en raison de l'application du 8.3 page 68. Il est donc possible d'utiliser l'hypothèse de récurrence. Nous obtenons alors un ensemble de contraintes F_χ de χ qui vérifie déjà le premier point de l'hypothèse de récurrence. Pour le second point, le raisonnement est similaire au cas de la règle de protocole. Reste à vérifier le troisième point.

Soit $\theta_\chi \models F_\chi$. L'hypothèse de récurrence nous fournit alors une preuve du séquent $T, \chi\theta_\chi \vdash u\ [\![E'\theta_\chi]\!]$. Soit π_1 la sous-preuve maximale ne contenant que des règles d'affaiblissement et des règles de composition. On y applique le lemme 10.1 page 83 pour obtenir la preuve π_2. La règle de divulgation de nonce peut toujours être appliquée car les conditions d'application n'ont pas changé (le terme de la conclusion de π_2 n'a pas d'importance). On obtient alors la preuve Π_{θ_χ} de $T, \chi\theta_\chi \vdash N_i(s)\ [\![E'\theta_\chi]\!]$.

Soit $T, \chi\theta_\chi \vdash t\ [\![c\theta_\chi]\!]$ un séquent de Π_{θ_χ}. Si celui-ci est en dehors de π_2 et n'est pas le séquent final, l'hypothèse de récurrence permet de conclure. S'il est dans π_2, il est la composition du terme d'un des séquents $F(V, T, \Sigma, \xi, \chi\theta_\chi), E'_f$-admissible utilisé en tant qu'hypothèse de π_2 et de la combinaison des contraintes de certains séquents également $F(V, T, \Sigma, \xi, \chi\theta_\chi), E'_f$-admissible et hypothèses de π_2. Le séquent qui nous intéresse est donc aussi $F(V, T, \Sigma, \xi, \chi\theta_\chi), E'_f$-admissible.

Enfin, si on considère le séquent conclusion de Π_{θ_χ}, il s'agit de la combinaison de la contrainte $E'\theta_\chi$ du séquent $T, \chi\theta_\chi \vdash u'\ [\![E'\theta_\chi]\!]$, $F(V, T, \Sigma, \xi, \chi\theta_\chi), E'_f$-admissible comme on vient de le voir et du terme $N_i(s)$ qui appartient à $F(V, T, \Sigma, \xi, \chi\theta_\chi)$. Il est donc aussi $F(V, T, \Sigma, \xi, \chi\theta_\chi), E'_f$-admissible.

10.7 Application au camouflage

Nous allons présenter ici une application possible de ce théorème au principe du camouflage, encore appelé *signatures en aveugle* [Cha82]. Elles est utilisée dans des protocoles de vote électronique tels que FOO92 [FOO93, KR05]. Elle consiste à donner la possibilité de signer un message sans connaître le message en question. Ce dernier est préalablement camouflé à l'aide d'un nombre r. Avec la connaissance de ce nombre, il est possible de retirer le camouflage.

Une analogie avec la vie réelle consiste à voir le camouflage comme la mise sous enveloppe d'un message accompagné d'une feuille de papier carbone [Sch93]. Il est alors possible de signer l'enveloppe sans voir le message contenu à l'intérieur. La feuille de papier carbone effectue le transfert en aveugle de la signature.

10.7.1 Règles de déduction de l'intrus

Nous introduisons dans un premier temps les règles de déduction de l'intrus. Celles-ci sont inspirées de [KR05] qui utilise une théorie équationnelle. Il serait possible d'appliquer les variants finis à cette théorie équationnelle, mais nous allons donner directement un système de déduction sans théorie équationnelle. La figure 10.24 page suivante donne les règles de $\overline{S}$ concernant le camouflage. Ces règles seront complétées par les règles d'instanciation, de divulgation de nonce, d'affaiblissement et d'ouverture et d'avancement de sessions telles que décrites dans le chapitre 6. Il faut de plus y rajouter la règle d'axiome.

10.7.2 Localité

Afin d'appliquer le théorème, nous devons dans un premier temps trouver une fonction F telle que la propriété de localité 9.2 page 74 soit vérifiée.

Lemme 10.3 *La propriété de localité 9.2 page 74 avec F la plus petite fonction telle que :*

1. *F est close par sous-terme.*
2. *Si $\{m\}_{sk}^{-1} \in F(T)$ alors $pub(sk) \in F(T)$.*
3. *Si $\{camoufle(m,r)\}_{sk}^{-1} \in F(T)$ alors $\{m\}_{sk}^{-1} \in F(T)$.*

est vérifiée pour les règles de déduction de l'intrus présentées dans la figure 10.24 page suivante.

Preuve. La première propriété de F nous assure que $T \subseteq F(T)$. Elle nous permet également d'obtenir l'inclusion $F(T) \subseteq F(F(T))$. Étant donné que cette fonction est définie par une propriété de clôture, nous avons également $F(F(T)) \subseteq F(T)$ ce qui nous permet d'obtenir l'égalité $F(T) = F(F(T))$. Il nous reste alors à montrer que la propriété de localité en elle-même est vérifiée.

Soit Π une preuve de $T \vdash s$ dans le système défini dans la figure 10.24 (sans les contraintes). Nous procédons par récurrence sur la taille de Π. Le cas de base est constitué par la règle d'axiome et est traité de manière triviale. Nous considérons la dernière règle de Π.

Règle de chiffrement

$$\frac{\dfrac{\Pi_1}{T \vdash m} \qquad \dfrac{\Pi_2}{T \vdash r}}{T \vdash \{m\}_r}$$

Il s'agit d'une règle de composition. On applique l'hypothèse de récurrence sur les preuves de Π_1 et Π_2 pour obtenir respectivement Π'_1 et Π'_2. On sait alors que tous les termes qui apparaissent dans Π'_1 et Π'_2 sont dans $F(T,m)$, $F(T,r)$ ou $F(T)$. Ils sont

Chiffrement

$$\frac{T \vdash m \;\; [\![E_1]\!] \quad T \vdash r \;\; [\![E_2]\!]}{T \vdash \{m\}_r \;\; [\![E_1 \wedge E_2]\!]}$$

Signature

$$\frac{T \vdash m \;\; [\![E_1]\!] \quad T \vdash sk \;\; [\![E_2]\!]}{T \vdash \{m\}_{sk}^{-1} \;\; [\![E_1 \wedge E_2]\!]}$$

Camouflage

$$\frac{T \vdash m \;\; [\![E_1]\!] \quad T \vdash r \;\; [\![E_2]\!]}{T \vdash \mathrm{camoufle}(m, r) \;\; [\![E_1 \wedge E_2]\!]}$$

Clef publique

$$\frac{T \vdash sk \;\; [\![E_1]\!]}{T \vdash \mathrm{pub}(sk) \;\; [\![E_1]\!]}$$

Déchiffrement

$$\frac{T \vdash \{m\}_r \;\; [\![E_1]\!] \quad T \vdash r' \;\; [\![E_2]\!]}{T \vdash m \;\; [\![E_1 \wedge E_2]\!]} \text{ si } r = r'$$

Vérification de signature

$$\frac{T \vdash \{m\}_{sk}^{-1} \;\; [\![E_1]\!] \quad T \vdash \mathrm{pub}(sk') \;\; [\![E_2]\!]}{T \vdash m \;\; [\![E_1 \wedge E_2]\!]} \text{ si } sk = sk'$$

Signature camouflée

$$\frac{T \vdash \{\mathrm{camoufle}(m, r)\}_{sk}^{-1} \;\; [\![E_1]\!] \quad T \vdash r' \;\; [\![E_2]\!]}{T \vdash \{m\}_{sk}^{-1} \;\; [\![E_1 \wedge E_2]\!]} \text{ si } r = r'$$

FIG. 10.24 - *Règles de déduction de l'intrus dans le cas du camouflage*

donc également dans $F(T, \{m\}_r)$ par clôture de F par sous-terme. Nous obtenons Π' respectant l'hypothèse de récurrence en appliquant la règle de chiffrement à Π'_1 et Π'_2.

Règle de signature

$$\frac{\dfrac{\Pi_1}{T \vdash m} \qquad \dfrac{\Pi_2}{T \vdash sk}}{T \vdash \{m\}_{sk}^{-1}}$$

Il s'agit aussi d'une règle de décomposition. Le raisonnement est strictement identique que pour le chiffrement. Cela nous permet d'obtenir une preuve où tous les termes sont dans $F(V, \{m\}_{sk}^{-1})$.

Règle de camouflage

$$\frac{\dfrac{\Pi_1}{T \vdash m} \qquad \dfrac{\Pi_2}{T \vdash r}}{T \vdash \mathrm{camoufle}(m, r)}$$

Encore une fois, le raisonnement est strictement identique. C'est également une règle de composition.

Règle de clef publique

$$\frac{\dfrac{\Pi_1}{T \vdash sk}}{T \vdash \mathrm{pub}(sk)}$$

Nous sommes en présence d'une règle de composition et nous appliquons le même raisonnement que précédemment.

Règle de déchiffrement

$$\frac{\dfrac{\Pi_1}{T \vdash \{m\}_r} \qquad \dfrac{\Pi_2}{T \vdash r}}{T \vdash m}$$

Il s'agit d'une règle de décomposition. On applique l'hypothèse de récurrence sur Π_1 (respectivement Π_2) pour obtenir une preuve Π'_1 (respectivement Π'_2). Deux cas peuvent se présenter :

- Π'_1 se termine par une décomposition. Dans ce cas, on construit la preuve Π' en appliquant la règle de déchiffrement sur Π'_1 et Π'_2. Comme Π'_1 se termine par un déchiffrement, $\{m\}_r \in F(T)$. Par clôture de F par sous-terme, on a aussi $m \in F(T)$ et $r \in F(T)$. Par idempotence de F, on a alors $F(T, r) \subseteq F(T)$. Tous les termes sont donc bien dans $F(T)$.
- Π'_1 se termine par une composition. Dans ce cas, il ne peut s'agir que du chiffrement. On a donc en fait une preuve se présentant ainsi :

$$\frac{\dfrac{\dfrac{\Pi'_{11}}{T \vdash m} \qquad \dfrac{\Pi'_{12}}{T \vdash r}}{T \vdash \{m\}_r} \qquad \dfrac{\Pi'_2}{T \vdash r}}{T \vdash m}$$

On considère la preuve Π'_{11} de $T \vdash m$ comme notre nouvelle preuve respectant l'hypothèse de récurrence.

Règle de vérification de signature

$$\dfrac{\dfrac{\Pi_1}{T \vdash \{m\}_{sk}^{-1}} \qquad \dfrac{\Pi_2}{T \vdash \mathrm{pub}(sk)}}{T \vdash m}$$

Il s'agit d'une règle de décomposition. On applique l'hypothèse de récurrence sur Π_1 et Π_2 pour obtenir des preuves Π'_1 et Π'_2. De nouveau, deux cas se présentent :

- Π'_1 se termine par une décomposition. On construit Π' en appliquant la règle de vérification de signature sur les preuves Π'_1 et Π'_2. Comme Π'_1 se termine par une décomposition, $\{m\}_{sk}^{-1} \in F(T)$ et par clôture de F, $m \in F(T)$. On utilise alors la seconde propriété de clôture de F pour conclure que comme $\{m\}_{sk}^{-1} \in F(T)$, $\mathrm{pub}(sk) \in F(T)$. Comme précédemment, on peut alors conclure que tous les termes de Π' sont dans $F(T)$.
- Π'_1 se termine par une composition. La dernière règle de Π'_1 est donc la règle de signature. On procède comme dans le cas de la règle de déchiffrement pour trouver une preuve plus courte de $T \vdash m$ respectant l'hypothèse de récurrence.

