Titre: Title:	Analyse formelle des protocoles cryptographiques et flux d'information admissible
Auteur: Author:	Sardaouna Hamadou
Date:	2008
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Hamadou, S. (2008). Analyse formelle des protocoles cryptographiques et flux d'information admissible [Thèse de doctorat, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/8154/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/8154/
Directeurs de recherche: Advisors:	John Mullins, & Srecko Brlek
Programme: Program:	Non spécifié

UNIVERSITÉ DE MONTRÉAL

ANALYSE FORMELLE DES PROTOCOLES CRYPTOGRAPHIQUES ET FLUX D'INFORMATION ADMISSIBLE

SARDAOUNA HAMADOU
DÉPARTEMENT DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR
(GÉNIE INFORMATIQUE)
MARS 2008

Your file Votre référence
ISBN: 978-0-494-46104-4
Our file Notre référence
ISBN: 978-0-494-46104-4

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée:

ANALYSE FORMELLE DES PROTOCOLES CRYPTOGRAPHIQUES ET FLUX D'INFORMATION ADMISSIBLE

présentée par: HAMADOU Sardaouna

en vue de l'obtention du diplôme de: Philosophiæ Doctor

a été dûment acceptée par le jury d'examen constitué de:

M. FERNANDEZ José M., Ing., Ph.D., président

M. MULLINS John, Ph.D., membre et directeur de recherche

M. BRLEK Srečko, Ph.D., membre et codirecteur de recherche

M. MERLO Ettore, Ph.D., membre

Mme. PALAMIDESSI Catuscia, Ph.D., membre

REMERCIEMENTS

C'est un grand plaisir pour moi de mettre fin à ce projet et surtout de remercier tous les gens qui m'ont prêté main forte dans l'accomplissement de ce défi. Mes premiers remerciements vont naturellement aux Professeurs John Mullins et Srečko Brlek, respectivement directeur et co-directeur de cette thèse. D'abord à John Mullins pour m'avoir fais confiance et pour les heures de travail très stimulantes et surtout si enrichissantes. Je me souviendrai longtemps de ta passion inépuisable pour la recherche et merci de m'avoir poussé à me surpasser quand il le fallait. Ensuite à Srečko Brlek avec qui j'ai tissé des liens d'amitié et qui m'a accompagné à travers les hauts et les bas de ma recherche. Tu as su bien m'écouter et me conseiller aux moments opportuns. Tu continueras à m'étonner par ton optimisme indéfectible et ta capacité de tirer le meilleur de tout le monde. Merci pour ton coté humain si chaleureux.

Je tiens à remercier très chaleureusement tous les membres du jury. Leur lecture minutieuse de la thèse ainsi que leurs commentaires et conseils ont contribué grandement à améliorer la qualité de cette thèse. José Fernandez me fait l'honneur d'être président du jury, je lui adresse mes plus vifs remerciements. Je témoigne toute ma gratitude à Catuscia Palamidessi de s'être attelé à la lourde tâche de rapporteur. *Grazie molto* pour les échanges scientifiques qui ont eu lieu ces deux dernières années ; merci à Ettore Merlo d'avoir également accepté d'être rapporteur.

J'adresse mes sincères remerciements à Milan Brlek-Bergeron qui a généreusement accepté de relire toute la thèse. Il a énormément contribué à la qualité du texte. Merci également à la Fondation Polytechnique et la FQRNT pour avoir financé cette thèse. Je souhaite aussi remercier les personnes suivantes qui se sont succédé au fil des années et avec qui le travail fut fort agréable : Stéphane Lafrance, Geneviève Bastien, Allaaedine Fellah, Hind Rakkay, Olfa Ben Sik Ali, Fayçal Abouzaïd, Raveca Oarga,

Damien Azembre, Mathieu Bergeron et Rachid Hadjidj. Je n'oublie certainement pas tous les membres du Laboratoire de Combinatoire et d'Informatique Mathématique (LaCIM) de l'université du Québec à Montréal qui font de ce laboratoire un cadre de travail idéal et surtout un lieu agréable.

Au plan personnel, je dois d'abord remercier mes parents, à qui je dédie cette thèse, et qui y sont certainement pour beaucoup dans tout ça. Gros merci à mes nombreux frères et soeurs en Afrique et en Europe qui, malgré la distance, ont su m'entourer de toute leur affection. Ensuite à mes cousins Mahfouze, Aboubakar, Ali et Nour et mes amis Nathalie, Kindy, Olivia, Michelle, Marie, Janvier Ibrahim, Eric, Mahmoud, Ted et Martial, merci pour les moments de joie intense si nombreux. Pour tous ceux que je n'ai pas mentionnés par nom, soyez certains que je ne vous ai pas oubliés. A tous un grand Merci.

Enfin, merci Véronique pour ton merveilleux sourire :)

RÉSUMÉ

Le domaine de la modélisation et de la vérification des systèmes de sécurité de l'information a connu une telle explosion dans les années 1990 que, maintenant, on dispose de toute une gamme de modèles et de méthodes. Parmi ces théories, celle du flux d'information est très naturelle : il n'y a pas de flux d'information si la réalisabilité d'une condition observable au niveau public n'est corrélée à la réalisabilité d'aucune condition observable uniquement au niveau confidentiel. Cette contrainte est toutefois trop forte dans le cas où une interférence inévitable, à cause de la nature de l'application, doit être admise. Dans un tel cas, la propriété caractérisant la sécurité d'un tel système doit prendre en compte le contexte de cette déclassification admissible. Notre recherche dans cette thèse s'est donc tournée vers l'étude de aspects formels de l'analyse d'une importante catégorie des systèmes de sécurité à flux d'information admissible : les protocoles cryptographiques.

Notre contribution comporte plusieurs volets. Nous avons d'abord proposé de nouvelles caractérisations de l'interférence admissible et avons montré que presque tous les requis de sécurité de ces protocoles sont des instances de l'interférence admissible e.g. la confidentialité, l'authentification, la non-répudiation [E,A,B], ainsi que l'anonymat des clients envers les marchands dans une extension que nous avons proposé au protocole de e-commerce SET de Visa et MasterCard [A,B]. Notre méthode à été implémentée dans un prototype d'analyseur symbolique des protocoles cryptographiques ASPiC [E] qui nous a permis de détecter une faille de sécurité dans le protocole SET [C,D]. Le fait que l'article [C] tienne dans la revue "Information Processing Letters", le premier rang au palmarès des articles les plus téléchargés dès sa parution en début de l'année 2006, et continue de figurer dans le top 25 de ce palmarès presque deux ans après sa parution, témoigne de l'intérêt que des telles approches suscitent au sein de la communauté scientifique internationale.

Toutefois, notre méthode ainsi que la plupart des méthodes formelles développées à ce jour, dans le domaine de la sécurité de l'information, sont basées, d'une part, sur l'approche dite "possibiliste", dans laquelle le non-déterminisme sert à modéliser les mécanismes aléatoires de génération de tous les comportements possibles des systèmes analysés et, d'autre part, sur l'hypothèse de cryptographie parfaite qui stipule que les primitives cryptographiques sont des boites noires. Ces approches sont trop grossières pour décrire les fuites d'information probabilistes et ainsi prévenir les attaques basées sur les distributions probabilistes des évènements observables du système.

Aussi, nous avons étendu notre modèle avec des primitives cryptographiques probabilistes polynomiales et une sémantique probabiliste [F,G,H]. Comme il est d'usage d'utiliser des ordonnanceurs pour résoudre le non-déterminisme dans les modèles probabilistes, et que le réseau de communication est généralement considéré sous le contrôle de l'attaquant, l'ordonnancement est alors contrôlé par ce dernier. Ainsi, ils doivent être soigneusement définis pour refléter autant que possible la capacité de l'intrus à contrôler le réseau de communication, sans pour autant contrôler la réaction interne du système avec lequel il interagit. Nous avons muni notre sémantique de tels ordonnanceurs et avons reformulé l'équivalence observation de Mitchell *et al.*, 2006 sous une forme plus pratique, prenant en compte toute trace observable, au lieu d'un observable à la fois. Cependant, comme la plupart des problèmes d'équivalence checking, nos équivalences asymptotiques souffrent du problème de quantification sur tous les intrus possibles. Aussi, pour contourner cette dernière difficulté, nous avons défini une sémantique contextuelle pour notre modèle [I]. Nous y avons reformulé nos équivalences asymptotiques et avons prouvé que, contrairement à la sémantique concrète, elles ne sont pas équivalentes : l'équivalence observationnelle étant moins fine, justifiant le passage à l'équivalence de trace. Nous avons également défini une bisimulation qui raffine les équivalences asymptotiques. Enfin nous avons illustré l'utilité de notre approche par deux importantes études de cas : l'analyse du protocole *Dînner des cryptographes* de David Chaum et le *"Crowds protocol"* de Reiter et Ruben.