Règle de signature camouflée

$$\dfrac{\dfrac{\Pi_1}{T \vdash \{\mathrm{camoufle}(m,r)\}_{sk}^{-1}} \qquad \dfrac{\Pi_2}{T \vdash r}}{T \vdash \{m\}_{sk}^{-1}}$$

Il s'agit d'une règle de décomposition. On applique l'hypothèse de récurrence sur Π_1 et Π_2 pour obtenir des preuves Π'_1 et Π'_2. Deux cas se présentent alors :

- Π'_1 s'achève par une règle de décomposition. On construit Π' en appliquant la règle de signature camouflée sur Π'_1 et Π'_2. On a $\{\mathrm{camoufle}(m,r)\}_{sk}^{-1} \in F(T)$ et donc $r \in F(T)$. Cela implique que $F(T,r) \subseteq F(T)$. À part la conclusion, tous les termes de Π' sont donc dans $F(T)$. Grâce à la dernière propriété de clôture de F, $\{m\}_{sk}^{-1}$ est également dans $F(T)$, ce qui permet de conclure.
- Π'_1 se termine par une composition. Sa dernière règle est donc une règle de signature :

$$\dfrac{\dfrac{\dfrac{\Pi'_{11}}{T \vdash \mathrm{camoufle}(m,r)} \qquad \dfrac{\Pi'_{12}}{T \vdash sk}}{T \vdash \{\mathrm{camoufle}(m,r)\}_{sk}^{-1}} \qquad \dfrac{\Pi'_2}{T \vdash r}}{T \vdash \{m\}_{sk}^{-1}}$$

Il est possible de construire la preuve suivante :

$$\dfrac{\dfrac{\dfrac{\Pi'_{11}}{T \vdash \mathrm{camoufle}(m,r)} \qquad \dfrac{\Pi'_2}{T \vdash r}}{T \vdash m} \qquad \dfrac{\Pi'_{12}}{T \vdash sk}}{T \vdash \{m\}_{sk}^{-1}}$$

Cette preuve se termine par une composition. $sk \in F(T, \{m\}_{sk}^{-1})$ car F est close par sous-terme donc tous les termes dans Π'_{12} sont dans $F(T, \{m\}_{sk}^{-1})$. Pour la même raison, on a bien $m \in F(T, \{m\}_{sk}^{-1})$. Deux nouveaux cas se présentent :

1. Π'_{11} se termine par une règle de décomposition et donc $\mathrm{camoufle}(m,r) \in F(T)$ et, par clôture par sous-terme, $r \in F(T)$. L'hypothèse de récurrence nous indique alors que tous les termes dans Π'_{11} sont dans $F(T)$. Tous les termes dans Π'_2 sont dans $F(T)$ ou $F(T,r) \subseteq F(T)$.
2. Pi'_{11} se termine par une règle de composition, c'est à dire que l'on a en fait la preuve suivante :

$$\dfrac{\dfrac{\dfrac{\dfrac{\Pi'_{111}}{T \vdash m} \quad \dfrac{\Pi'_{112}}{T \vdash r}}{T \vdash \mathrm{camoufle}(m,r)} \quad \dfrac{\Pi'_2}{T \vdash r}}{T \vdash m} \quad \dfrac{\Pi'_{12}}{T \vdash sk}}{T \vdash \{m\}_{sk}^{-1}}$$

On construit alors la preuve suivante :

$$\dfrac{\dfrac{\Pi'_{111}}{T \vdash m} \quad \dfrac{\Pi'_{12}}{T \vdash sk}}{T \vdash \{m\}_{sk}^{-1}}$$

Cette preuve se termine par une règle de composition. Tous ces termes doivent donc être dans $F(T, \{m\}_{sk}^{-1})$. C'est le cas de $\{m\}_{sk}^{-1}$. Par hypothèse de récurrence tous les termes de Π_{111} (respectivement Π'_{12}) sont dans $F(T,m)$ (respectivement $F(T,sk)$) ou $F(T)$. Mais comme, par clôture par sous-terme, $m \in F(T)$ (respectivement $sk \in F(T)$), $F(T,m) \subseteq F(T)$ (respectivement $F(T,sk) \subseteq F(T)$). Cette preuve vérifie donc la propriété de localité.

□

10.7.3 Restrictions syntaxiques

Nous devons vérifier que le système de la figure 10.24 vérifie les propriétés syntaxiques sur les règles de composition et de décomposition présentées dans les section 9.3.1 et 9.3.2 page 76.

Les règles de composition sont les règles de chiffrement, de signature, de camouflage et de clef publique. Il s'agit bien de règles se contentant d'effectuer un ajout de symbole en tête des termes : $\{\cdot\}_{\cdot}$, $\{\cdot\}_{\cdot}^{-1}$, $\mathrm{camoufle}(\cdot,\cdot)$ et $\mathrm{pub}(\cdot)$.

La règle de déchiffrement n'a que des hypothèses à profondeur 0 ou 1. La règle de vérification de signature également. Enfin, la règle de signature camouflée a sa première hypothèse de profondeur 2. Cette hypothèse, $\{\mathrm{camoufle}(m,r)\}_{sk}^{-1}$, peut être réécrite sous la forme $C[\mathrm{camoufle}(m,r)]$ avec $C = \{\cdot\}_{sk}^{-1}$. La conclusion est alors $C[m]$. Cette règle est donc bien une règle de décomposition vérifiant les restrictions syntaxiques imposées. On notera que $\mathrm{camoufle}(\cdot,\cdot)$ est bien un symbole de composition.

Enfin, nous devons également vérifier que les restrictions supplémentaires introduites dans la section 10.4 page 94 de ce chapitre sont également vérifiées. Les deux premiers points ne s'appliquent pas ici. Dans la règle de déchiffrement, r et r' sont à profondeur 1 et 0. Dans la règle de vérification de signature, sk et sk' sont à profondeur 1 et 1. Dans la règle de signature camouflée, r et r' sont à profondeur 2 et 0. En ce qui concerne la dernière condition, la fonction $\mathrm{camoufle}(\cdot,\cdot)$ n'apparaît que dans la dernière règle de décomposition. La dernière condition est donc également vérifiée.

10.7.4 Application du théorème principal

Nous obtenons alors le théorème suivant :

Théorème 10.2 *Soit Π une preuve de $T \vdash s \ \llbracket E \rrbracket$ utilisant les règles de la figure 10.24 page 122. Soit F la plus petite fonction telle que :*

1. *F est close par sous-terme.*
2. *Si $\{m\}_{sk}^{-1} \in F(T)$ alors $pub(sk) \in F(T)$.*
3. *Si $\{camoufle(m,r)\}_{sk}^{-1} \in F(T)$ alors $\{m\}_{sk}^{-1} \in F(T)$.*

Si E est un ensemble d'équations satisfaisable et sous forme résolue, utilisant l'ensemble des règles de protocole V dont Σ est l'ensemble des affectations initiales, alors il existe une preuve Π' de $T \vdash s' \ \llbracket E' \rrbracket$ dans $\overline{S}$ telle que $s'\sigma_{E'} = s\sigma_{E'}$ et telle que tout séquent $T \vdash t \ \llbracket E'' \rrbracket$ dans Π' soit $F(V,T,\Sigma,s,\xi), E'$-admissible.

Troisième partie

Algorithmique de recherche de preuves

CHAPITRE 11

Nombre borné de sessions

Sommaire

Le théorème du chapitre 10 nous permet de restreindre les termes possibles dans les séquents contraints à un ensemble fini de termes. La taille de cet ensemble est paramétré par la fonction F. Ce résultat va nous permettre d'obtenir une procédure de décision sur la présence d'une attaque sur un protocole donné. Cet algorithme est présenté dans la section 11.2 page 133. Cette procédure est complète et correcte. Dans le cas d'un nombre borné de sessions, il termine et constitue alors un résultat de décision.

Dans le cas d'un nombre non borné de sessions, la procèdure correcte et complète consiste à rechercher une attaque en 0 session, 1 session, 2 sessions, etc. Une procédure plus réaliste sera cependant proposée dans le chapitre 12 page 137 qui fournira une procédure déterministe, correct et complet et qui s'avère plus efficace que l'implémentation simple de la procédure de la section 11.2 de ce chapitre. Elle terminera également pour un nombre borné de sessions. De plus, elle ne permet pas simplement d'obtenir l'attaque trouvée. Ce défaut sera également corrigé dans le chapitre 12.

Nous supposons dans ce chapitre et dans le suivant que nous avons un système de preuves $\overline{\mathcal{S}}$ vérifiant les conditions nécessaires, énoncées dans la partie II, pour appliquer le théorème 10.1 page 82.

11.1 Définition du système $\overline{\mathcal{S}}'$

Dans un arbre de preuve de $\overline{\mathcal{S}}$, une fois la contrainte finale déterminée, les contraintes intermédiaires sont non seulement limitées à un ensemble calculé par la fonction F, mais il n'y a en fait aucun degré de liberté dans leur choix une fois le squelette de l'arbre de preuve décidé.

Nous allons définir un système $\overline{\mathcal{S}}'$ dans lequel toutes les contraintes d'un arbre de preuve sont identiques, à l'exception des points de contrôle. Nous verrons qu'un tel système est équivalent, vis-à-vis de la recherche d'attaque, à $\overline{\mathcal{S}}$. Nous pourrons ainsi travailler dans $\overline{\mathcal{S}}'$ pour notre algorithme de recherche d'attaque.

Définition 11.1 *$\overline{\mathcal{S}}'$ est défini en enrichissant $\overline{\mathcal{S}}$ de la règle d'axiome suivante :*

$$\frac{}{T \vdash t \ \ [\![E]\!]}\ \mathcal{A}$$

avec $t \in T$ et E une contrainte arbitraire dans laquelle aucune variable de ξ n'apparaît en membre gauche (pas de point de contrôle).

On impose de plus la condition suivante : pour tout couple de séquents $T \vdash t_1 \ [\![E_1]\!]$ et $T \vdash t_2 \ [\![E_2]\!]$, si on note E_1' (respectivement E_2') la contrainte E_1 (respectivement E_2) à laquelle on a retiré les équations où une variable de ξ apparaissait en membre gauche, $E_1' = E_2'$ (égalité syntaxique).

Nous montrons un résultat plus général que la simple équivalence de recherche d'attaque entre $\overline{\mathcal{S}}$ et $\overline{\mathcal{S}}'$:

Lemme 11.1 *Pour toute preuve Π de $T \vdash s \ [\![E]\!]$ dans $\overline{\mathcal{S}}$ (respectivement $\overline{\mathcal{S}}'$), il existe une preuve Π' de $T \vdash s \ [\![E]\!]$ dans $\overline{\mathcal{S}}'$ (respectivement $T \vdash s \ [\![F]\!]$ dans $\overline{\mathcal{S}}$ avec $F \subset E$) telle que $\Pi = \sigma(\Pi')$ où σ est un opérateur qui transforme une contrainte E en une contrainte E' telle que $E' \subset E$ (respectivement $E \subset E'$).*

À titre d'exemple de l'application d'un tel lemme, la preuve de $\overline{\mathcal{S}}$ de la figure 4.3 page 43 se trouve transformée par le lemme 11.1 en la preuve de $\overline{\mathcal{S}}'$ présentée sur la figure 11.1 page suivante.

On utilise les notations : $T = A, B, I, pub(I)^{-1}$ et $\sigma_{s_{123}}$ la contrainte $a_1 = A \wedge b_1 = I \wedge n_{a_1} = N_1(s_1) \wedge c_2 = B \wedge n_{c_2} = N_1(s_2) \wedge y_2 = a_1 \wedge z_2 = n_{a_1} \oplus B \oplus I \wedge x_1 = n_{c_2}$.

Preuve. Le passage d'une preuve de $\overline{\mathcal{S}}$ en une preuve de $\overline{\mathcal{S}}'$ est simple : on considère la contrainte finale de la preuve à transformer, on en supprime les équations dont le membre gauche est une variable de χ et pour chacun des séquents, on effectue l'union de la contrainte obtenue avec la contrainte du séquent.

Soit Π une preuve de $\overline{\mathcal{S}}'$. On applique les règles de réécriture de la figure 11.1 page 132 pour obtenir la preuve Π'.

Les règles de réécriture conservent l'invariant que toute contrainte E est transformée en une contrainte F telle que $F \models E$. Toutes les règles transformées par les règles de réécriture sont des règles valides de $\overline{\mathcal{S}}$. En ce qui concerne les règles de protocole et de divulgation de nonces, les conditions d'application sont les mêmes dans $\overline{\mathcal{S}}$ et dans $\overline{\mathcal{S}}'$, ce qui explique la validité des règles de réécriture concernées. Enfin, une variable $x \in V$ ne peut se trouver en tant que terme d'un séquent uniquement s'il a été précédemment introduit par une règle de protocole. Il apparaît alors en tant que membre gauche de la contrainte. Cela justifie le fait que la dernière règle de réécriture concerne en réalité l'ensemble des règles d'instanciation.

□

On notera que la transformation n'est pas symétrique : si Π est une preuve de $\overline{\mathcal{S}}'$, la transformer en une preuve de $\overline{\mathcal{S}}$ puis de nouveau en une preuve de $\overline{\mathcal{S}}'$ permet d'obtenir

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{T \vdash \{\langle a_1, n_{a_1}\rangle\}_{\mathrm{pub}(b_1)} \ \llbracket \sigma_{s_{123}} \wedge x_{\mathrm{A},1,s_1} = 1 \rrbracket}\,\mathcal{P} \qquad \dfrac{}{T \vdash \mathrm{pub}(I)^{-1} \ \llbracket\rrbracket}\,\mathcal{A}}{T \vdash \langle a_1, n_{a_1}\rangle \ \llbracket \sigma_{s_{123}} \wedge x_{\mathrm{A},1,s_1} = 1 \rrbracket}\,\mathcal{D}}{T \vdash n_{a_1} \ \llbracket \sigma_{s_{123}} \wedge x_{\mathrm{A},1,s_1} = 1 \rrbracket}\,\mathcal{D} \qquad \dfrac{}{T \vdash B \ \llbracket\rrbracket}\,\mathcal{A} \qquad \dfrac{}{T \vdash I \ \llbracket\rrbracket}\,\mathcal{A}}{T \vdash n_{a_1} \oplus B \oplus I \ \llbracket \sigma_{s_{123}} \wedge x_{\mathrm{A},1,s_1} = 1 \rrbracket}\,\mathcal{C}$$

$$\dfrac{\dfrac{\dfrac{}{T \vdash A \ \llbracket\rrbracket}\,\mathcal{A} \qquad T \vdash n_{a_1} \oplus B \oplus I \ \llbracket \sigma_{s_{123}} \wedge x_{\mathrm{A},1,s_1} = 1 \rrbracket}{T \vdash \langle A, n_{a_1} \oplus B \oplus I\rangle \ \llbracket \sigma_{s_{123}} \wedge x_{\mathrm{A},1,s_1} = 1 \rrbracket}\,\mathcal{C} \qquad \cdots \ \llbracket\rrbracket}{\dfrac{T \vdash \{\langle A, n_{a_1} \oplus B \oplus I\rangle\}_{\mathrm{pub}(B)} \ \llbracket \sigma_{s_{123}} \wedge x_{\mathrm{A},1,s_1} = 1 \rrbracket}{\dfrac{T \vdash \{\langle z_2 \oplus c_2, n_{c_2}\rangle\}_{\mathrm{pub}(y_2)} \ \llbracket \sigma_{s_{123}} \wedge x_{\mathrm{A},1,s_1} = 1 \wedge x_{\mathrm{B},1,s_2} = 1 \rrbracket}{T \vdash \{x_1\}_{\mathrm{pub}(b_1)} \ \llbracket \sigma_{s_{123}} \wedge x_{\mathrm{A},1,s_1} = 1 \wedge x_{\mathrm{B},1,s_2} = 1 \wedge x_{\mathrm{A},2,s_1} = 1 \rrbracket}\,\mathcal{P}}\,\mathcal{P}}\,\mathcal{C}$$