LISTE DES PUBLICATIONS

A. S. Brlek, **S. Hamadou**, J. Mullins, ASET, un protocole anonyme et sécuritaire pour les transactions électroniques, *Actes de SAR'2003, 2-ème rencontre francophone sécurité et Architecture Réseaux* (30 juin - 4 juillet 2003, Nancy, France) (2003) 1–15

B. S. Brlek, **S. Hamadou**, J. Mullins, Anonymous and Secure Electronic Transaction Protocol, *Annals of Telecommunications* 60 no. 5-6 (2005) 1–28

C. S. Brlek, **S. Hamadou**, J. Mulllins, A Flaw in the Electronic Commerce Protocol SET, *Information Processing Letters*, 97 no.3 (2006) 104–108

D. S. Brlek, **S. Hamadou**, J. Mullins, Some Remarks on the Certificates Registration of the Electronic Commerce Protocol SET, *Proc. ICIW06, International Conference on Internet and Web Applications and Services* (February 23-25, 2006, Guadeloupe) Actes électroniques (2006)

E. G. Bastien, **S. Hamadou**, J. Mullins, *ASPiC : a Tool for Symbolic Analysis of Crypto-Protocols Based on Interference Checking*, Annals of Telecommunications (2006) 1-30 (submitted for publication).

F. S. Brlek, **S. Hamadou**, J. Mullins, A probabilistic scheduler for the analysis of cryptographic protocols, *5th International Workshop on Security Issues in Concurrency (SecCo'07)* (September 3rd, 2007, Lisboa, Portugal) (2007) 53–67

G. S. Brlek, **S. Hamadou**, J. Mullins, A probabilistic scheduler for the analysis of cryptographic protocols, *Electronic Notes in Theoretical Computer Science* Volume 194, Issue 1, (2007) 61–83 (this paper superseded [F.])

H. S. Brlek, **S. Hamadou**, J. Mullins, Calibrating the power of schedulers for probabilistic systems, *Journal of Computer Security*, (2008) 1-32 (submitted for publication).

I. S. Brlek, **S. Hamadou**, J. Mullins, Context sensitive semantics for probabilistic models, *Theoretical Computer Science* (2008) 1-34 (submitted for publication).

ABSTRACT

The wide spread of distributed systems, where resources and data are shared among users located almost everywhere in the world, has enormously increased the interest in security issues. It is important to precisely define security properties in order to have formal statements of the correctness of a security mechanism. As a consequence, in recent years, many formal definitions of security properties have been proposed. Among these theories, the information flow approach which aims at characterizing the complete absence of any causal flow from high level entities to low level ones, is very natural. However this requirement is too strong. Indeed, complete absence of any information flow can hardly be achieved in real systems. In order to deal with real applications, it is often necessary to admit mechanisms for downgrading or declassifying information. Also, in this thesis we deal with a particular class of admissible information flow security systems, namely *cryptographic protocols.*

Our contribution concerns several issues. We have first, proposed new characterizations of admissible information flow security properties and proved that they are general enough to capture most of security properties of cryptographic protocols. Indeed security properties such as confidentialy, authentication, non-repudiation [E,A,B] as well as the anonymity of customers in a recent extension of Visa and MasterCard e-commerce security payment protocol SET that we proposed [A,B], are some instances of admissible information flow security properties. Our approach is mechanized in a tool for symbolic analysis of crypto-protocols based on interference checking, called ASPiC [E]. This tool allowed us, in particular, to detect a new security flaw in the industrial e-commerce SET protocol [C,D]. The fact that the article [C] was rated number one in the TOP25 Hottest Articles - downloaded during January-March 2006- within the journal IPL, and continues to appear in this prize list almost two years after its publication, illustrates the importance of such formal approaches for the international scientific community.

However, our calculus and most of the formal approaches to analysis of cryptographic protocols are based on the possibilistic approach for modelling behaviors resulting from interactions between the protocol's agents and the perfect cryptography assumption for modelling the intruder. In the possibilistic approach, non-determinism is used as a mechanism for the random generation of all possible behaviors while the perfect cryptography assumption considers the encryption scheme as black boxes and assumes that an intruder cannot learn anything from a cypher text if she does not know the key. The possibilistic approach is not expressive enough to model probabilistic behaviors and so, to prevent attacks based on probabilistic distribution able to infer a secret from its observations.

Also, we proposed a probabilistic extension, *à la* Mitchell *et al.*, 2006, of our model, extended with probabilistic polynomial time cryptographic primitives and probabilistic semantics. But when modelling cryto-protocols by means of process calculi which express both nondeterministic and probabilistic behavior, it is customary to view the scheduler as an intruder. It has been established that the traditional scheduler needs to be carefully calibrated in order to more accurately reflect the intruder's capabilities for controlling communication channels. We have proposed such a class of schedulers through our probabilistic semantic extention [F,G,H]. Along these lines, we have defined a new characterization of Mitchell *et al.*'s asymptotic observational equivalence more suited for taking into account any observable traces instead of just a single observable as required in the analysis of many security protocols. However, as most contextual equivalences for cryptographic process calculi, our asymptotic equivalences suffer from quantification over all possible intruders. To overcome this problem, we have defined a context-sensitive semantics [I] allowing an implicit representation of the intruder. We have also defined a probabilistic bisimulation and showed that it refines the asymptotic equivalences. Finally, to illustrate the usefullness of our approach, we have proposed extensive case studies, the analysis of the Chaum's *Dining Cryptographers* protocol and the Ruben and Reiter's *Crowds protocol.*

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

Protocoles et Crypto-systèmes

ASET	*Anonymous and Secure Electronic Transaction*
DCP	*Dining Cryptographers Protocol*
SET	*Secure Electronic Transaction*
WMF	*Wide Mouthed Frog*
AES	*Advanced Encryption Standard*
DES	*Data Encryption Standard*
RSA	*Chiffrement à clé publique de Rivest, Shamir et Adleman*
OTP	*Chiffrement à clé jetable One-Time Pad*

Algèbres de processus

CCS	*Calculus of Communicating Systems*
CSP	*Communicating Sequential Processes*
CryptoSPA	*Cryptographic Security Process Algebra*
CSPAD	*Cryptographic Security Process Algebra with Downgrading*
PPC	*Probabilistic Poly-time Calculus*
ProSPA	*Probabilistic Security Process Algebra*
SPA	*Security Process Algebra*
SPPA	*Security Protocols Process Algebra*

Non-Interférence

NDC	*NonDeducibility on Composition*
BNDC	*Bisimulation-based NonDeducibility on Composition*
GNDC	*Generalized NonDeducibility on Composition*
SNNI	*Strong Non-deterministic Non-Interference*
BSNNI	*Bisimulation-based Strong Non-deterministic Non-Interference*

Interférence admissible

NAI	*Non-deterministic Admissible Interference*
SNAI	*Strong Non-deterministic Admissible Interference*
BNAI	*Bisimulation-based Non-deterministic Admissible Interference*
SBNAI	*Strong Bisimulation-based Non-deterministic Admissible Interference*
NDCIA	*NonDeducibility on Composition Admissible Interference*
SNDCIA	*Strong NonDeducibility on Composition Admissible Interference*
BNDCIA	*Bisimulation-based NonDeducibility on Composition Admissible Interference*
SBNDCIA	*Strong Bisimulation-based NonDeducibility on Composition Admissible Interference*

Autres acronymes

CRAC	*Laboratoire de Conception et Réalisation des Applications Complexes*
EOO	*Evidence Of Origin*
EOR	*Evidence Of Reception*
NRO	*Non-Repudiation of Origin*
NRR	*Non-Repudiation of Reception*
NSA	*National Security Agency*

Notations mathématiques

2^E	l'ensemble des sous ensembles de E
$[e]_\sim$	classe d'équivalence de l'élément e induite par la relation $\sim$
E^n	l'ensemble des tuples à n dimensions sur l'ensemble E
$\|E\|$	cardinal de l'ensemble E
$\mathbb{N}(\mathbb{N}^+)$	l'ensemble des entiers naturels (strictement positifs)
$\mathbb{R}(\mathbb{R}^+)$	l'ensemble des nombres réels (positifs ou nuls)
$\sum$	la somme
$\prod$	le produit

Modèle possibiliste CSPAD

$\mathcal{V}$	ensemble des variables.
$\mathcal{T}$	ensemble des termes.
$fv(t)$	ensemble des variables dans le terme t.
$\mathcal{M}$	ensemble des messages, i.e. des termes clos.
$\mathcal{K}$	ensemble des clés de chiffrement.
$\{m\}_K$	cryptogramme obtenu en chiffrant le message m par la clé K.
$h(m)$	condensé (haché) du message m.
$[m]_X$	signature du message m par le principal X.
ϕ, ψ	base de connaissance de l'environnement hostile (i.e. des attaquants).
$\phi \vdash m$	le message m est déductible de l'ensemble des messages ϕ.
$\mathcal{D}(\phi)$	ensemble des messages déductibles de ϕ.
$m \prec m'$	le message m' contient m.
$m \prec^{\phi}_{clear} m'$	le message m' contient en clair, relativement à ϕ, le message m.
$\mathcal{S}$	ensemble des messages secrets.
$I_\phi(\mathcal{S})$	l'idéal de $\mathcal{S}$ dans l'environnement ϕ.
$\mathcal{P}roc$	ensemble des processus clos.
$fv(P)$	ensemble des variables libre du processus P.
$\mathcal{D}(P)$	ensemble des processus dérivables du processus P.

$\mathcal{D}_\phi(P)$	ensemble des processus dérivables de P dans un environnement de connaissance ϕ.
$\phi \rhd P$	une configuration, i.e un processus évoluant dans un environnement hostile.
$\mathcal{D}(\phi \rhd P)$	ensemble des configurations dérivables de $\phi \rhd P$.
$\mathcal{C}onf$	ensemble des configurations.
$\mathcal{E}nemy$	ensemble des processus ennemis.
$\mathcal{E}nemy_\phi$	ensemble des processus ennemis de base de connaissance ϕ.
$ID(\Pi)$	ensemble des messages apparaissant syntaxiquement dans le processus Π.
$Top(\phi)$	le plus grand attaquant de connaissance ϕ.
$\mathcal{C}$	ensemble des canaux publics.
$\mathrm{dom}(c)$	domaine du canal c.
$\mathcal{A}ct$	ensemble des actions.
$\mathcal{V}is$	ensemble des actions visibles.
H, D, L	ensembles des actions de haut niveau, de déclassification, et de bas niveau.
$\mathcal{P}roc_H(\mathcal{P}roc_L)$	ensemble des processus de haut niveau (de bas niveau).
γ	fonction de décoration.
$\gamma_\phi^{\mathcal{S}}$	fonction de déclassification pour la confidentialité des messages $\mathcal{S}$ dans l'environnement ϕ.
$\mathrm{Spec}_{\mathcal{S}}$	fonction de décoration de bas niveau pour la confidentialité des messages $\mathcal{S}$.
$\gamma_{Integ}^{M(A,B)}$	fonction de décoration pour l'intégrité du message M du principal A vers le principal B.
$\gamma_{NRO}^{M(A,B)}$	fonction de décoration pour la non-répudiation d'origine du message M du principal A vers le principal B.
$\gamma_{NRR}^{M(A,B)}$	fonction de décoration pour la non-répudiation de reception du message M du principal A vers le principal B.

Modèle probabiliste ProSPA

$\mathbf{N}$	paramètre de sécurité.

λ	fonction probabiliste polynomiale.
$\lambda(m) \hookrightarrow x$	affectation de la valeur $\lambda(m)$ à la variable x.
$\lambda(m) \overset{p}{\hookrightarrow} m'$	$\lambda(m)$ s'évalue à m' avec la probabilité p.
$Im(\lambda(m))$	ensemble image de $\lambda(m)$, i.e. $\{m' \mid \exists p \in]0..1]\ \lambda(m) \overset{p}{\hookrightarrow} m'\}$.
Λ	ensemble des fonctions probabilistes polynomiales.
$bw(c)$	bande passante du canal c.
$q : \mathbb{N} \rightarrow \mathbb{R}^+$	polynôme positif.
$\mathcal{P}oly$	ensemble des polynômes positifs.
$\text{Prob}[f(a) = b]$	probabilité que $f(a)$ s'évalue à b.
$\mathcal{D}_{poly(\mathbf{N})}(\phi)$	ensemble des messages déductibles de ϕ en temps polynomial.
$\mathcal{P}artial$	ensemble des actions partielles.
$\mathcal{A}ctual$	ensemble des actions de communication.
$\mathcal{B}locked$	ensemble des processus bloqués.
$\mathcal{E}$	un ensemble des processus clos (ou des configurations).
$o = (c, m)$	un observable.
$\mathcal{O}$	ensemble des observables.
π	une permutation.
$\text{Perm}(E)$	ensemble des permutations des éléments de E.
S	ordonnanceur ou stratégie d'attaque.
$S = (F, \zeta)$	un ordonnanceur dans la sémantique contextuelle de *ProSPA*.
(Π, S)	un attaquant dans la sémantique concrète de *ProSPA*.
(ϕ, S)	un attaquant dans la sémantique contextuelle de *ProSPA*.
$Sched$	ensemble des ordonnanceurs.
$Sched_\tau$	ensemble des ordonnanceurs donnant priorité aux actions internes.
$\mathcal{I}ndex$	ensemble des indices.
$\mathcal{A}ct \times Ind$	ensemble des actions indexées.
$\text{support}(\alpha)$	support de l'action α, i.e. le canal sur lequel l'action a eu lieu.
Υ	fonction de normalisation des probabilité. $\Upsilon(P, \alpha)$ compte le nombre de façons différentes dont P peut exécuter l'action α.

$\chi(P)$	localise les positions des actions exécutables de P.
$\overline{\chi}$	fonction ne localisant que les actions de communication.
$F(c,\phi)$	sélection d'un domaine de recherche pour le canal d'écoute c conformément à ϕ.
$\mathcal{F}$	ensemble des fonctions de sélection.
$[(\alpha,id)]_{I_1\times I_2}$	classe d'équiprobabilité de l'action indexée (α,id) (sémantique concrète).
$[\alpha]_\sim$	classe d'équiprobabilité de l'action α (sémantique contextuelle).

Systèmes de transitions

$(\mathcal{E},\mathscr{T},E_0)$	système de transitions d'origine E_0, de sommets $\mathcal{E}\subseteq \mathcal{P}roc$ et de transitions $\mathscr{T}\subseteq \mathcal{E}\times\mathcal{A}ct\times\mathcal{E}$.
$P\xrightarrow{\alpha}P'$	transition (P,α,P') du sommet P au sommet P' par l'action α.
$(\mathcal{E},\mathscr{T},E_0)$	système de transitions probabiliste d'origine E_0, de sommets $\mathcal{E}\subseteq \mathcal{P}roc$ et de transitions $\mathscr{T}\subseteq \mathcal{E}\times\mathcal{A}ct\times[0,1]\times\mathcal{E}$.
$P\xrightarrow{\alpha[p]}P'$	transition probabiliste (P,α,p,P') du sommet P au sommet P' par l'action α avec la probabilité p.
Exec(P)	ensemble des transitions sortantes de P.
σ	un chemin.
α-path	un α-chemin, i.e un chemin formé d'un nombre fin d'actions internes suivies par l'action α.
$P\stackrel{\sigma}{\Longrightarrow}P'$	chemin de P à Q.
$Tr(P)$	ensemble des traces visibles de P.
Prob$[S(\sigma)]$	probabilité que l'ordonnanceur S choisisse le chemin σ.
$Paths(P,\stackrel{\alpha}{\Longrightarrow},\mathscr{E})$	ensemble de tous les α-chemins minimaux de P vers un élément de $\mathscr{E}$.
$\mu(P,\stackrel{\hat{\alpha}}{\Longrightarrow}_S,\mathscr{E})$	probabilité cumulative du cône défini par $Paths(P,\stackrel{\alpha}{\Longrightarrow},\mathscr{E})$, selon l'ordonnanceur S.
$\mu(P,\stackrel{\hat{\alpha}}{\Longrightarrow}_{S/H},\mathscr{E})$	probabilité cumulative up to H.
Prob$[P\rightsquigarrow_S\alpha]$	la probabilité que P génère α selon l'ordonnanceur S.
Prob$[P\rightsquigarrow_S\Gamma]$	la probabilité que P génère des actions dans Γ selon l'ordonnanceur S.