TAB. 11.1 - *Première partie de la preuve d'attaque de NS avec* $\oplus$ *dans* $\overline{\mathcal{S}}'$

1.
$$\frac{}{T \vdash t \ \llbracket E \rrbracket}\mathcal{A} \Rightarrow \frac{}{T \vdash t \ \llbracket \rrbracket}\mathcal{A}$$

2.
$$\frac{T \vdash u_1 \ \llbracket E_1 \rrbracket \quad \dots \quad T \vdash u_n \ \llbracket E_n \rrbracket}{T \vdash u \ \llbracket E \rrbracket}\mathcal{S} \Rightarrow \frac{T \vdash u_1 \ \llbracket E_1 \rrbracket \quad \dots \quad T \vdash u_n \ \llbracket E_n \rrbracket}{T \vdash u \ \llbracket E_1 \wedge \dots \wedge E_n \rrbracket}\mathcal{S}$$

3.
$$\frac{T \vdash u_1 \ \llbracket E_1 \rrbracket \quad T \vdash u_2 \ \llbracket E_2 \rrbracket}{T \vdash u_1 \ \llbracket E \rrbracket}\mathcal{W} \Rightarrow \frac{T \vdash u_1 \ \llbracket E_1 \rrbracket \quad T \vdash u_2 \ \llbracket E_2 \rrbracket}{T \vdash u_1 \ \llbracket E_1 \wedge E_2 \rrbracket}\mathcal{W}$$

4.
$$\frac{T \vdash u \ \llbracket E \rrbracket}{T \vdash w \ \llbracket F \rrbracket}\mathcal{P} \Rightarrow \frac{T \vdash u \ \llbracket E \rrbracket}{T \vdash w \ \llbracket E \wedge x_{R,k,s=1} \wedge \sigma_s \wedge u = v \rrbracket}\mathcal{P}$$

Avec $v \rightarrow w$ la règle de protocole appliquée et les formes résolues σ_s et de $u = v$ étant celles que l'on trouve dans la contrainte F.

5.
$$\frac{T \vdash u \ \llbracket E \rrbracket}{T \vdash N_i(s) \ \llbracket F \rrbracket}\mathcal{ND} \Rightarrow \frac{T \vdash u \ \llbracket E \rrbracket}{T \vdash N_i(s) \ \llbracket E \rrbracket}\mathcal{ND}$$

6.
$$\frac{T \vdash C[x] \ \llbracket E \wedge x = u \rrbracket}{T \vdash C[u] \ \llbracket F \rrbracket}\mathcal{I} \Rightarrow \frac{T \vdash C[x] \ \llbracket E \wedge x = u \rrbracket}{T \vdash C[u] \ \llbracket E \wedge x = u \rrbracket}\mathcal{I}$$

FIG. 11.1 - *Règles de réécritures pour transformer une preuve de $\overline{\mathcal{S}}'$ en preuve de $\overline{\mathcal{S}}$*

une preuve où les contraintes sont des sous-ensembles des contraintes initiales. Cette double transformation permet de supprimer les équations inutiles dans les contraintes, c'est-à-dire celles portant sur des variables qui ne sont introduites dans aucune des règles de protocole. Par la suite, on considérera que l'on travaille uniquement avec des preuves où toutes les contraintes sont utiles.

11.2 Algorithme

Le lemme précédent nous permet de travailler dans le système $\overline{\mathcal{S}}'$ tout en bénéficiant des propriétés du théorème 10.1.

Nous proposons ici un algorithme non déterministe, complet et correct qui permet de déterminer si, dans $\overline{\mathcal{S}}'$, une attaque existe sur le terme s pour un nombre donné de sessions. Cet algorithme agit en deux étapes : une étape non déterministe qui va deviner la contrainte finale et l'entrelacement des sessions puis une étape de construction par point fixe de l'ensemble des séquents accessibles à partir des axiomes. Cette approche est similaire à celle de [RT01].

11.2.1 Partie non déterministe

Nous avons en entrée le système $\overline{\mathcal{S}}'$, comprenant entre autres les règles de protocoles, et le nombre de sessions n. On note $s_1, \ldots, s_n$ les sessions qui apparaîtront. On devine les éléments suivants :

- n rôles $r_1, \ldots, r_n$ associés aux sessions $s_1, \ldots, s_n$,
- une liste ordonnée, $\mathcal{L}$, de couples (s, k) représentant un ordre total sur l'entrelacement des sessions. Si (s_1, k_1) se trouve avant (s_2, k_2) dans cette liste, la règle k_1 dans la session s_1 est jouée avant la règle k_2 dans la session s_2. Chaque couple ne peut apparaître qu'une seule fois dans la liste et il est nécessaire de respecter l'ordre naturel des règles de protocole, c'est à dire que si le couple (s, k_1) se trouve avant le couple (s, k_2) alors $k_1 < k_2$. Si (s, k) se trouve dans la liste, alors, pour tout $1 \leq l \leq l-1$, (s, l) se trouve également dans la liste. Enfin, si (s, k) se trouve dans $\mathcal{L}$, il existe une règle k pour le rôle attaché à la session s.
- pour chaque variable $x_1, \ldots, x_m$ de V apparaissant dans l'une des règles de l'un des rôles $r_1, \ldots, r_n$, un terme $\sigma(x_i) \in F(V, T, \Sigma, s, \xi)$.

Le troisième point permet d'obtenir la contrainte $E_V = \bigcup_{1 \leq i \leq m} \{x_i = \sigma(x_i)\}$. Cette contrainte sera commune à tous les séquents que nous construirons dans la section suivante. Nous mettons cette contrainte sous forme factorisée. Les contraintes seront complétées par des points de contrôle uniquement.

L'algorithme présenté ici est le même que [RT01]. Le fait de chercher la contrainte dans $F(V, T, \Sigma, s, \xi)$ plutôt que parmi les séquents $F(V, T, \Sigma, s, \xi), E$-admissible est justifié par le lemme suivant :

Lemme 11.2 *Si le séquent $T \vdash s \; [\![E]\!]$ est G, E-admissible, alors il existe $E' \models\!\mid E$ où tous les termes membres droits de E' sont dans G.*

Preuve. Soit $x = t \in E$. Comme le séquent $T \vdash s \; [\![E]\!]$ est G, E-admissible, le séquent $T \vdash t \; [\![E \setminus \{x = t\}]\!]$ est G, E-admissible. Nous avons donc soit $t \in G$ et dans ce cas là, on met $x = t$ dans E', soit il existe un terme $t' \in G$ tel que $t =_{E \setminus \{x=t\}} t'$ et on place alors $x = t'$ dans E'. On construit ainsi E' en itérant sur les égalités présentes dans E. $\square$

11.2.2 Partie déterministe

À partir de la contrainte E_V et de l'entrelacement $\mathcal{L}$ des sessions, nous allons décrire un algorithme déterministe qui va construire l'ensemble des séquents accessibles par l'intrus depuis sa connaissance initiale T.

On notera I_j l'ensemble des séquents $T \vdash u \ \llbracket E \rrbracket$ prouvables en j étapes au plus (c'est à dire avec une preuve dont la branche la plus longue est de taille au plus j) et respectant l'ordre imposé dans $\mathcal{L}$. I_1 est l'ensemble des séquents que l'on peut obtenir après application d'une règle d'axiome :

$$I_1 = \{T \vdash t \ \llbracket E_V \rrbracket \mid t \in T\}$$

En prenant comme notation $\mathcal{L}_i$ le $i$ème élément de la liste ordonnée $\mathcal{L}$, $R(s)$ le rôle attaché à la session s, on note $C_{\mathcal{L}}(i)$ la contrainte suivante :

$$C_{\mathcal{L}}(i) = \bigwedge_{\mathcal{L}_i=(s,k)} x_{R(s),k,s} = 1$$

À partir des séquents de l'ensemble I_j, on peut construire l'ensemble de séquents I_{j+1} contenant l'ensemble I_j ainsi que toutes les combinaisons possibles de séquents de I_j sur lesquels ont été appliqués une règle d'affaiblissement, de protocole, de $\mathcal{S}$, d'instanciation ou de divulgation de nonce. Dans cet ensemble, on notera comme invariant que les séquents sont de la forme $T \vdash u \ \llbracket E_v \wedge C_{\mathcal{L}}(i) \rrbracket$ pour un certain i (différent pour chaque séquent).

Le paragraphe suivant contient les règles de construction de I_{j+1} à partir de I_j. Les séquents y sont sous la forme normalisée $T \vdash u \ \llbracket E_v \wedge C_{\mathcal{L}}(i) \rrbracket$.

Inclusion Si $T \vdash u \ \llbracket E_v \wedge C_{\mathcal{L}}(i) \rrbracket$ est dans I_j, alors $T \vdash u \ \llbracket E_v \wedge C_{\mathcal{L}}(i) \rrbracket$ est dans I_{j+1}

Affaiblissement Si $T \vdash u_1 \ \llbracket E_v \wedge C_{\mathcal{L}}(i) \rrbracket$ est dans I_j, $T \vdash u_2 \ \llbracket E_v \wedge C_{\mathcal{L}}(k) \rrbracket$ est dans I_j et $k > i$, alors $T \vdash u_1 \ \llbracket E_v \wedge C_{\mathcal{L}}(k) \rrbracket$ est dans I_{j+1}

Instanciation Si $T \vdash C[x] \ \llbracket E_v \wedge C_{\mathcal{L}}(i) \rrbracket$ est dans I_j et $x = u$ est dans E_v, alors le séquent $T \vdash C[u] \ \llbracket E_v \wedge C_{\mathcal{L}}(i) \rrbracket$ est dans I_{j+1}.

Composition/Décomposition Si $T \vdash u_l \ \llbracket E_v \wedge C_{\mathcal{L}}(i_l) \rrbracket$ est dans I_j pour $1 \leq l \leq k$, alors $T \vdash u \ \llbracket E_v \wedge C_{\mathcal{L}}(i) \rrbracket$ est dans I_{j+1} avec i le plus grand des entiers $i_1, \ldots, i_k$ et u le résultat de l'application d'une règle de $\mathcal{S}$.

Divulgation de nonce Si $T \vdash u \ \llbracket E_v \wedge C_{\mathcal{L}}(i) \rrbracket$ est dans I_j et $x_{R,1,s} = 1$ est dans $C_{\mathcal{L}}(i)$, alors $T \vdash N_l(s) \ \llbracket E_v \wedge C_{\mathcal{L}}(i) \rrbracket$ est dans I_{j+1}.

Règle de protocole Si $T \vdash u \ \llbracket E_v \wedge C_{\mathcal{L}}(i) \rrbracket$ est dans I_j, $\mathcal{L}_{i+1} = (s, k)$, $v \rightarrow w$ est la règle k du rôle attaché à la session s, u s'unifie avec v, alors $T \vdash w \ \llbracket E_v \wedge C_{\mathcal{L}}(i+1) \rrbracket$ est dans I_{j+1}.

L'existence d'une attaque sur le terme s se traduit par la présence du séquent contraint $T \vdash s \ \llbracket E_V \wedge C_{\mathcal{L}}(n) \rrbracket$ dans l'un des I_j où n est le cardinal de $\mathcal{L}$.

11.2.3 Terminaison, correction et complétude

Le théorème 10.1 indique que seuls des séquents $F(V, T, \Sigma, s, \xi) - E$-admissibles peuvent apparaître dans la preuve. Ceux-ci étant en nombre fini, il existe un indice j tel que $I_{j+1} = I_j$.

En ce qui concerne la correction, chacune des règles de construction de I_j correspondent à l'application de la règle correspondante. I_1 contient les séquents déductibles à partir de la règle d'axiome.

La complétude est un peu plus délicate. En effet, nous imposons un ordre total sur les règles de protocole. Or, avec $\mathcal{L} = \{(s_1, 1), (s_2, 1)\}$, le théorème 10.1 peut nous donner comme preuve sous forme normale une preuve où la première règle des sessions s_1 et s_2 sont incomparables :

$$\dfrac{\dfrac{T \vdash u_1 \ \llbracket E_V \rrbracket}{T \vdash w_1 \ \llbracket E_V \wedge x_{R,1,s_1} = 1 \rrbracket}\mathcal{P} \quad \dfrac{T \vdash u_2 \ \llbracket E_V \rrbracket}{T \vdash w_2 \ \llbracket E_V \wedge x_{R,1,s_2} = 1 \rrbracket}\mathcal{P}}{T \vdash w_1 \ \llbracket E_V \wedge x_{R,1,s_1} = 1 \wedge x_{R,1,s_2} = 1 \rrbracket}\mathcal{W}$$

Avec notre construction de I_j, le séquent $T \vdash w_2 \ \llbracket E_V \wedge x_{R,1,s_2} = 1 \rrbracket$ ne peut pas apparaître car toutes les contraintes doivent être de la forme $E_V \wedge C_{\mathcal{L}}(i)$. Pour corriger ce problème, nous allons montrer que l'on peut transformer une preuve dans laquelle des règles de protocole sont incomparables en une preuve dans laquelle un ordre total est respecté (compatible avec l'ordre partiel).