$\text{Prob}[P \rightsquigarrow_S o]$	la probabilité que P génère l'observable o selon l'ordonnanceur S.
$\text{Prob}[P \rightsquigarrow_S^{tr} s]$	la probabilité que P génère la trace observable $s = o_1 o_2 \cdots o_n$ selon l'ordonnanceur S.

Équivalences

$\mathcal{R}, \Re$	relations d'équivalence.
$\equiv$	équivalence syntaxique.
$\preceq$	équivalence de trace (modèle possibiliste).
$\approx$	bisimulation faible (modèle possibiliste).
$\preceq_\phi$	équivalence de trace contextuelle.
$\approx_\phi$	bisimulation faible contextuelle.
$\simeq$	équivalence observationnelle asymptotique (modèle probabiliste).
$\simeq^{tr}$	équivalence de trace asymptotique (modèle probabiliste).
$\approx$	bisimulation probabiliste.
$\simeq_\phi$	équivalence observationnelle asymptotique contextuelle.
$\simeq_\phi^{tr}$	équivalence de trace asymptotique contextuelle.
$\approx_\phi$	bisimulation probabiliste contextuelle.

LISTE DES ANNEXES

INTRODUCTION

Les Enjeux

Avec l'introduction des ordinateurs dans notre quotidien, la nécessité des services de protection automatisés des données contenues dans ces machines devient évidente, surtout pour les systèmes partagés. Mais le plus grand changement qu'ait connu le monde informatique est l'utilisation des réseaux qui ont éliminé toutes les frontières géographiques. L'Internet est tellement pratique qu'il est devenu un outil incontournable pour notre société moderne. Le flux d'informations sensibles y transitant chaque jour ne cesse de s'accroître, qu'il s'agisse de commerce électronique pour lequel il faut sécuriser les transactions financières, des courriers émanant de professions libérales, ou tout simplement, d'informations concernant notre propre vie privée, qui doit être respectée dans toute démocratie. Toutes ces données doivent être protégées !

Toutefois, développer ou analyser un système de sécurité informatique de manière efficace est un problème très complexe. On doit définir d'une manière systématique les objectifs de sécurité et caractériser les approches permettant de les atteindre. Une bonne analyse doit prendre en compte les trois aspects de sécurité suivants : les *objectifs* de sécurité à atteindre, les *attaques* éventuelles, et les *mécanismes* de sécurité utilisés dans le système pour détecter ou prévenir ces éventuelles attaques. Nous rappelons brièvement ces trois aspects dans les sections qui suivent :

Services de sécurité

Les documents électroniques ayant pris une ampleur considérable pour la bonne marche de nos affaires quotidiennes, ils doivent avoir les différents types de fonctionnalités traditionnelles associées aux documents physiques : identification, signature,

licence, certification, etc. La liste est longue mais on peut extraire cinq critères qui englobent toutes ces fonctionnalités.

Confidentialité : pour assurer que seuls les utilisateurs habilités, dans les conditions normalement prévues, aient accès aux données. Cet accès inclut la lecture, l'impression, la visualisation ou simplement révéler l'existence.

Authenticité : pour assurer que l'origine d'un message ou un document électronique soit parfaitement identifiée, avec l'assurance que l'identité n'est pas fausse.

Intégrité : pour assurer qu'une information ne soit modifiée que par les utilisateurs habilités, dans les conditions normalement prévues. La modification inclut l'écriture, le changement, le changement de statut, l'effacement et la création.

Non-répudiation : pour assurer que ni l'émetteur, ni le récepteur d'un message ne puisse nier la transmission.

Disponibilité : pour assurer qu'un système puisse être utilisé par les utilisateurs habilités, dans les conditions d'accès et d'usage normales.

Attaques

Les attaques contre les systèmes informatiques sont nombreuses et revêtent de multiple formes. En considérant un tel système comme un moyen de transmission d'un flux d'informations d'une source, par exemple un fichier ou la mémoire, vers une destination, par exemple un autre fichier ou une autre machine, on peut caractériser les attaques comme l'indique la figure 1.

Interruption : rendre un service ou un élément non-disponible, inutilisable. C'est une attaque contre la *disponibilité.*

Interception : accès non autorisé à un service ou une ressource. C'est une attaque contre la *confidentialité.*

Modification : sabotage des ressources, changement non autorisé d'un fichier ou d'un message. C'est une attaque contre l'*intégrité*.

Fabrication : création de faux. Une tierce partie non-autorisée contrefait les objets dans un système. C'est une attaque contre l'*authenticité*.

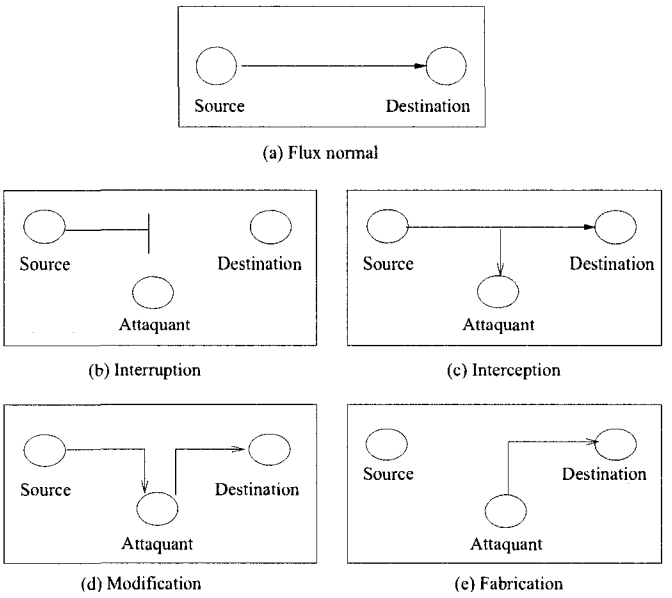

FIGURE 1 *Attaques contre les systèmes informatiques*

Ces différentes attaques peuvent être classées en deux catégories : les *attaques passives* (interception) et les *attaques actives* (interruption, modification et fabrication). Les attaques passives sont en général difficiles à détecter mais faciles à prévenir. Par contre, les attaques actives sont faciles à détecter mais souvent difficiles à contrer.

Mécanismes

Il n'existe pas un seul mécanisme de sécurité permettant de contrer n'importe quelle attaque. Mais il en existe un particulièrement efficace et incontournable, basé sur l'utilisation des *techniques cryptographiques* : le chiffrement symétrique et le chiffrement à

clé publique, le hachage cryptographique, et la signature électronique. Voir (Stallings, 2002) pour une description détaillée des aspects théoriques et pratiques des techniques cryptographiques.

Protocoles cryptographiques

La sécurité des systèmes d'information est aujourd'hui un domaine en pleine effervescence. Parmi les différents aspects de la sécurité informatique, la cryptologie et l'étude des protocoles cryptographiques ont pris une importance considérable. La cryptologie est originellement l'étude des moyens permettant de transmettre des messages secrets, et remonte à l'antiquité. Depuis, la portée des méthodes cryptologiques s'est considérablement étendue, pour inclure les propriétés de garantie d'authenticité, d'intégrité, de la distribution sécurisée des clés, etc.

La cryptologie est donc un domaine très riche, mais qui s'est longtemps limité essentiellement aux applications militaires. Il a fallu attendre le développement du commerce électronique au début des années 1990 pour voir un surgissement d'intérêt pour la cryptographie. Ces nouvelles applications de la cryptographie nécessitent des garanties de sécurité élevées.