Lemme 11.3 *Pour tout ordre partiel $\mathcal{L}'$ d'entrelacement de session défini sur les couples (s, k), pour toute preuve Π' dans $\overline{\mathcal{S}}'$ dont les règles de protocole respectent l'ordre $\mathcal{L}'$ (c'est-à-dire que si $(s_1, k_1) < (s_2, k_2)$, la règle de protocole k_2 de la session s_2 ne peut être appliquée que si la contrainte contient le point de contrôle de la règle k_1 de la session s_1), pour tout ordre total $\mathcal{L}$ compatible avec l'ordre $\mathcal{L}'$, il existe une preuve Π de $\overline{\mathcal{S}}'$ respectant l'ordre $\mathcal{L}$.*

De plus, si Π' respectait le théorème 10.1 alors Π également.

Preuve. Pour des raisons de simplicité, on impose que si (s, k) se trouve dans $\mathcal{L}'$, alors la règle de protocole correspondante se trouve dans la preuve Π'.

On exploite alors la règle de réécriture suivante, avec $(s_2, k_2) <_{\mathcal{L}} (s_1, k_1)$, Π_2 la preuve terminant par l'application de la règle de protocole k_2 pour la session s_2 :

$$\dfrac{\dfrac{\Pi_1}{T \vdash u_1 \ \llbracket E_V \wedge F_1 \rrbracket}}{T \vdash w_1 \ \llbracket E_V \wedge F_1 \wedge x_{R_1,k_1,s_1} = 1 \rrbracket}\mathcal{P}$$

$$\Rightarrow \dfrac{\dfrac{\dfrac{\Pi_1}{T \vdash u_1 \ \llbracket E_V \wedge F_1 \rrbracket} \quad \dfrac{\Pi_2}{T \vdash w_2 \ \llbracket E_V \wedge F_2 \wedge x_{R_2,k_2,s_2} = 1 \rrbracket}}{T \vdash u_1 \ \llbracket E_V \wedge F_1 \wedge F_2 \wedge x_{R_2,k_2,s_2} = 1 \rrbracket}\mathcal{W}}{T \vdash w_1 \ \llbracket E_V \wedge F_1 \wedge F_2 \wedge x_{R_1,k_1,s_1} = 1 \wedge x_{R_2,k_2,s_2} = 1 \rrbracket}\mathcal{P}$$

De plus, $x_{R_2,k_2,s_2} = 1 \notin F_1$. On propage l'insertion de la contrainte $x_{R_2,k_2,s_2} = 1$ aux séquents qui suivent.

On conserve à tout moment les invariants suivants :

- on dispose d'une preuve de $\overline{\mathcal{S}}'$,
- l'ordre partiel $\mathcal{L}'$ est respecté,
- les propriétés du théorème 10.1 sont respectées.

Lorsqu'il n'est plus possible d'appliquer la règle de réécriture, nous obtenons la preuve Π' voulue. □

Ce lemme justifie que tous les séquents sont sous la forme $T \vdash u\ [\![E_V \wedge C_{\mathcal{L}}(i)]\!]$. Il permet de conclure que toute preuve respectant le théorème 10.1 dans $\overline{\mathcal{S}}'$ et transformée comme indiqué dans ce lemme a tous ses séquents dans l'un dans I_j, y compris la preuve présentant une attaque sur le terme s. Notre algorithme est alors complet.

Nous pensons qu'il est possible de raffiner cet algorithme pour obtenir un algorithme non déterministe en la complexité de F et ainsi obtenir un algorithme NP dans le cas de Dolev-Yao, comme dans [RT01].

CHAPITRE 12

Algorithme pratique

Sommaire

Dans le chapitre 11, nous avons présenté un algorithme « en avant » qui permet de prouver l'existence d'une attaque. Si l'on voulait en plus donner une preuve de cette attaque, il nous faudrait stocker les arbres de preuve permettant d'aboutir à chacun des séquents, ce qui serait particulièrement inefficace. Nous aimerions de plus nous débarasser de la partie non déterministe tout en disposant d'un algorithme plus intelligent que la simple transformation exponentielle en algorithme déterministe. Enfin, nous voudrions un algorithme exploitable pour un nombre non borné de sessions, sans pour autant adopter l'approche naïve qui consiste à chercher successivement une preuve en une session, deux sessions, etc.

Nous allons présenter ici un algorithme de recherche de preuve en arrière. Cet algorithme fonctionne pour un nombre borné et non borné de sessions et il est correct et complet.

Nous présentons dans la section 12.1 un certain nombre de normalisations sur les preuves de $\overline{\mathcal{S}}$ qui nous seront utiles pour améliorer en pratique l'algorithme de recherche en arrière. Ces normalisations visent à limiter les occurrences possibles de certaines règles. Dans la section 12.2 page 143, nous présentons l'algorithme en lui-même ainsi que les résultats de correction et de complétude associés.

12.1 Normalisations

Pour effectuer une recherche de preuve « en arrière », nous devons à partir d'un terme trouver quelle règle a pu être appliquée et quelles ont été les hypothèses utilisées. Dans ce cadre, certaines règles de $\overline{\mathcal{S}}$ peuvent accepter des hypothèses arbitraires pour une même conclusion. Ces règles sont les règles d'affaiblissement, de divulgation de nonce.

Le premier résultat, présenté dans la section 12.1.1, nous permet de restreindre la forme des preuves sous forme normale : les règles d'affaiblissement ne seront désormais utilisées que lorsqu'il nous sera nécessaire d'ajouter un point de contrôle dans la contrainte d'un séquent.

Le second résultat, présenté dans la section 12.1.2, impose d'utiliser les règles de divulgation de nonce juste après l'application de la première règle de protocole pour la session concernée.

Le résultat de la section 12.1.3 page 140 impose d'utiliser les règles de $\mathcal{S}$ uniquement quand toutes les contraintes des hypothèses sont identiques (y compris les points de contrôle). Cela évite de cumuler dans de telles règles le rôle des règles d'affaiblissement.

12.1.1 Normalisation des règles d'affaiblissement

Considérons une preuve de $\overline{\mathcal{S}}$ que l'on transforme en une preuve de $\overline{\mathcal{S}}'$. Dans cette nouvelle preuve, on rencontre des règles d'affaiblissement de la forme :

$$\frac{T \vdash u_1 \ \llbracket E \rrbracket \quad T \vdash u_2 \ \llbracket E \rrbracket}{T \vdash u_1 \ \llbracket E \rrbracket} \mathcal{W}$$

. Ces règles sont inutiles et contredisent la minimalité de la preuve : on peut donc les supprimer. De manière plus générale, on peut supprimer toutes les règles de la forme

$$\frac{T \vdash u_1 \ \llbracket E \rrbracket \quad T \vdash u_2 \ \llbracket F \rrbracket}{T \vdash u_1 \ \llbracket E \rrbracket} \mathcal{W}$$

avec $F \subset E$ pour les mêmes raisons.

Cette remarque va nous permettre de normaliser les règles d'affaiblissement de façon à ce que toute règle d'affaiblissement ait comme preuve ancrée à droite une preuve terminant par une règle de protocole. Pour cela, nous utilisons les règles de réécriture des figures 12.1 page ci-contre et 12.2 page 140. Par convention, les contraintes $F_1, F_2, \ldots$ ne contiennent que de des contraintes sur les points de contrôle de ξ tandis que la contrainte E n'en contient pas. Cette convention n'est pas respectée sur la première règle.

À tout moment, on a l'invariant que la preuve est une preuve de $\overline{\mathcal{S}}'$. On considère par la suite que toutes les preuves de $\overline{\mathcal{S}}'$ se sont vues appliquer ces règles de réécriture et donc que les règles d'affaiblissement ont comme seconde hypothèse une preuve terminant par une règle de protocole. Le cas de la règle d'axiome est comprise dans le premier cas de la figure 12.1 car la contrainte ne peut contenir de variables de ξ.

12.1.2 Normalisation des règles de divulgation de nonce

L'application d'une règle de divulgation permet d'obtenir un nonce à la seule condition que nous avons dans la contrainte le point de contrôle indiquant que la première règle de la session concernée a bien été jouée (et bien sûr, que l'agent de la session considérée est compromis). Il paraît donc possible de limiter l'utilisation d'une telle règle suite à la règle de protocole en question. Pour cela, nous appliquons les règles des figures 12.3 page 141 et 12.4 page 142.

À tout moment, on a l'invariant que la preuve est une preuve de $\overline{\mathcal{S}}'$. On notera qu'une règle de divulgation de nonce ne peut pas suivre une règle d'axiome. Ainsi, l'ensemble des

1.

$$\dfrac{\dfrac{\Pi_1}{T \vdash u_1 \;\; \llbracket E \rrbracket} \quad \dfrac{\Pi_2}{T \vdash u_2 \;\; \llbracket F \rrbracket}}{T \vdash u_1 \;\; \llbracket E \rrbracket}\mathcal{W} \Rightarrow \dfrac{\Pi_1}{T \vdash u_1 \;\; \llbracket E \rrbracket}$$

avec $F \subset E$

2.

$$\dfrac{\dfrac{\Pi_1}{T \vdash u_1 \;\; \llbracket E \wedge F_1 \rrbracket} \quad \dfrac{\dfrac{\Pi_2}{T \vdash C[x] \;\; \llbracket E \wedge F_2 \rrbracket}}{T \vdash C[u] \;\; \llbracket E \wedge F_2 \rrbracket}\mathcal{I}}{T \vdash u_1 \;\; \llbracket E \wedge F_1 \wedge F_2 \rrbracket}\mathcal{W}$$

$$\Rightarrow \dfrac{\dfrac{\Pi_1}{T \vdash u_1 \;\; \llbracket E \wedge F_1 \rrbracket} \quad \dfrac{\Pi_2}{T \vdash C[x] \;\; \llbracket E \wedge F_2 \rrbracket}}{T \vdash u_1 \;\; \llbracket E \wedge F_1 \wedge F_2 \rrbracket}\mathcal{W}$$

3.

$$\dfrac{\dfrac{\Pi_1}{T \vdash u_1 \;\; \llbracket E \wedge F_1 \rrbracket} \quad \dfrac{\dfrac{\Pi_2}{T \vdash u_2 \;\; \llbracket E \wedge F_2 \rrbracket}}{T \vdash N_i(s) \;\; \llbracket E \wedge F_2 \rrbracket}\mathcal{ND}}{T \vdash u_1 \;\; \llbracket E \wedge F_1 \wedge F_2 \rrbracket}\mathcal{W}$$

$$\Rightarrow \dfrac{\dfrac{\Pi_1}{T \vdash u_1 \;\; \llbracket E \wedge F_1 \rrbracket} \quad \dfrac{\Pi_2}{T \vdash u_2 \;\; \llbracket E \wedge F_2 \rrbracket}}{T \vdash u_1 \;\; \llbracket E \wedge F_1 \wedge F_2 \rrbracket}\mathcal{W}$$

4.

$$\dfrac{\dfrac{\Pi_1}{T \vdash u_1 \;\; \llbracket E \wedge F_1 \rrbracket} \quad \dfrac{\dfrac{\Pi_2}{T \vdash u_2 \;\; \llbracket E \wedge F_2 \rrbracket} \; \ldots \; \dfrac{\Pi_n}{T \vdash u_n \;\; \llbracket E \wedge F_n \rrbracket}}{T \vdash u \;\; \llbracket E \wedge F_2 \wedge \ldots \wedge F_n \rrbracket}\mathcal{S}}{T \vdash u_1 \;\; \llbracket E \wedge F_1 \wedge F_2 \wedge \ldots \wedge F_n \rrbracket}\mathcal{W}$$

$$\Rightarrow \dfrac{\dfrac{\Pi_1}{T \vdash u_1 \;\; \llbracket E \wedge F_1 \rrbracket} \quad \dfrac{\Pi'}{T \vdash u_2 \;\; \llbracket E \wedge F_2 \wedge \ldots \wedge F_n \rrbracket}}{T \vdash u_1 \;\; \llbracket E \wedge F_1 \wedge F_2 \wedge \ldots \wedge F_n \rrbracket}\mathcal{W}$$

avec Π' le résultat de l'application de la règle de réécriture de la figure 10.1 page 83 du lemme 10.1 page 83.