Or, il se trouve que si, depuis les années 1980, on dispose d'algorithmes cryptographiques suffisamment sûrs, on ne peut pas en dire de même pour les protocoles cryptographiques. Il est bien connu maintenant que même en présence d'hypothèses de cryptographie parfaite, plusieurs protocoles cryptographiques sensés assurer la sécurité de la transmission sont erronés (Clark et Jacob, 1997). De nombreuses attaques s'appuient en effet sur les faiblesses logiques de ces protocoles. L'exemple classique est le protocole d'authentification (Needham et Schroeder, 1978). Ce protocole qui utilise le chiffrement à clé publique a pour but d'établir une authentification mutuelle entre deux usagers A et B. Nous supposons que la clé publique de A, notée k_A, est connue

TABLEAU 1 *Le protocole d'authentification de Needham-Schroeder*

Message 1 :	A	$\overset{\{n_A, A\}_{k_B}}{\longrightarrow}$	B
Message 2 :	B	$\overset{\{n_A, n_B\}_{k_A}}{\longrightarrow}$	A
Message 3 :	A	$\overset{\{n_B\}_{k_B}}{\longrightarrow}$	B.

de B, et réciproquement pour la clé publique de B. L'authentification par le protocole de Needham-Schroeder est décrite par l'échange des trois messages et illustrée dans le Tableau 1.

Dans le premier message, A envoie son nom et un *nonce de session* (i.e. un nombre généré aléatoirement) n_A chiffrés ensemble avec la clé k_B. Nous utilisons la notation $\{n_A, A\}_{k_B}$ pour désigner le résultat du chiffrement. L'usager B peut déchiffrer ce premier message et obtenir le nonce de A. Pour le deuxième message, B répond avec le message chiffré par la clé k_A contenant n_A et un nouveau nonce n_B. Si le nonce n_A retourné par B correspond bien au nonce initialement généré par A, alors A a authentifié B. Pour le dernier message, A retourne à B le nonce n_B chiffré avec le clé publique de B. L'usager B peut alors authentifier A si le nonce n_B retourné par A n'a pas été modifié.

Dans l'article (Lowe, 1996), l'auteur a démontré qu'il est possible d'attaquer le protocole de Needham-Schroeder. En effet, une attaque est possible lorsque l'usager A initie le protocole avec un usager malveillant E. L'usager E peut utiliser les informations reçues de A afin de se faire passer pour A envers un autre usager B. Cette attaque se déroule comme indiqué dans le Tableau 2.

TABLEAU 2 *Attaque contre l'authentification de Needham-Schroeder*

(1)	A	$\xrightarrow{\{n_A,A\}_{k_B}}$	E	(*A initie le protocole avec E*);
(1′)	E	$\xrightarrow{\{n_A,A\}_{k_B}}$	B	(*E utilise le nom de A pour initier le protocole avec B*);
(2′)	B	$\xrightarrow{\{n_A,n_B\}_{k_A}}$	E	(*E intercepte la réponse de B*);
(2)	E	$\xrightarrow{\{n_A,n_B\}_{k_A}}$	A	(*E rejoue la réponse de B à A*);
(3)	A	$\xrightarrow{\{n_B\}_{k_B}}$	E	(*A répond á E*);
(3′)	E	$\xrightarrow{\{n_B\}_{k_B}}$	B	(*B rejoue la réponse de A à B*).

Objectifs de la thèse

Il est bien connu maintenant que même en présence d'hypothèses de cryptographie parfaite, plusieurs protocoles cryptographiques sensés assurer la sécurité (confidentialité, authentification, non-répudiation, atomicité de l'argent, l'anonymat, l'intégrité des données, etc.) de la transmission sont erronés (Clark et Jacob, 1997). De nombreuses attaques s'appuient en effet sur les faiblesses logiques de ces protocoles.

Le domaine de la modélisation et de la vérification de tels protocoles a connu une véritable explosion dans les années 1990. On dispose au début des années 2000 de toute une gamme de modèles et de méthodes. Parmi les méthodes les plus prometteuses, il y a celles basées sur les algèbres de processus pour la spécification et l'équivalence-checking pour la vérification. En effet, d'une part, puisque les protocoles cryptographiques sont essentiellement des processus communicants, les algèbres de processus s'imposent naturellement comme langage de modélisation autant des participants au protocole que l'environnement. D'autre part, parmi les théories existantes de la sécurité de l'information développées au moment où on a débuté ce projet de doctorat, celle du flux d'information est très naturelle. Brièvement, l'idée est la suivante : il n'y a

pas de flux d'information si la réalisabilité d'une condition observable au niveau public n'est corrélée à la réalisabilité d'aucune condition observable uniquement au niveau confidentiel. L'absence de flux d'information s'exprime de façon naturelle en termes d'équivalences de comportement (e.g. équivalence de trace, de test, bisimulation) entre deux contextes, celui du système idéal, i.e. le système confiné n'interagissant pas avec l'environnement extérieur et celui du système tel qu'observé dans un environnement hostile quelconque.

Cette contrainte est toutefois trop forte dans le cas où une interférence inévitable, à cause de la nature de l'application, doit être admise : en particulier pour les protocoles cryptographiques pour lesquels l'observation au niveau public d'un cryptogramme ne permet pas d'en déduire le secret. Dans un tel cas, la propriété caractérisant la sécurité d'un tel système doit prendre en compte le contexte de cette déclassification admissible. Aussi, le professeur John Mullins a proposé la notion *d'interférence admissible* (Mullins, 2000), une version non déterministe de l'interférence intransitive qui n'admet la fuite d'information confidentielle qu'au travers d'un agent de déclassification. Ainsi, John Mullins et Stéphane Lafrance ont réussi à exprimer la plupart des propriétés de l'information transmise par un protocole cryptographique comme propriété de flot d'information, considéré comme admissible, du domaine privé vers le domaine public. Ils ont également établi une méthode de vérification de ces propriétés, dans le cadre de SPPA, d'une nouvelle algèbre de processus communicants avec passage de paramètres et primitives cryptographiques (Lafrance et Mullins, 2002). Par contre si SPPA permet l'expression de propriétés arbitraires de flux d'information admissible, aucune méthode de preuve effective n'y est associée. Il était donc nécessaire de développer une méthode de preuve cohérente et complète de ces propriétés : *c'est l'un des objectifs de cette thèse.*

De plus, le modèle SPPA ainsi que la plupart des modèles de spécification des protocoles cryptographiques existants au début de ce projet sont basés sur l'approche

dite possibiliste dans laquelle le non-déterminisme sert à modéliser les mécanismes aléatoires de génération de tous les comportements possibles des systèmes de sécurité. Ces modèles, dérivés du modèle de Dolev-Yao initialement proposé dans (Dolev et Yao, 1983), se basent également sur l'hypothèse de cryptographie parfaite et souffrent de deux limitations importantes. En effet, *l'absence des notions de probabilité* limite l'expressivité des ces modèles aux protocoles non probabilistes. Or, de plus en plus de protocoles cryptographiques utilisent des procédés aléatoires pour atteindre certains objectifs de sécurité (que l'on pense au protocole probabiliste de signature de contrat électronique (Norman et Shmatikov, 2002a) ou le "Crowds system" (Reiter et Rubin, 1998) qui garantit l'anonymat pour les transactions sur le web). La deuxième limitation importante est *l'hypothèse de cryptographie parfaite* qui est une abstraction très forte limitant énormément les capacités de l'attaquant. En effet, des crypto-systèmes jugés suffisamment sûrs tels que le One-Time Pad, le chiffrement RSA, et le chiffrement symétrique en mode ECB (Electronic Code Bloc), pour ne citer que ceux là, présentent des propriétés algébriques (l'associativité, la commutativité et la nilpotence du ou-exclusif du One-Time Pad, l'homomorphisme du mode ECB et du chiffrement à clé publique RSA, etc.) qui peuvent devenir des vulnérabilités exploitables par l'attaquant dans un protocole donné. Il était donc indispensable d'étendre ces modèles pour prendre en compte des systèmes probabilistes et d'éliminer, ou du moins justifier, l'hypothèse de cryptographie parfaite qui apparaît comme un choix arbitraire.

Il existe plusieurs approches pour étendre ces modèles possibilistes à la Dolev-Yao. La première, adoptée par Chatzikokolakis et Palamidessi, 2005; Aldini *et al.*, 2002; Bengt *et al.*, 2002, consiste à les étendre avec des notions de probabilité. Ces modèles s'attaquent donc uniquement au problème d'absence des notions de probabilité. La deuxième, adoptée par Abadi et Rogaway, 2000; Cortier et Warinschi, 2005; Herzog, 2005; Adão *et al.*, 2005; Janvier *et al.*, 2005, consiste à enrichir le modèle avec des théories équationnelles pour exprimer les propriétés algébriques des primitives

cryptographiques. Ces modèles, dont la solidité relativement au modèle réaliste des machines de Turing probabilistes cher aux cryptologues est maintenant bien établie, s'attaquent uniquement au problème de l'hypothèse de cryptographie parfaite. Enfin, la troisième approche que nous adoptons dans cette thèse consiste à définir un modèle qui, bien que formel, est relativement proche du formalisme computationnel de la cryptographie moderne puisqu'il travaille directement au niveau des primitives cryptographiques. Cette approche, également adoptée par Mitchell *et al.*, 2006; Blanchet, 2006; Laud et Vene, 2005, a l'avantage de s'attaquer aux deux problèmes cités ci-dessus et surtout d'établir un pont entre les modèles formels classiques de la vérification des crypto-protocoles et les modèles computationnels de la cryptographie moderne utilisés par cryptologues.

Contenu de la thèse

Les contributions principales de cette thèse s'articulent essentiellement en trois axes. Nous les décrivons ici brièvement. Les détails de ces contributions sont données au début de chacun des chapitres 2 à 5.

Modélisation. Le modèle SPPA, ainsi que tous les modèles formels proposés jusqu'à présent pour l'analyse des protocoles cryptographiques dans le cadre de la théorie du flux d'information admissible se heurtent à un problème majeur : l'obligation de quantifier sur tous les attaquants possibles. Par conséquent, l'interférence admissible telle que définie dans ces modèles n'est pas très pratique comme moyen de vérification. Aussi, nous proposons dans cette thèse un modèle de spécification des protocoles cryptographiques très simple dénommé *Cryptographic Security Process Algebra with Downgrading* ou CSPAD en abrégé. C'est une algèbre de processus à la CCS (Milner, 1989) avec passage de paramètres par valeur qui étend l'algèbre de processus SPA (Focardi et Gorrieri, 2001) syntaxiquement, pour prendre explicitement en compte la

spécification des primitives cryptographiques, et sémantiquement, pour prendre en compte la spécification des niveaux de sécurité avec mécanisme de déclassification. Nous y définissons de nouvelles caractérisations de l'interférence admissible grâce à une représentation symbolique (donc finie) des attaquants. Le problème d'analyse se réduit alors à un problème d'équivalence checking de systèmes infinis. Ces équivalences sont définies sur un système de transitions enrichi dont les transitions sont contraintes par la connaissance acquise par l'environnement, c'est-à-dire, des attaquants éventuels, à la réception de chaque message du protocole. On a défini sur ce système de transitions enrichi une équivalence de trace et une bisimulation faible à partir desquelles nous avons défini nos nouvelles caractérisations de l'interférence admissible qui évitent de quantifier sur tous les attaquants.

Vérification. Comme signalé dans la section précédente, si SPPA permet l'expression de propriétés arbitraires de flux d'information admissible, aucune méthode de preuve effective n'y est associée. Aussi, notre deuxième contribution importante est la formalisation des différentes propriétés de sécurité des protocoles cryptographiques en termes d'interférence admissible et surtout la preuve de la solidité de notre approche comparativement aux approches classiques basées sur la non-atteignabilité d'états non sécuritaires. En effet, nous avons exprimé le requis de secret en termes d'interférence admissible et avons établi formellement l'équivalence entre notre approche et l'approche classique de spécification du secret en termes de non-atteignabilité d'états non sécuritaires. A notre connaissance, c'est la première fois qu'une telle équivalence entre les approches basées sur le flux d'information et celles exprimées en termes de propriété d'atteignabilité est formellement établie. Nous avons également pu exprimer formellement les propriétés de correspondance telles que l'authentification d'entités, l'intégrité des données et la non-répudiation comme instances de l'interférence admissible. Notre approche, basée sur la déclassification, a permis de simplifier grandement la spécification et la vérification de ces propriétés comparativement aux approches basées sur la non-interférence (Focardi, 2001) et le modèle CSP (Lowe, 1997).

Modèle probabiliste. Enfin, nous avons proposé un modèle probabiliste polynomial pour l'analyse des crypto-protocoles qui s'est attaqué aux deux problèmes importants des modèles formels classiques à la Dolev-Yao : *l'absence des notions de probabilité* et *l'hypothèse de cryptographie parfaite*[1]. C'est un modèle dénommé ProSPA (*Probabilistic Security Process Algebra*) qui étend aussi bien syntaxiquement que sémantiquement le modèle CSPAD pour prendre en compte la spécification des systèmes de sécurité probabilistes utilisant des primitives cryptographiques (probabilistes) polynomiales et l'analyse de tels systèmes dans un environnement hostile. Nous avons montré que la sémantique de notre modèle reflète adéquatement la capacité de l'intrus à contrôler le réseau de communication, sans pour autant contrôler la réaction interne du système avec lequel il interagit. En effet, la combinaison du non-déterminisme et des notions de probabilité engendre d'autres nouveaux défis concernant la puissance de nuisance des attaquants. Pour obtenir des systèmes analysables, i.e. des systèmes totalement probabilistes, il est d'usage d'utiliser des ordonnanceurs pour résoudre le non-déterminisme dans ces protocoles. Mais, puisque le réseau de communication est généralement supposé être sous le contrôle de l'attaquant, l'ordonnancement est par conséquent contrôlé par ce dernier. Ainsi, si aucune restriction n'est imposée à l'ordonnancement, l'attaquant devient trop puissant et pourrait briser la plupart des systèmes puisqu'il pourra baser son choix sur des données secrètes qu'il ignore pourtant. Ce problème n'a été découvert que très récemment. Aussi, nous avons partitionné les actions du protocole en classes d'actions non distinguables par n'importe quel attaquant. Si ce problème a été abordé presque en même temps par d'autres chercheurs - (Chatzikokolakis et Palamidessi, 2007) pour une extension probabiliste (mais non computationnelle) du CCS et (Canetti *et al.*, 2006;Garcia *et al.*, 2007) pour des automates probabilistes - nous sommes certainement les premiers à le traiter dans un modèle computationnel à base d'algèbre de processus (Brlek *et al.*, 2007b).

[1]Les primitives cryptographiques sont considérés comme des boites noires et ne possèdent donc aucune faille de sécurité.

Toutefois, notre modèle ProSPA ainsi que tous les modèles formels probabilistes (computationnels ou non) pour la vérification des protocoles cryptographiques proposés à ce jour se heurtent à l'obligation de quantifier sur tous les attaquants possibles. Aussi, s'inspirant des techniques déjà utilisées pour le modèle possibiliste CSPAD, nous proposons une sémantique contextuelle (Brlek *et al.*, 2007a) de notre modèle probabiliste, malgré les difficultés techniques dues aux particularités des modèles computationnels. Cette sémantique contextuelle, en plus de nous débarrasser du quantificateur universel sur les attaquants, a grandement simplifié les difficultés techniques rencontrées dans la sémantique concrète de ProSPA, notamment lors de la définition des classes d'actions stratégiquement équivalentes et de la définition formelle d'ordonnanceurs.

Enfin, signalons que nous avons démontré tout au long de cette thèse l'utilité de nos différentes approches par d'importantes études de cas.

Organisation de la thèse

Le reste de ce document est organisé comme suit : une brève revue de la littérature sur la spécification des requis de sécurité des protocoles cryptographiques dans les modèles formels est donnée dans le chapitre 1. Le chapitre 2 présente notre modèle de spécification ainsi que nos nouvelles caractérisations de l'interférence admissible. Nous abordons la validation des propriétés de sécurité des protocoles cryptographiques dans le chapitre 3. L'extension de notre modèle en un modèle probabiliste polynomial est présentée dans le chapitre 4 et sa sémantique contextuelle dans le chapitre 5. Nous concluons la thèse par un rappel de nos principales contributions et les perspectives d'avenir.

CHAPITRE 1

REQUIS DES PROTOCOLES CRYPTOGRAPHIQUES ET MÉTHODES FORMELLES

L'étude des protocoles cryptographiques commence par une première étape de modélisation. Depuis quelques années, de nombreux modèles dédiés aux protocoles se sont développés. Nous allons en présenter une liste, certe non exhaustive, mais que nous espérons être représentative.