FIG. 12.1 - *Règles de réécriture pour la normalisation des règles d'affaiblissement (1)*

5.

$$
\dfrac{\dfrac{\Pi_1}{T \vdash u_1 \ \llbracket E \wedge F_1 \rrbracket} \qquad \dfrac{\dfrac{\Pi_2}{T \vdash u_2 \ \llbracket E \wedge F_2 \rrbracket} \quad \dfrac{\dfrac{\Pi_3}{T \vdash u_3 \ \llbracket E \wedge F_3 \wedge x_{R',k',s'} = 1 \rrbracket}}{T \vdash w \ \llbracket E \wedge F_2 \wedge x_{R,k,s} = 1 \wedge x_{R',k',s'} = 1 \rrbracket}\,\mathcal{P}}{T \vdash u_2 \ \llbracket E \wedge F_2 \wedge F_3 \wedge x_{R,k,s} = 1 \wedge x_{R',k',s'} = 1 \rrbracket}\,\mathcal{W}}{T \vdash u_2 \ \llbracket E \wedge F_1 \wedge F_2 \wedge F_3 \wedge x_{R,k,s} = 1 \wedge x_{R',k',s'} = 1 \rrbracket}\,\mathcal{W}
$$

$$
\Rightarrow \dfrac{\dfrac{\Pi_1}{T \vdash u_1 \ \llbracket E \wedge F_1 \rrbracket} \qquad \dfrac{\dfrac{\Pi_2}{T \vdash u_2 \ \llbracket E \wedge F_2 \rrbracket} \quad \dfrac{\Pi_3}{T \vdash u_2 \ \llbracket E \wedge F_2 \wedge F_3 \wedge x_{R',k',s'} = 1 \rrbracket}\,\mathcal{W}}{T \vdash w \ \llbracket E \wedge F_2 \wedge F_3 \wedge x_{R',k',s'} = 1 \wedge x_{R,k,s} = 1 \rrbracket}\,\mathcal{P}}{T \vdash u_1 \ \llbracket E \wedge F_1 \wedge F_2 \wedge F_3 \wedge x_{R',k',s'} = 1 \wedge x_{R,k,s} = 1 \rrbracket}\,\mathcal{W}
$$

avec $v \rightarrow w$ la règle k de la session s.

FIG. 12.2 - *Règles de réécriture pour la normalisation des règles d'affaiblissement (2)*

règles des figures 12.3 et 12.4 page 142 permet d'obtenir la propriété voulue : toute règle de divulgation de nonce est précédée de la première règle de protocole qui correspond à sa session.

12.1.3 Normalisation des règles de $\mathcal{S}$

Les règles de $\mathcal{S}$ partagent un rôle avec les règles d'affaiblissement : elles permettent de déplacer le point de contrôle dans une session d'un protocole. Nous allons, dans cette section, déléguer ce rôle aux règles d'affaiblissement. Pour cela, nous utilisons la règle de réécriture de la figure 12.5 page 142.

À tout moment, on a l'invariant que la preuve est une preuve de $\overline{\mathcal{S}}'$. Ainsi, quand il n'est plus possible d'appliquer la règle de réécriture, toutes les règles de $\mathcal{S}$ sont appliquées avec des contraintes identiques.

12.1.4 Non interférence

Pour conclure cette section, nous noterons le lemme suivant :

Lemme 12.1 *Soit Π une preuve de $\overline{\mathcal{S}}$ vérifiant le théorème 10.1. Soit Π_1 la transformation de cette preuve en une preuve de $\overline{\mathcal{S}}'$ comme décrit dans la section 11.1 page 129, Π_2 la preuve obtenue par normalisation des règles d'affaiblissement comme décrit dans la section 12.1.1, celle des règles de divulgation de nonce de la section 12.1.2 et celle des règles de $\mathcal{S}$ dans la section 12.1.3 et enfin Π_3 la preuve obtenue par transformation de Π_2 en une preuve de $\overline{\mathcal{S}}$.*

Π_3 vérifie également le théorème 10.1.

Preuve. Les transformations des figures 12.1 page précédente, 12.2, 12.3 page suivante et 12.4 page 142 n'altèrent pas la $F(V, T, \Sigma, s, \xi) - E'$-admissibilité du théorème. Elles peuvent de

1.

$$\dfrac{\dfrac{\dfrac{\Pi_1}{T \vdash u_1 \ [\![E_1 \wedge x_{R,1,s} = 1]\!]} \quad \dfrac{\Pi_2}{T \vdash u_2 \ [\![E_2]\!]}}{T \vdash u_1 \ [\![E_1 \wedge E_2 \wedge x_{R,1,s} = 1]\!]}\,\mathcal{W}}{T \vdash N_i(s) \ [\![E_1 \wedge E_2 \wedge x_{R,1,s} = 1]\!]}\,\mathcal{ND}$$

$$\Rightarrow \dfrac{\dfrac{\dfrac{\Pi_1}{T \vdash u_1 \ [\![E_1 \wedge x_{R,1,s} = 1]\!]}}{T \vdash N_i(s) \ [\![E_1 \wedge x_{R,1,s} = 1]\!]}\,\mathcal{ND} \quad \dfrac{\Pi_2}{T \vdash u_2 \ [\![E_2]\!]}}{T \vdash N_i(s) \ [\![E_1 \wedge E_2 \wedge x_{R,1,s} = 1]\!]}\,\mathcal{W}$$

Une règle similaire existe dans le cas où le point de contrôle est dans Π_2.

2.

$$\dfrac{\dfrac{\dfrac{\Pi_1}{T \vdash u_1 \ [\![E_1 \wedge x_{R,1,s} = 1]\!]} \ \ldots \ \dfrac{\Pi_2}{T \vdash u_k \ [\![E_k]\!]}}{T \vdash u \ [\![E_1 \wedge \cdots \wedge E_k \wedge x_{R,1,s} = 1]\!]}\,\mathcal{S}}{T \vdash N_i(s) \ [\![E_1 \wedge \ldots \wedge E_k \wedge x_{R,1,s} = 1]\!]}\,\mathcal{ND}$$

$$\Rightarrow \dfrac{\dfrac{\Pi'}{T \vdash u_1 \ [\![E_1 \wedge \ldots \wedge E_k \wedge x_{R,1,s} = 1]\!]}}{T \vdash N_i(s) \ [\![E_1 \wedge x_{R,1,s} = 1]\!]}\,\mathcal{ND}$$

Π' est l'application du lemme 10.1 page 83 sur la règle de $\mathcal{S}$. On appliquera ensuite la première règle de réécriture pour traiter les règles d'affaiblissement introduites. Des règles similaires existent dans le cas où le point de contrôle est dans $\Pi_2, \ldots, \Pi_k$.

3.

$$\dfrac{\dfrac{\dfrac{\Pi}{T \vdash C[x] \ [\![E \wedge x_{R,1,s} = 1]\!]}}{T \vdash C[u] \ [\![E \wedge x_{R,1,s} = 1]\!]}\,\mathcal{I}}{T \vdash N_i(s) \ [\![E \wedge x_{R,1,s} = 1]\!]}\,\mathcal{ND} \quad \Rightarrow \dfrac{\dfrac{\Pi}{T \vdash C[x] \ [\![E \wedge x_{R,1,s} = 1]\!]}}{T \vdash N_i(s) \ [\![E \wedge x_{R,1,s} = 1]\!]}\,\mathcal{ND}$$

4.

$$\dfrac{\dfrac{\dfrac{\Pi}{T \vdash t \ [\![E \wedge x_{R,1,s} = 1 \wedge x_{R',1,s'}]\!]}}{T \vdash N_{i'}(s') \ [\![E \wedge x_{R,1,s} = 1 \wedge x_{R',1,s'}]\!]}\,\mathcal{ND}}{T \vdash N_i(s) \ [\![E \wedge x_{R,1,s} = 1 \wedge x_{R',1,s'}]\!]}\,\mathcal{ND} \Rightarrow \dfrac{\dfrac{\Pi}{T \vdash t \ [\![E \wedge x_{R,1,s} = 1 \wedge x_{R',1,s'}]\!]}}{T \vdash N_i(s) \ [\![E \wedge x_{R,1,s} = 1 \wedge x_{R',1,s'}]\!]}\,\mathcal{ND}$$

FIG. 12.3 - *Règles de réécriture pour la normalisation des règles de divulgation de nonce (1)*

5.

$$\dfrac{\dfrac{\dfrac{\Pi}{T \vdash u \ \llbracket E \wedge x_{R,1,s} = 1 \rrbracket}}{T \vdash w \ \llbracket E \wedge x_{R,1,s} = 1 \wedge x_{R',k',s'} = 1 \rrbracket}\,\mathcal{P}}{T \vdash N_i(s) \ \llbracket E \wedge x_{R,1,s} = 1 \wedge x_{R',k',s'} = 1 \rrbracket}\,\mathcal{ND}$$

$$\Rightarrow \dfrac{\dfrac{\dfrac{\Pi}{T \vdash u \ \llbracket E \wedge x_{R,1,s} = 1 \rrbracket}}{T \vdash N_i(s) \ \llbracket E \wedge x_{R,1,s} = 1 \rrbracket}\,\mathcal{ND} \qquad \dfrac{\dfrac{\Pi}{T \vdash u \ \llbracket E \wedge x_{R,1,s} = 1 \rrbracket}}{T \vdash w \ \llbracket E \wedge x_{R,1,s} = 1 \wedge x_{R',k',s'} = 1 \rrbracket}\,\mathcal{P}}{T \vdash N_i(s) \ \llbracket E \wedge x_{R,1,s} = 1 \wedge x_{R',k',s'} = 1 \rrbracket}\,\mathcal{W}$$

si $s \neq s'$.

FIG. 12.4 - *Règles de réécriture pour la normalisation des règles de divulgation de nonce (2)*

$$\dfrac{\dfrac{\Pi_1}{T \vdash u_1 \ \llbracket E_1 \rrbracket} \ \ldots \ \dfrac{\Pi_i}{T \vdash u_i \ \llbracket E_i \rrbracket} \ \ldots \ \dfrac{\Pi_j}{T \vdash u_j \ \llbracket E_j \wedge x_{R,k,s} = 1 \rrbracket} \ \ldots \ \dfrac{\Pi_k}{T \vdash u_k \ \llbracket E_k \rrbracket}}{T \vdash u \ \llbracket E_1 \wedge \ldots \wedge E_k \wedge x_{R,k,s} = 1 \rrbracket}\,\mathcal{S}$$

$$\Rightarrow \dfrac{\dfrac{\dfrac{\Pi_1}{T \vdash u_1 \ \llbracket E_1 \rrbracket} \quad \dfrac{\Pi_j}{T \vdash u_j \ \llbracket E_j \wedge x_{R,k,s} = 1 \rrbracket}}{T \vdash u_1 \ \llbracket E_1 \wedge x_{R,k,s} = 1 \rrbracket}\,\mathcal{W} \qquad \ldots \qquad \dfrac{\dfrac{\Pi_k}{T \vdash u_k \ \llbracket E_1 \rrbracket} \quad \dfrac{\Pi_j}{T \vdash u_j \ \llbracket E_j \wedge x_{R,k,s} = 1 \rrbracket}}{T \vdash u_k \ \llbracket E_1 \wedge x_{R,k,s} = 1 \rrbracket}\,\mathcal{W}}{T \vdash u \ \llbracket E_1 \wedge \ldots \wedge E_k \wedge x_{R,k,s} = 1 \rrbracket}\,\mathcal{S}$$

avec $x_{R,k,s} = 1 \notin E_i$; on notera que l'on peut éviter les règles d'affaiblissement sur Π_l quand $x_{R,k,s} = 1 \in E_l$; dans le cas contraire, on complète cette règle de réécriture par l'application de la première règle de la figure 12.1 page 139.

FIG. 12.5 - *Règle de réécriture pour la normalisation des règles de $\mathcal{S}$*

plus être vues comme un seul système de réécriture convergent. En effet, la première règle de la figure 12.3 ne change pas le prédécesseur droit de la règle d'affaiblissement et la règle d'affaiblissement introduite dans la figure 12.4 n'est pas du type de la première règle de la figure 12.1 : $x_{R',k',s'}$ ne peut pas se trouver dans $E \wedge x_{R,1,s} = 1$ car la règle de protocole ne pourrait être appliquée dans ce cas. La règle de la figure 12.5 page précédente ne s'oppose à aucune des règles de réécriture pour la règle d'affaiblissement.

Le lemme 11.1 page 130 permet alors de conclure. □

12.2 Algorithme

Nous allons présenter ici un algorithme de recherche de preuve « en arrière », c'est-à-dire partant du séquent final. Nous ne disposons toutefois que du terme s et non du séquent en entier. Deux approches sont alors possibles :

- deviner la contrainte finale et tomber alors dans un algorithme non déterministe ou
- effectuer une construction paresseuse de la contrainte finale.

Nous allons mener ici la seconde approche. Nous considérons que l'algorithme est paramétré par une fonction de choix S qui indiquera une stratégie pour le parcours de l'arbre de preuve.

12.2.1 Description

Pour construire la preuve du séquent $T \vdash s \ [\![E]\!]$, nous allons appliquer l'ensemble de règles de réécriture de la figure 12.6 page suivante et celles de la figure 12.7 page 145 sur la preuve de départ suivante :

$$\frac{\Pi_?}{T \vdash s \ [\![E_?]\!]}$$

La contrainte $E_?$ est un symbole indiquant une contrainte « quelconque ». Bien entendu, cette partie de la contrainte devra être compatible avec la seconde partie de la contrainte du séquent. Cela explique la présence de contraintes du type $x_{R,k,s} \neq 1$: elles permettent de faire respecter l'ordre des règles de protocole. De telles contraintes seront retirées de la preuve finale.

De même, le symbole $\Pi_?$ désigne une preuve qu'il nous reste à construire. Ces deux symboles sont destinés à appliquer les règles de réécriture : si le symbole $\Pi_?$ apparaît à deux endroits différents de la preuve, il ne représente pas forcément la même preuve.