1.1 Premiers modèles

La plupart des approches qui appliquent les méthodes formelles à l'analyse des cryptoprotocoles se basent sur celle de Dolev et Yao, 1983, qui ont développé le premier modèle formel de l'intrus communément utilisé aujourd'hui. Les protocoles sont décrits par des règles de réécriture de mots ; un mot est secret s'il n'est pas accessible par réécriture. Cependant, les travaux de Dolev et Yao, et ses successeurs immédiats se concentrèrent sur une seule propriété de sécurité et de loin la plus simple : certains termes qualifiés de secrets ne doivent pas être appris par l'intrus. C'est ainsi que les premiers outils d'analyse automatisés des protocoles cryptographiques, comme les premières versions de *Interrogator* (Millen, 1984), se limitèrent à la définition du secret. D'autres, telles que les premières versions de *NRL Protocol Analyzer* (Meadows, 1996) permettent à l'usager de spécifier les propriétés de sécurité en terme de non-atteignabilité d'états non sécuritaires. Il y est notamment possible de spécifier ces états en termes de messages connus par l'intrus et les valeurs des variables locales des principaux. Cependant, l'usager n'avait aucune assistance pour spécifier ces états.

Probablement, le premier système d'analyse formelle des crypto-protocoles qui apporta un réel mécanisme de spécification formelle des propriétés de sécurité est la logique de croyance de Burrows *et al.*, 1990 (BAN). La logique de BAN ne s'intéresse pas du tout au secret, elle se confine aux problèmes d'authentification. La logique BAN permet de déduire les croyances des principaux participants au protocole concernant l'origine et l'usage de l'information telles que :

- D'où provient l'information ?
- À quoi sert elle ?
- Est-elle nouvelle ou une simple reprise ?
- Qui d'autres, parmi les participants, ont ces croyances ? etc.

On utilise la logique de BAN pour déterminer lesquelles de ces croyances peuvent être déduites à partir d'une idéalisation du protocole. Ce modèle ne dicte pas quelles croyances un protocole doit vérifier, c'est plutôt à celui qui analyse le protocole de décider lesquelles doivent être satisfaites et de déterminer si le protocole permet de les garantir.

1.2 Modèles classiques

Au début des années 90, l'approche de la vérification des protocoles cryptographiques a tendance à appliquer les outils existants tels que les *"model-checkers"* et les *"theorem-provers"*. C'est ainsi qu'on a eu besoin de développer des moyens de spécification des propriétés qu'on désire prouver. Et comme, en général, les chercheurs raisonnent directement par rapport aux messages échangés dans le protocole et non des croyances qu'on peut déduire de la réception de ces messages, il est sensé de développer des requis en termes des messages envoyés et reçus plutôt que des croyances que l'on peut déduire.

Les requis des protocoles cryptographiques peuvent être classés en deux catégories

qui dépendent des propriétés que les protocoles sont censés assurer : *le secret* et la *correspondance*. Le secret décrit qui a le droit d'accéder à l'information et permet de spécifier la confidentialité, tandis que la correspondance décrit les dépendances entre les événements qui se produisent dans le protocole. Elle est habituellement utilisée pour exprimer les propriétés d'authentification, d'intégrité et de non-répudiation. Ces deux types de requis sont en fait très proches l'un de l'autre. En effet, Syverson et Meadows, 1993, ainsi que Schneider et Sidiropoulos, 1996, définissent le secret comme un type de correspondance.

Bien sûr, on ne peut pas caractériser tous les requis en termes de secret et de correspondance. En effet, ce sont toutes deux des propriétés de sûreté[1]. Toute propriété autre que les propriétés de sûreté (telle que l'équité qui est très importante pour les protocoles de commerce électronique) n'entre dans aucune de ces deux catégories. Cependant le secret et la correspondance couvrent la plupart des requis pertinents pour les protocoles d'authentification et d'échange de clé, ce qui en fait un bon point de départ.

A première vue, la correspondance apparaît être la plus subtile et la plus compliquée, ce qui explique que les premiers travaux se concentrèrent sur ces requis de correspondance. De plus, l'emphase était mise sur l'habilité à caractériser une notion générale de correspondance dans une seule définition. Probablement le premier travail dans ce domaine était celui de Bird *et al.*, 1991. Dans l'introduction de leur article, ils ont décrit un protocole entre deux principaux A et B pour lequel toutes les exécutions telles que vues par les deux parties sont équivalentes une à une. Diffie *et al.*, 1992a ont raffiné cette idée pour obtenir le *"matching protocol runs (MPR)"* qui stipule qu'au moment où A complète un protocole l'observation de l'autre partie doit "correspondre" avec celle de A. Cette notion fût raffinée et formalisée par Bellare et

[1]Les propriétés de sûreté permettent de s'assurer qu'aucun fonctionnement anormal ne peut arriver alors que les propriétés de vivacité garantissent que le système évolue (afin d'effectuer ses fonctionnalités).

Rogaway, 1994, dans la notion de *"matching conversations"* qui développa l'idée en termes de la théorie de complexité.

Ces notions générales de correspondance peuvent être très utiles, mais ont aussi un désavantage. En effet, elles peuvent nous permettre de déterminer si l'information a été correctement distribuée ou non, mais ne permettent pas de déterminer si toute l'information qui devrait être authentifiée était comprise dans l'exécution ou non. Pour illustrer ce fait, considérons l'attaque de Lowe et Roscoe, 1997 contre le protocole *Station-to-Station* (Diffie *et al.*, 1992b). Le protocole est défini comme suit :

(1) A $\rightarrow$ B : x^{N_A}

(2) B $\rightarrow$ A : $x^{N_B}, \{S_B(x^{N_A}, x^{N_B})\}_K$
où K est la clé de Diffie-Hellman générée par A et B.

(3) A $\rightarrow$ B : $\{S_A(x^{N_A}, x^{N_B})\}_K$

Et l'attaque de Low comme suit :

(1) A $\rightarrow$ I(B) : x^{N_A}
un intrus I intercepte le message de A destiné à B
et le fait suivre à B en se faisant passer pour C.

(1') I(C) $\rightarrow$ B : x^{N_A}

(2) B $\rightarrow$ I(C) : $x^{N_B}, \{S_B(x^{N_A}, x^{N_B})\}_K$
I fait suivre le message à A et ainsi de suite.

À la fin, A croit partager une clé avec B, alors que ce dernier pense communiquer avec C et va donc rejeter le dernier message de A (il attend un message signé par C et non par A!). D'autre part, le protocole satisfait la définition du MPR puisque les messages reçus par B dans les deux cas (protocole normal et protocole attaqué) sont les mêmes. C'est d'ailleurs ce protocole qui a servi à illustrer le concept dans (Diffie *et al.*, 1992b).

Bien sûr, l'attaque de Lowe ne signifie pas que le protocole n'est pas sécuritaire. Tout

ce qu'elle nous indique, c'est que si les noms des principaux ne sont pas inclus dans les messages, une définition de sécurité basée sur la correspondance des messages ne peut pas détecter cette absence d'accord sur une information qui n'a jamais été envoyée.

La solution de Lowe à ce problème dans (Diffie *et al.*, 1992b) fût de renforcer la définition des MPR, pour inclure la condition suivante : si A a complété le protocole avec B, alors non seulement leurs visions doivent correspondre mais B doit en plus croire qu'il communique avec A. Dans un article ultérieur (Lowe et Roscoe, 1997), il formalisa ce concept en utilisant l'algèbre de processus CSP et développa une hiérarchie d'authentification.

Ces requis sont habituellement spécifiés en utilisant des logiques temporelles qui sont généralement utilisées pour prouver la correction des spécifications par model-checking. C'est d'ailleurs cette approche qu'ont adoptée Syverson et Meadows lors du développement de leur outil *NRL Protocol Analyzer* (Syverson et Meadows, 1993), qui deviendra le *NRL Protocol Analyzer Temporal Requirements Language (*NPATRL*)*. NPATRL est un langage très simple qui a permis de spécifier plusieurs protocoles de distribution des clés (Syverson et Meadows, 1996; Syverson et Meadows, 1995) et des protocoles de commerce électronique complexes tel que le protocole SET (Meadows et Syverson, 1998). Mais il montra ses limites notamment en ce qui concerne des propriétés de *secret parfaitement transmis* qui exigent qu'une clé de session ne soit compromise qu'après la compromission de la clé longue (la clé privée).