Pour chaque règle de réécriture, il y a deux conditions nécessaires implicites :

- chaque contrainte doit être sous forme factorisée et avoir une solution ;
- tous les séquents doivent être $F(V, T, \Sigma, s), E$-admissible.

On interdit les arbres de preuve non minimaux, c'est-à-dire ceux sur lesquels un même séquent apparaît deux fois sur une même branche. Un tel cas est considéré comme un échec. Un second cas d'échec est l'impossibilité d'appliquer une des règles de réécriture.

Nous allons exploiter l'arbre de recherche tel que pour chaque nœud de celui-ci, les fils sont le résultat de l'application d'une règle de réécriture à la preuve attachée au nœud, à condition que cette preuve soit minimale et que les hypothèses du théorème 10.1 soient vérifiées (une telle vérification est possible car les propriétés sur les séquents ne dépendent que du séquent final).

1.

$$\frac{\Pi_?}{T \vdash w \;\; [\![E_? \wedge E']\!]} \Rightarrow \dfrac{\dfrac{\Pi_?}{T \vdash u \;\; [\![E' \wedge E_? \wedge \bigwedge_{l \geq k} x_{R,l,s} \neq 1 \wedge \bigwedge_{l < k} x_{R,l,s} = 1]\!]}}{T \vdash w \;\; [\![E' \wedge E_? \wedge x_{R,k,s} = 1 \wedge \sigma_s \wedge u = v]\!]}\,\mathcal{P}$$

si $v \to w$ est la règle k du rôle R.

2.

$$\frac{\Pi_?}{T \vdash N_i(s) \;\; [\![E' \wedge E_?]\!]} \Rightarrow \dfrac{\dfrac{\dfrac{\Pi_?}{T \vdash u \;\; [\![E' \wedge E_? \wedge \bigwedge_{l \geq 1} x_{R,l,s} \neq 1]\!]}}{T \vdash w \;\; [\![E' \wedge E_? \wedge x_{R,1,s} = 1 \wedge \sigma_s \wedge u = v]\!]}\,\mathcal{P}}{T \vdash N_i(s) \;\; [\![E' \wedge E_? \wedge x_{R,1,s} = 1 \wedge \sigma_s \wedge u = v]\!]}\,\mathcal{ND}$$

si $v \to w$ est la règle 1 du rôle R.

3.

$$\frac{\Pi_?}{T \vdash C[u] \;\; [\![E' \wedge E_?]\!]} \Rightarrow \dfrac{\dfrac{\Pi_?}{T \vdash C[x] \;\; [\![E' \wedge E_? \wedge x = u]\!]}}{T \vdash C[u] \;\; [\![E' \wedge E_? \wedge x = u]\!]}\,\mathcal{I}$$

4.

$$\frac{\Pi_?}{T \vdash u \;\; [\![E' \wedge E_?]\!]} \Rightarrow \dfrac{\dfrac{\Pi_?}{T \vdash u_1 \;\; [\![E' \wedge E_?]\!]} \quad \ldots \quad \dfrac{\Pi_?}{T \vdash u_k \;\; [\![E' \wedge E_?]\!]}}{T \vdash u \;\; [\![E' \wedge E_?]\!]}\,\mathcal{S}$$

si l'application de la règle de $\mathcal{S}$ est valide.

5.

$$\frac{\Pi_?}{T \vdash t \;\; [\![E' \wedge E_?]\!]} \Rightarrow \frac{}{T \vdash t \;\; [\![E']\!]}\,\mathcal{A}$$

si $t \in T$ et que E' ne contient pas de variables de contrôle positives.

FIG. 12.6 - *Règles de réécriture pour la construction de la preuve d'un séquent dans $\overline{\mathcal{S}}'$ (1)*

6.

$$\frac{\Pi_?}{T \vdash t \;\; [\![E' \wedge x_{R,k,s} = 1 \wedge E_?]\!]}$$

$$\Rightarrow \frac{\dfrac{\Pi_?}{T \vdash t \;\; [\![E' \wedge E_?]\!]} \qquad \dfrac{\dfrac{\Pi_?}{T \vdash u \;\; [\![E' \wedge E_? \wedge \bigwedge\limits_{l \geq k} x_{R,l,s} \neq 1 \wedge \bigwedge\limits_{l<k} x_{R,l,s} = 1]\!]}}{T \vdash w \;\; [\![E' \wedge x_{R,k,s} = 1 \wedge x_{R,k,s} = 1 \wedge u = v \wedge \sigma_s \wedge E_?]\!]}\,\mathcal{P}}{T \vdash t \;\; [\![E' \wedge x_{R,k,s} = 1 \wedge E_?]\!]}\,\mathcal{W}$$

si $v \rightarrow w$ est la règle k du rôle R. Pour limiter l'exploration, il est également possible d'imposer $t \in T$: on cherche à appliquer une règle d'axiome une fois que l'on s'est débarrassé des points de contrôle.

7.

$$\frac{\dfrac{\Pi_1}{T \vdash u_1 \;\; [\![E'_1]\!]} \;\; \dots \;\; \dfrac{\Pi_k}{T \vdash u_k \;\; [\![E'_k]\!]}}{T \vdash u' \;\; [\![E'' \wedge E_?]\!]} \Rightarrow \frac{\dfrac{\Pi_1}{T \vdash u_1 \;\; [\![E'_1]\!]} \;\; \dots \;\; \dfrac{\Pi_k}{T \vdash u_k \;\; [\![E'_k]\!]}}{T \vdash u' \;\; [\![E'']\!]}$$

si $E_?$ n'apparaît pas dans E'', $E'_1, \dots, E'_k$. Cette règle est destinée à retirer les symboles $E_?$ sur une branche lorsqu'on a atteint la fin de la branche.

FIG. 12.7 - *Règles de réécriture pour la construction de la preuve d'un séquent dans $\overline{\mathcal{S}}'$ (2)*

Si plusieurs règles de réécriture s'appliquent à un même endroit, cela constitue autant de fils pour l'arbre de recherche. Par contre, pour choisir la feuille de la preuve à développer, on exploite la stratégie donnée par la fonction de choix S. Voici quelques exemples (informels) de fonction :

- développer la feuille la plus à gauche dans la représentation classique en arbre de la preuve ;
- développer la feuille de profondeur minimale la plus à gauche ;
- développer une feuille au hasard ;
- développer une feuille au hasard parmi les feuilles où une ouverture de session ne peut être appliquée ; choisir une des autres stratégies si une ouverture de session peut être appliquée sur chacune des feuilles.

Nous verrons que cette approche est aussi complète que celle consistant à développer toutes les feuilles possibles à chacun des nœuds de l'arbre de recherche. Elle permet cependant de limiter les branchements (sans autre incidence).

La stratégie de recherche S n'a pas d'importance lorsque l'on travaille avec un nombre borné de sessions car nous verrons dans la section 12.2.2 que l'arbre de recherche est fini.

12.2.2 Terminaison

Nous allons montrer que l'arbre de recherche est fini dans le cas d'un nombre borné de sessions. On ignore la dernière règle de réécriture que l'on considère comme un simple nettoyage à effectuer à la fin.

Les nœuds de cet arbre sont des preuves partielles de $\overline{\mathcal{S}}'$ (certaines hypothèses sont remplacées par le symbole $\Pi_?$). Les règles de réécriture s'appliquant toujours aux feuilles des arbres de preuve, il n'est pas possible d'obtenir deux fois le même arbre de preuve à deux endroits différents d'une même branche de l'arbre de recherche.

Il nous reste donc à montrer qu'il existe un nombre fini d'arbres de preuve. Chacun des arbres de preuve (une fois transposé dans $\overline{\mathcal{S}}$) vérifie le théorème 10.1 ; ils sont de plus minimaux. Ces deux propriétés entraînent que la taille d'une branche d'un arbre de preuve est bornée : les séquents doivent être $F(V, T, \Sigma, s, \xi), E$-admissibles (ils sont donc en nombre fini) et ils ne peuvent se répéter deux fois. Il n'y a donc qu'un nombre fini d'arbres de preuve possibles car les branchements sont bornés par le maximum entre 2 et le branchement maximal des règles de $\mathcal{S}$.

On remarquera que les normalisations des sections 12.1.1, 12.1.2 et 12.1.3 page 140 ne sont pas nécessaires pour prouver la terminaison. Il est donc possible, pour ce point, de travailler directement dans $\overline{\mathcal{S}}$.

12.2.3 Correction et complétude

Les règles de réécriture des figures 12.6 page 144 et 12.7 page précédente permettent d'obtenir une preuve de $\overline{\mathcal{S}}'$. Ainsi, si une attaque est trouvée, celle-ci est valide. L'algorithme est donc correct si l'on nettoie la preuve obtenue à l'aide de la dernière règle de réécriture et que l'on supprime les points de contrôle négatifs.

En ce qui concerne la complétude, le théorème 10.1 nous indique que si une attaque existe, il existe une preuve dans $\overline{\mathcal{S}}$ vérifiant les hypothèses du théorème. Le lemme 12.1 page 140 nous permet d'affirmer qu'il existe aussi une preuve dans $\overline{\mathcal{S}}'$ où les règles d'affaiblissement et de divulgation de nonces sont normalisées. Soit Π une telle preuve. Nous allons montrer qu'elle se trouve dans notre arbre de recherche.

On note $\sigma(\Pi)$ l'ensemble des arbres de preuve obtenus à partir de Π en coupant une ou plusieurs branches. Lorsqu'une branche est coupée, elle est remplacée par :

$$\frac{\Pi_?}{T \vdash u \ \llbracket E \rrbracket}$$

où $T \vdash u \ \llbracket E \rrbracket$ est le séquent sur lequel était enracinée la branche.

On exclut de $\sigma(\Pi)$ les arbres de preuve contenant en feuille l'une des deux constructions suivantes :

$$\frac{\dfrac{\Pi_1}{T \vdash u_1 \ \llbracket E_1 \rrbracket} \quad \dfrac{\Pi_?}{T \vdash u_2 \ \llbracket E_2 \rrbracket}}{T \vdash u_1 \ \llbracket E_1 \wedge E_2 \rrbracket}\mathcal{W}$$

$$\frac{\dfrac{\Pi_?}{T \vdash u \ \llbracket E \rrbracket}}{T \vdash N_i(s) \ \llbracket E \rrbracket}\mathcal{ND}$$

Ainsi, les règles de divulgation de nonce et d'affaiblissement sont toujours accompagnées de la règle de protocole qui lui correspond.

On définit la distance $d_{\sigma(\Pi)}(\pi_1, \pi_2) = |n(\pi_1) - n(\pi_2)|$ la distance entre deux preuves π_1 et π_2 de $\sigma(\Pi)$ où $n(\pi)$ est le nombre de séquents de la preuve π diminué du nombre d'occurrences de $\Pi_?$. On note $N = n(\Pi)$ le maximum de la fonction $d_{\sigma(\Pi)}$. Le minimum étant 0.

On procède par récurrence sur les entiers de N à 0 en utilisant l'hypothèse suivante : pour tout entier n, il existe un entier $m \leq n$ une preuve π_m dans $\sigma(\Pi)$ telle que $d_{\sigma(\Pi)}(\pi_n, \Pi) = m$ et telle que π_m se trouve dans l'arbre de recherche. Pour $n = 0$, nous pourrons conclure sur la présence de Π dans l'arbre de recherche.

La seule preuve ne contenant qu'un séquent dans $\sigma(\Pi)$ est la suivante :

$$\frac{\Pi_?}{T \vdash s \ \llbracket E_? \rrbracket}$$

et elle se trouve à la racine de notre arbre de recherche.

Supposons désormais que l'hypothèse de récurrence est vérifiée pour $n > 0$. On a donc un entier $m \leq n$, une preuve π_m dans $\sigma(\Pi)$ telle que $d_{\sigma(\Pi)}(\pi_n, \Pi) = m$ et telle que π_m se trouve dans l'arbre de recherche. Si $m \leq n - 1$, on peut conclure immédiatement car l'entier m et la preuve π_m conviennent déjà.

Dans le cas contraire, au niveau du nœud où se trouve π_m dans l'arbre de recherche, la fonction de choix S désigne l'une des feuilles $\Pi_?$ à la position p de π_m. Il existe donc une preuve Π'' telle que :

$$\pi_m = \frac{\Pi_?}{\Pi''}$$

Considérons la règle appliquée à la position p dans Π. Selon le type de règle qui se trouve à cette position, nous allons construire une preuve Π' de $\sigma(\Pi)$ qui se trouve être un des fils de π_m dans l'arbre de recherche. Cette preuve Π' sera celle qui vérifiera l'hypothèse de récurrence.

Règle d'axiome

$$\Pi' = \cfrac{\overline{T \vdash t \ \ [\![E']\!]}\ \mathcal{A}}{\Pi''}$$

Comme nous avons pu appliquer la règle d'axiome dans Π à la position p, c'est que $t \in T$ et que E' ne contient pas de points de contrôle. Nous pouvons donc appliquer la cinquième règle de réécriture de la figure 12.6 page 144 sur π_m qui nous permet d'obtenir Π'. La preuve obtenue diminue strictement la distance avec Π.

Règle de protocole

$$\Pi' = \cfrac{\cfrac{\cfrac{\Pi_?}{T \vdash u \ \ [\![E' \wedge E_? \wedge \bigwedge_{l \geq k} x_{R,l,s} \neq 1 \wedge \bigwedge_{l<k} x_{R,l,s} = 1]\!]}}{T \vdash w \ \ [\![E' \wedge E_? \wedge x_{R,k,s} = 1 \wedge \sigma_s \wedge u = v]\!]}\ \mathcal{P}}{\Pi''}$$

Comme nous avons pu appliquer la règle de protocole dans Π, nous sommes dans les conditions nécessaires pour appliquer la première règle de réécriture de la figure 12.6 sur π_m et nous obtenons Π'. Là encore, la distance avec Π diminue strictement.

Règle d'instanciation

$$\Pi' = \cfrac{\cfrac{\cfrac{\Pi_?}{T \vdash C[x] \ \ [\![E' \wedge E_? \wedge x = u]\!]}}{T \vdash C[u] \ \ [\![E' \wedge E_? \wedge x = u]\!]}\ \mathcal{I}}{\Pi''}$$

Nous procédons de la même façon. C'est la troisième règle de la figure 12.6 que l'on applique sur π_m.

Règle de $\mathcal{S}$

$$\Pi' = \cfrac{\cfrac{\cfrac{\Pi_?}{T \vdash u_1 \ \ [\![E']\!]} \ \ \ldots \ \ \cfrac{\Pi_?}{T \vdash u_i \ \ [\![E']\!]} \ \ \ldots \ \ \cfrac{\Pi_?}{T \vdash u_k \ \ [\![E']\!]}}{T \vdash u \ \ [\![E']\!]}\ \mathcal{S}}{\Pi''}$$

On procède comme précédemment en appliquant la quatrième règle.

On notera que le fait de ne considérer que les contraintes identiques est une approche complète grâce à la normalisation de la section 12.1.3 page 140.

Règle de divulgation de nonce . On sait que dans ce cas, la règle de divulgation de nonce est précédée dans Π par la règle de protocole qui lui correspond. Nous considérons donc également cette règle de protocole dans Π'. Nous sommes alors en présence du cas suivant :

$$\Pi' = \cfrac{\cfrac{\cfrac{\Pi_?}{T \vdash u \ [\![E' \wedge E_? \wedge \bigwedge_{l \geq 1} x_{R,l,s} \neq 1]\!]}}{T \vdash w \ [\![E' \wedge E_? \wedge x_{R,1,s} = 1 \wedge \sigma_s \wedge u = v]\!]}\ \mathcal{P}}{T \vdash N_i(s) \ [\![E' \wedge E_? \wedge x_{R,1,s} = 1 \wedge \sigma_s \wedge u = v]\!]}\ \mathcal{ND}}{\Pi''}$$

On peut alors procéder comme dans les cas précédents. On utilise la seconde règle de la figure 12.6 page 144.

Règle d'affaiblissement . Comme pour le cas de la divulgation de nonce, on sait que la règle est précédée dans Π, à droite, par une règle de protocole. On se retrouve dans le cas suivant :

$$\Pi' = \cfrac{\cfrac{\cfrac{\Pi_?}{T \vdash t \ [\![E' \wedge E_?]\!]} \quad \cfrac{\cfrac{\Pi_?}{T \vdash u \ [\![E' \wedge E_? \wedge \bigwedge_{l \geq k} x_{R,l,s} \neq 1 \wedge \bigwedge_{l < k} x_{R,l,s} = 1]\!]}}{T \vdash w \ [\![E' \wedge x_{R,k,s} = 1 \wedge x_{R,k,s} = 1 \wedge u = v \wedge \sigma_s \wedge E_?]\!]}\ \mathcal{P}}{T \vdash t \ [\![E' \wedge x_{R,k,s} = 1 \wedge E_?]\!]}\ \mathcal{W}}{\Pi''}$$

On peut alors conclure avec la première règle de la figure 12.7 page 145. Encore une fois, la distance avec Π diminue strictement.

L'algorithme proposé est donc complet.

12.3 Perspectives

Il est possible d'améliorer encore cet algorithme, notamment dans le cas d'un nombre non borné de sessions en adoptant des stratégies plus variées pour la recherche de preuve :

- chercher d'abord pour une session, puis deux sessions, puis trois, etc.,
- répartir de manière équitable les transitions de l'arbre de recherche ouvrant sur une nouvelle session de celles continuant une session (recherche en zigzag),
- traiter simultanément plusieurs branches en affectant des priorités, ce qui permet de mettre temporairement de côté une branche ayant ouvert un nombre excessif de sessions,
- regrouper des branches aboutissant à un séquent identique pour ne le résoudre qu'une seule fois. Les preuves devant être minimales, cela nécessite de poser des hypothèses supplémentaires sur les preuves communes à plusieurs branches. Ces hypothèses devraient réduire la complexité réelle de l'algorithme.

Parallèlement, il serait intéressant d'implanter un tel algorithme et de tester les différentess stratégies S possibles et déterminer de manière empirique laquelle est la plus adaptée, notamment dans le cas d'un nombre non fini de sessions.

Nous pensons que cet algorithme peut être raffiné afin d'obtenir un algorithme exponentiel quand F est de complexité polynomiale.

CHAPITRE 13

Conclusion et perspectives

Sommaire

13.1 Travail réalisé

Le modèle de protocoles cryptographiques présenté dans ce mémoire nous a permis de proposer un algorithme de décision dans le chapitre 11 page 129 pour une vaste classe de protocoles. Cet algorithme est correct et complet pour un nombre borné ou non borné de sessions. Il termine pour un nombre borné de sessions. Cet algorithme permet de retrouver des résultats connus mais aussi de nombreux autres résultats similaires jusqu'ici non étudiés et ceci avec un travail minimal puisque les propriétés à vérifier sont simples :

- Si le modèle comprend une théorie équationnelle, celle-ci transforme le système étudié à l'aide de la propriété des variants finis, comme indiqué au chapitre 6 page 51 ; nous devons nous restreindre au cas où la théorie équationnelle restante est vide.
- La propriété de localité de la section 9.2 page 73.
- Les propriétés purement syntaxiques des sections 9.3.1 page 76, 9.3.2 page 76 et 10.4 page 94.

Dans le chapitre 12 page 137, nous avons présenté une autre approche algorithmique permettant de rechercher les attaques à l'aide d'un algorithme déterministe. Ce dernier est correct et complet. Bien que n'étant pas de complexité optimale pour le cas borné de sessions, cet algorithme est plus intelligent que l'algorithme du chapitre 11 et permet des optimisations pratiques, qu'il conviendra de tester, rendant l'implémentation réalisable en limitant l'explosion exponentielle.

Ces deux algorithmes s'appuient sur le résultat théorique du chapitre 10 page 81. Il utilise le modèle de preuve présenté dans le chapitre 4 page 33. Ce modèle est suffisamment général pour englober une très vaste classe de protocoles cryptographiques dont ceux qui vont appel à des propriétés algébriques telles que l'associativité, la commutativité, les groupes abéliens, etc. Toutefois, le résultat théorique du chapitre 10 est déjà suffisamment complexe dans le cas d'une théorie équationnelle vide et donc des restrictions ont été nécessaires pour le démontrer.

Enfin, grâce à l'application du théorème principal dans le cadre des signatures en aveugle, nous obtenons un résultat de décision pour celui-ci dans le cas d'un nombre borné de sessions et un algorithme correct et complet, terminant en cas d'attaque, dans le cas d'un nombre non borné de sessions. Ce résultat est à ce jour inédit.

13.2 Perspectives

Le travail présenté dans ce mémoire peut être poursuivi de trois façons :
- améliorer les algorithmes en terme de complexité,
- généraliser le résultat principal,
- travailler sur l'implémentation pratique.

13.2.1 Complexité

Nous n'avons pas démontré la complexité des algorithmes dans les chapitres 11 et 12 dans le cas d'un nombre borné de sessions. Certaines parties de ces algorithmes sont déjà de complexité optimale (pour Dolev-Yao) mais d'autres posent quelques problèmes. Un premier travail consisterait donc à corriger les deux algorithmes proposés pour leur permettre d'atteindre la complexité optimale dans le cas de Dolev-Yao (et correspondre ainsi au résultat de [RT01]). Nous pensons que le travail à effectuer sur cette partie reste raisonnable.

13.2.2 Généraliser le résultat principal

De nombreuses restrictions sont faites sur le système étudié. Celles de la section 10.4 page 94 sont d'ordre purement pratiques et devraient pouvoir être levées en travaillant davantage sur le cas des décompositions dans le cas principal. Par exemple, l'ajout d'un symbole fonctionnel supplémentaire et d'une règle le rendant équivalent à celui qu'il remplace permettrait de lever la dernière restriction de la section 10.4.

Les limitations sur les règles de décomposition de la section 9.3.2 page 76 peuvent être assouplies en y ajoutant d'autres formes autorisées. En contre-partie, il convient de traiter, dans le théorème principal, l'impact de chacune des formes ajoutées au niveau de la dernière partie du cas des règles de décomposition.

De même, les restrictions de la section 9.3.1 page 76 sur les règles de composition peuvent être limitées. Cela passe de nouveau par une preuve du théorème principale plus complexe à manipuler. Le chapitre 9 page 71 fournit les outils nécessaires pour choisir ce que l'on peut ou non permettre dans le système de preuve : nous sommes en effet limité par les résultats d'indécidabilité de ce chapitre.

La limitation aux théories équationnelles vides est une limitation plus importante sur laquelle il conviendra de travailler plus en profondeur. L'inclusion de la théorie AC dans le résultat principal complexifie la preuve de celui-ci.

Outre les restrictions, il est possible de généraliser les règles de composition et de décomposition pour y ajouter des conditions à leur application. Une telle règle deviendrait :

$$\frac{T \vdash u_1 \ \llbracket E_1 \rrbracket \ldots T \vdash u_n \ \llbracket E_n \rrbracket}{T \vdash u \ \llbracket E_1 \wedge \ldots \wedge E_n \rrbracket} \text{si } C$$

Selon les formes acceptées pour C, cette extension peut simplement consister en une relecture attentive des preuves pour vérifier si son inclusion ne perturbe pas celle-ci ou à de larges modifications. A priori, si la condition C est invariante par instanciation par des substitutions vérifiant les contraintes, elle ne devrait poser aucun aucun problème pour les règles de décomposition. Par contre, s'il s'agit de l'appliquer sur une règle de composition, on perd la simplicité de celle-ci (ajout de symbole en tête) et cela nous oblige à traiter celle-ci d'une manière similaire aux règles de décomposition, au moins en ce qui concerne les non linéarités introduites par C (comme $u_1 = u_2$).

13.2.3 Implémentation pratique

Le second point sur lequel il est possible de travailler est l'algorithme pratique du chapitre 12 page 137. Le besoin le plus immédiat est d'implanter celui-ci pour tester son efficacité pratique. L'implémentation permettra ensuite de détecter les points à optimiser. Par exemple, les règles de décomposition semblent jouer un poids important dans l'efficacité d'un algorithme de recherche en arrière car il convient alors de deviner des termes en entier.

Parallèlement, il est possible de chercher à raffiner cet algorithme pour lui permettre d'atteindre la complexité optimale, tout en conservant le principe de recherche en arrière, en évitant d'y inclure des parties non déterministes et en gardant l'extension possible à un nombre non borné de sessions. Le chapitre 12 présente déjà des pistes à ce sujet.

Bibliographie

[ABB$^+$05] Alessandro Armando, David Basin, Yohan Boichut, Yannick Chevalier, Luca Compagna, Jorge Cuéllar, Paul Hankes Drielsma, Pierre-Cyrille Heám, Olga Kouchnarenko, Jacopo Mantovani, Sebastian Mödersheim, Michael Rusinowitch, Judson Santiago, Mathieu Turuani, Lucas Viganò, and Laurent Vigneron. The avispa tool for the automated validation of internet security protocols and applications. In Kousha Etessami and Sriram K. Rajamani, editors, *CAV*, volume 3576 of *Lecture Notes in Computer Science*, pages 281–285. Springer, 2005.

[AFG00] Martín Abadi, Cédric Fournet, and Georges Gonthier. Authentication primitives and theur compilation. In *In Proceedings of the 27th ACM Symposium on Principles of Programming Languages*, pages 302–315, January 2000.

[AG97] Martín Abadi and Andrew D. Gordon. A calculus for cryptographic protocols : The spi calculus. In *Fourth ACM Conference on Computer and Communications Security*, pages 36–47. ACM Press, 1997.

[Air] Aircrack, a 802.11 sniffer and wep key cracker. `http://www.cr0.net:8040/code/network/`.

[AT91] Martín Abadi and Mark R. Tuttle. A semantics for a logic of authentication. In *In Proceedings of the 10th ACM Symposium on Principles of Distributed Computing*, pages 201–216, August 1991.

[BAF05] Bruno Blanchet, Martín Abadi, and Cédric Fournet. Automated Verification of Selected Equivalences for Security Protocols. In *20th IEEE Symposium on Logic in Computer Science (LICS 2005)*, pages 331–340, Chicago, IL, June 2005. IEEE Computer Society.

[Bau05] Mathieu Baudet. Random polynomial-time attacks and Dolev-Yao models. *Journal of Automata, Languages and Combinatorics*, 2005.

[BCLM05] Stefano Bistarelli, Iliano Cervesato, Gabriele Lenzini, and Fabio Martinelli. Relating Multiset Rewriting and Process Algebras for Security Protocol Analysis. *Journal of Computer Security*, 13(1) :3–47, February 2005.

[Ber03] Vincent Bernat. Towards a logic for verification of security protocols. In *Security Protocol Verification*, Marseille, September 2003.

[BGW01] Nikita Borisov, Ian Goldberg, and David Wagner. Intercepting mobile communications : The insecurity of 802.11. `http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html`, 2001.

[Bla01] Bruno Blanchet. An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. In *14th IEEE Computer Security Foundations Workshop (CSFW-*

14), pages 82–96, Cape Breton, Nova Scotia, Canada, June 2001. IEEE Computer Society.

[Bla02] Bruno Blanchet. From Secrecy to Authenticity in Security Protocols. In Manuel Hermenegildo and Germán Puebla, editors, *9th International Static Analysis Symposium (SAS'02)*, volume 2477 of *Lecture Notes on Computer Science*, pages 342–359, Madrid, Spain, September 2002. Springer Verlag.

[Bla04] Cédric Blancher. La sécurité des réseaux 802.11 : quoi de neuf depuis un an ? *MISC 12*, 12(12), Mars 2004.

[Bla05] Bruno Blanchet. Security Protocols : From Linear to Classical Logic by Abstract Interpretation. *Information Processing Letters*, 95(5) :473–479, September 2005.

[BLP02] Liana Bozga, Yassine Lakhnech, and Michael Périn. Outil de vérification HERMES. Rapport numéro 6 du projet RNTL EVA, May 2002.

[BN98] Franz Baader and Tobias Nipkow. *Term rewriting and all that.* Cambridge University Press, New York, NY, USA, 1998.

[BP03] Bruno Blanchet and Andreas Podelski. Verification of cryptographic protocols : Tagging enforces termination. In Andrew D. Gordon, editor, *Foundation of Software Science and Computation Structures (FOSSACS)*, volume 2620 of *Lecture Notes in Computer Science*, pages 136–152, Warsaw, Poland, April 2003. Springer Verlag.

[CDL+99] Iliano Cervesato, Nancy Durgin, Patrick Lincoln, John C. Mitchell, and A. Scedrov. A meta-notation for protocol analysis. In *CSFW*, pages 55–69, 1999.

[CDL+03] Iliano Cervesato, Nancy Durgin, Patrick Lincoln, John C. Mitchell, and A. Scedrov. A Comparison between Strand Spaces and Multiset Rewriting for Security Protocol Analysis. In M. Okada, B. Pierce, A. Scedrov, H. Tokuda, and A. Yonezawa, editors, *Software Security - Theories and Systems — ISSS 2002*, pages 356–383, Tokyo, Japan, 8–10 November 2003. Springer-Verlag LNCS 2609.

[CDL06] Véronique Cortier, Stéphanie Delaune, and Pascal Lafourcade. A survey of algebraic properties used in cryptographic protocols. *Journal of Computer Security*, 14(1) :1–43, 2006.

[Cha82] David Chaum. Blind signatures for untraceable payments. In *CRYPTO*, pages 199–203, 1982.

[CJ97] J. Clark and J. Jacob. A survey of authentication protocol literature : Version 1.0, 1997.

[CKR+03] Yannick Chevalier, Ralf Küsters, Michael Rusinowitch, Mathieu Turuani, and Laurent Vigneron. Extending the dolev-yao intruder for analyzing an unbounded number of sessions. In Matthias Baaz and Johann A. Makowsky, editors, *CSL*, volume 2803 of *Lecture Notes in Computer Science*, pages 128–141. Springer, 2003.

[CKRT03a] Yannick Chevalier, Ralf Küsters, Michael Rusinowitch, and Mathieu Turuani. Deciding the security of protocols with Diffie-Hellman exponentiation and products in exponents. In J. Radhakrishnan and P.K. Pandya, editors, *Proc.*

FST/TCS, Mumbai, volume 2914 of *Lecture Notes in Computer Science*, 2003.

[CKRT03b] Yannick Chevalier, Ralf Küsters, Michael Rusinowitch, and Mathieu Turuani. An NP decision procedure for protocol insecurity with xor. In Kolaitis [Kol03].

[CKRT05] Yannick Chevalier, Ralf Küsters, Michael Rusinowitch, and Mathieu Turuani. Deciding the Security of Protocols with Commuting Public Key Encryption. *Electronic Notes in Theoretical Computer Science*, 125(1) :55–66, 2005.

[CKS01] R. Chadha, M. Kanovich, and A. Scedrov. Inductive methods and contract-signing protocols. In *Proceedings of the 8th ACM conference on Computer and Communications Security*, pages 176–185. ACM Press, 2001.

[CL04] Hubert Comon-Lundh. Intruder theories (ongoing work). In Igor Walukiewicz, editor, *Proceedings of the 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, March 2004. Springer. Invited talk.

[CLC03a] Hubert Comon-Lundh and Véronique Cortier. New decidability results for fragments of first-order logic and application to cryptographic protocols. In Robert Nieuwenhuis, editor, *RTA*, volume 2706 of *Lecture Notes in Computer Science*, pages 148–164. Springer, 2003.

[CLC03b] Hubert Comon-Lundh and Véronique Cortier. Security properties : two agents are sufficient. In *Proceedings of the 12th European Symposium On Programming (ESOP'03)*, volume 2618 of *Lecture Notes in Computer Science*, pages 99–113, Warsaw, Poland, April 2003. Springer Verlag.

[CLD05] Hubert Comon-Lundh and Stéphanie Delaune. The finite variant property : How to get rid of some algebraic properties. In Jürgen Giesl, editor, *Proceedings of the 16th International Conference on Rewriting Techniques and Applications (RTA'05)*, volume 3467 of *Lecture Notes in Computer Science*, pages 294–307, Nara, Japan, April 2005. Springer.

[CLS03] Hubert Comon-Lundh and Vitaly Shmatikov. Intruder deductions, constraint solving and insecurity decision in presence of exclusive or. In Kolaitis [Kol03].

[CLT03] Hubert Comon-Lundh and Ralf Treinen. Easy intruder deductions. In Nachum Dershowitz, editor, *Verification : Theory and 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, February 2003. Invited paper.

[CM97] Christian Cachin and Ueli M. Maurer. Unconditional security against memory-bounded adversaries. *Lecture Notes in Computer Science*, 1294 :292–??, 1997.

[CMR01] Véronique Cortier, Jonathan K. Millen, and Harald Ruess. Proving secrecy is easy enough. In *14th IEEE Computer Security Foundations Workshop*, pages 97–108. IEEE Computer Society, 2001.

[Cor02] Véronique Cortier. Outil de vérification SECURIFY. Rapport numéro 7 du projet RNTL EVA, May 2002.

[Cor03] Véronique Cortier. *Vérification automatique des protocoles cryptographiques*. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, March 2003.

[CR05] Yannick Chevalier and Michael Rusinowitch. Combining intruder theories. Technical report, INRIA Lorraine, February 2005.

[CRZ05] Véronique Cortier, Michael Rusinowitch, and Eugen Zalinescu. A resolution strategy for verifying cryptographic protocols with cbc encryption and blind signatures. In Pedro Barahona and Amy Felty, editors, *PPDP*, pages 12–22. ACM, 2005.

[CV01] Yannick Chevalier and Laurent Vigneron. A tool for lazy verification of security protocols. Technical Report A01RR-140, LORIA, 2001.

[CWHWW03] Nancy Cam-Winget, Russ Housley, David Wagner, and Jesse Walker. Security flaws in 802.11 data link protocols. *Commun. ACM*, 46(5) :35–39, 2003.

[CWMSW04] Nancy Cam-Winget, Tim Moore, Dorothy Stanley, and Jesse Walker. IEEE 802.11i overview. Technical report, IEEE, 2004.

[DJ90] Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 243–309. NH, 1990.

[DLMS99] Nancy Durgin, Patrick Lincoln, John C. Mitchell, and A. Scedrov. Undecidability of bounded security protocols. In N. Heintze and E. Clarke, editors, *Workshop on Formal Methods and Security Protocols*, Italy, July 1999.

[DM00] G. Denker and Jonathan K. Millen. Capsl integrated protocol environment. In *DARPA Information Survivability Conference (DISCEX 2000)*, pages 207–221. IEEE Computer Society, 2000.

[DY81] D. Dolev and A.C. Yao. On the security of public key protocols. In *Proc. IEEE Symp. on Foundations of Computer Science*, pages 350–357, 1981.

[FMS01] Scott R. Fluhrer, Itsik Mantin, and Adi Shamir. Weaknesses in the key scheduling algorithm of RC4. *Lecture Notes in Computer Science*, 2259 :1–24, 2001.

[FOO93] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A practical secret voting scheme for large scale elections. In *ASIACRYPT '92 : Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques*, pages 244–251, London, UK, 1993. Springer-Verlag.

[GP05] Jean Goubault-Larrecq and Fabrice Parrennes. Cryptographic protocol analysis on real C code. In Radhia Cousot, editor, *Proceedings of the 6th International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI'05)*, volume 3385 of *Lecture Notes in Computer Science*, pages 363–379, Paris, France, January 2005. Springer.

[HA03] Russ Housley and William Arbaugh. Security problems in 802.11-based networks. *Commun. ACM*, 46(5) :31–34, 2003.

[HPW96] James Harland, David J. Pym, and Michael Winikoff. Programming in lygon : An overview. In Martin Wirsing and Maurice Nivat, editors, *AMAST*, volume 1101 of *Lecture Notes in Computer Science*, pages 391–405. Springer, 1996.

[IEE02] IEEE. *Part 11 : Wireless LAN Medium Access Control and Physical Layer specifications : Higher-Speed Physical Layer Extension in the 2.4 GHz Band*, supplement to ieee standard for information technology – telecommunications and information exchange between systems – local and metropolitan area networks – specific requirements edition, 2002. `http://standards.ieee.org/reading/ieee/std/lanman/802.11b-1999`.

[Kol03] P. Kolaitis, editor. *Eighteenth Annual IEEE Symposium on Logic in Computer Science*, Ottawa, Canada, June 2003. IEEE Computer Society.

[KR05] Steve Kremer and Mark Ryan. Analysis of an electronic voting protocol in the applied pi-calculus. In Mooly 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.

[KSRW04] Tadayoshi Kohno, Adam Stubblefield, Aviel D. Rubin, and Dan S. Wallach. Analysis of an electronic voting system. In *IEEE Symposium on Security and Privacy*, pages 27–. IEEE Computer Society, 2004.

[Low96] Gavin Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 1055, pages 147–166. Springer-Verlag, Berlin Germany, 1996.

[Low97a] Gavin Lowe. Casper : A compiler for the analysis of security protocols. In *PCSFW : Proceedings of The 10th Computer Security Foundations Workshop*. IEEE Computer Society Press, 1997.

[Low97b] Gavin Lowe. A hierarchy of authentication specifications. In *PCSFW : Proceedings of The 10th Computer Security Foundations Workshop*. IEEE Computer Society Press, 1997.

[McA93] David A. McAllester. Automatic recognition of tractability in inference relations. *Journal of the ACM*, 40(2) :284–303, 1993.

[MR00] Jonathan K. Millen and Harald Ruess. Protocol-independent secrecy. In *IEEE Symposium on Security and Privacy*, pages 110–209, 2000.

[MRH04] Vebjørn Moen, Håvard Raddum, and Kjell J. Hole. Weaknesses in the temporal key hash of wpa. *SIGMOBILE Mob. Comput. Commun. Rev.*, 8(2) :76–83, 2004.

[Oss04] Michael Ossmann. Wep : Dead again. In Security Focus, December 2004. `http://www.securityfocus.com/infocus/1814`.

[Pau98] Lawrence C. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6 :85–128, 1998.

[Pau99] Lawrence C. Paulson. Proving security protocols correct. In *LICS : Proceedings of The 14th Annual IEEE Symposium on Logic in Computer Science*, pages 370–383. IEEE Computer Society Press, July 1999.

[RS03] R. Ramanujam and S. P. Suresh. Tagging makes secrecy decidable with unbounded nonces as well. In Paritosh K. Pandya and Jaikumar Radhakrishnan, editors, *FSTTCS*, volume 2914 of *Lecture Notes in Computer Science*, pages 363–374. Springer, 2003.

[RT01] Michael Rusinowitch and Mathieu Turuani. Protocol insecurity with finite number of sessions is np-complete. In *14th IEEE Computer Security Foundations Workshop*, pages 174–190, Cape Breton, Nova Scotia, June 2001.

[RV01] John Alan Robinson and Andrei Voronkov, editors. *Handbook of Automated Reasoning (in 2 volumes).* Elsevier and MIT Press, 2001.

[Sch93] Bruce Schneier. *Applied Cryptography : Protocols, Algorithms, and Source Code in C.* John Wiley & Sons, Inc., New York, NY, USA, 1993.

[Sch97] Steve Schneider. Verifying authentication protocols with CSP. In *PCSFW : Proceedings of The 10th Computer Security Foundations Workshop.* IEEE Computer Society Press, 1997.

[SM01] Vitaly Shmatikov and John C. Mitchell. Analysis of abuse-free contract signing. *Lecture Notes in Computer Science*, 1962, 2001.

[SPO] SPORE. Security protocols open repository. `http://www.lsv.ens-cachan.fr/spore/`.

[SS96] Steve Schneider and Abraham Sidiropoulos. CSP and anonymity. In *ESORICS*, pages 198–218, 1996.

[Tur03] Mathieu Turuani. *Sécurité des protocoles cryptographiques : décidabilité et complexité.* PhD thesis, Université Henri Poincaré - Nancy 1, December 2003.

Index

V