Les logiques temporelles ne sont évidemment pas indispensables pour spécifier ces types des propriétés. D'autres formalismes marchent parfaitement bien. Par exemple, dans (Schneider et Sidiropoulos, 1996) on définit l'authentification en termes des messages qui doivent précéder un message donné et le secret en termes d'autres requis de correspondance, c'est-à-dire, l'intrus ne doit pas prendre connaissance d'un message tant que ce message ne lui est pas explicitement envoyé. On y utilise l'algèbre CSP pour formaliser ces propriétés.

Une autre approche proposée par Focardi, 2001, permet de spécifier les propriétés de sécurité en utilisant des notions dérivées de la non-interférence. Leur notion de correction, *Generalized Nondeducibility on Composition* ou GNDC, est définie comme suit :

Soient P un processus représentant un protocole cryptographique opérant en l'absence d'intrus et $(P||X)$ la composition de P et un intrus X. Soit α une fonction des processus vers les processus où $\alpha(P)$ est un processus modélisant le comportement "correct" de P. Soit $\simeq$ une relation de préordre. Soient C l'ensemble de tous les canaux entre les principaux honnêtes et $Q\backslash C$ la restriction d'un processus Q à C. Alors le processus P satisfait $\text{GNDC}^{\alpha}_{\simeq}$, si pour tout intrus X on a

$$(P||X)\backslash C \simeq \alpha(P)$$

Dans le cas où α est la fonction identité et $\simeq$ une équivalence de trace, la propriété devient le NDC ou le Nondeducibility on Composition qui exige que les traces produites par la composition du processus avec un intrus quelconque soient les mêmes que celles produites en l'absence d'intrus. Ceci est une propriété de flot d'information dans laquelle l'intrus et P jouent respectivement les rôles de haut niveau et bas niveau. Puisque NDC exige que le protocole se comporte de la même manière en présence ou non d'un intrus quelconque, il est beaucoup plus solide que tous les autres modèles décrits ci-dessus. À vrai dire, on peut le considérer comme la définition la plus rigoureuse. En outre, GNDC fournit un cadre général pour la spécification des propriétés de sécurité qui dépassent les notions habituelles de secret et de correspondance telles que les propriétés de vivacité.

Cependant, tout système qui, comme un crypto-système, déclassifie de l'information, échappe à la non-interférence. Il y a en effet, dans ce cas, dépendance causale de la paire message-clé (m, K) vers le cryptogramme $\{m\}_K$ que l'on doit admettre. D'où

l'importance de l'*interférence admissible,* une propriété de flux d'information qui exige que les seules interférences du haut niveau sur le bas niveau le soient uniquement via les actions de déclassification. Il a été proposé par Mullins dans (Mullins, 2000) d'exprimer cette propriété en termes de la non-interférence de la façon suivante : tout sous-processus P' de P n'exécutant aucune de ses actions déclassifiantes doit satisfaire la non-interférence.

Une autre technique, qui mérite d'être signalée, est l'usage de la théorie de typage pour spécifier et vérifier des propriétés de sécurité des protocoles cryptographiques (Abadi, 1999; Gordon et Jeffrey, 2004). Ici à chaque composante d'un protocole tels que les messages et les canaux, est attribuée un type : par exemple *public* ou *secret.* Des règles d'inférence de types sont également définies pour les différentes opérations telles que le chiffrement, le déchiffrage, etc. Une faille de sécurité peut être spécifiée en termes de violation de typage, tel qu'un message de type privé apparaissant dans un canal de type public. Cette technique a été abondamment utilisée pour vérifier les propriétés de secret, mais Gordon et Jeffrey, 2004; Gordon et Jeffrey, 2002, ont développé une façon de l'appliquer aux propriétés de correspondance.

1.3 Modèles graphiques

Les modèles décrits à la section précédente nous donnent une grande flexibilité et expressivité pour spécifier les requis des protocoles cryptographiques. Mais, plus le système à modéliser est complexe, plus il est difficile de spécifier ses requis. Dans cette section, nous décrivons très brièvement un modèle graphique qui semble nous faciliter la modélisation des protocoles cryptographiques et leurs requis : le modèle des *"Strand Spaces"*.

Le modèle des "strand spaces" (Fabrega *et al.*, 1998b) est bien connu et populaire pour l'analyse des protocoles cryptographiques dans lequel les actions des principaux

sont modélisées par des graphes. Un *strand* représente un principal jouant un rôle dans le protocole. Les envois et réceptions des messages sont représentés par des noeuds positifs et négatifs. Les noeuds représentant deux événements successifs sur un strand sont reliés par des doubles arcs. Un *bundle* est une collection de strands, dans lequel les noeuds positifs (envois) peuvent être connectés aux noeuds négatifs (réceptions) par un arc simple si le message à envoyer correspond au message à recevoir. Ce modèle facilite la représentation graphique des protocoles et dans (Fabrega *et al.*, 1998a), les auteurs ont décrit plusieurs façons d'utiliser les caractéristiques graphiques des strands pour spécifier des propriétés de sécurité. Par exemple, en utilisant cette représentation graphique d'un protocole, on peut spécifier les propriétés de correspondance en termes des placements relatifs des strands dans le bundle. En effet, si un événement E1 doit précéder un autre événement E2, nous pouvons les représenter par des portions des strands et vérifier si dans toutes les configurations possibles du bundle, E1 apparaît avant E2. Cependant, si le modèle semble simplifier la modélisation des protocoles et leurs requis, à notre connaissance, il n'existe pas de travaux définissant très bien comment spécifier les propriétés de sécurité dans ce domaine.

1.4 Modèles quantitatifs et probabilistes

Jusqu'à présent, les modèles que nous avons présentés ici s'intéressent aux propriétés de sécurité des systèmes discrets. Ça marche bien lorsqu'on veut analyser des protocoles suivant le modèle de Dolev-Yao, où les systèmes cryptographiques sont considérés comme des boites noires et les principaux communiquant à travers un médium contrôlé par un intrus hostile qui peut intercepter, lire et altérer tout le trafic. Or, la sécurité d'un protocole dépend non seulement de la façon dont il utilise les crypto-systèmes, mais aussi et surtout de la sécurité de ces crypto-systèmes. Il est donc important de développer des critères d'analyse des protocoles cryptographiques qui prennent en compte les propriétés spécifiques aux crypto-systèmes utilisés. Pa-

rallèlement à l'explosion des méthodes formelles d'analyse des crypto-protocoles, il y a eu des efforts pour développer des critères de validation des crypto-systèmes et des protocoles cryptographiques basés sur la théorie de la probabilité et de la complexité (Bellare et Rogaway, 1994; Bellare *et al.*, 1994; Bellare *et al.*, 1997; Goldreich *et al.*, 1991; Goldwasser et Micali, 1984; Goldwasser *et al.*, 1988). Dans ces modèles, les messages sont des chaînes de bits tandis que l'adversaire est une machine de Turing probabiliste polynomiale. Si ces modèles reflètent adéquatement l'exécution réelle des protocoles de sécurité dans un environnement hostile, les preuves de sécurité sont par contre très complexes et surtout manuelles. Depuis l'article séminal d'(Abadi et Rogaway, 2000), il y a eu beaucoup de travaux (voir par exemple (Cortier et Warinschi, 2005; Herzog, 2005; Adão *et al.*, 2005; Janvier *et al.*, 2005)) qui tentent de relier ces deux visions de la cryptographie et ainsi pouvoir justifier les hypothèses "arbitraires" des modèles à la Dolev-Yao relativement au modèle réaliste à base de machines de Turing utilisées par cryptologues. Toutefois, ces modèles exigent de restreindre les primitives cryptographiques utilisables, puisque ces dernières doivent souvent satisfaire des propriétés de sécurité très importantes. De même, les protocoles vérifiables doivent également satisfaire certaines conditions telles que l'absence des clés cycliques (Adão *et al.*, 2005). C'est pourquoi d'autre modèles plus récents ont exploré des moyens d'intégrer les techniques d'analyse des crypto-systèmes à base des machines de Turing directement dans les modèles formels (Mitchell *et al.*, 2006; Brlek *et al.*, 2007b; Blanchet, 2006; Laud et Vene, 2005). En effet, bien que formels, ces modèles sont relativement proches du formalisme computationnel de la cryptographie moderne puisqu'ils travaillent directement au niveau des primitives cryptographiques.

L'utilisation des crypto-systèmes n'est pas le seul domaine qui implique des propriétés quantitatives. En effet, plusieurs protocoles censés assurer l'anonymat (Chaum, 1988; Reiter et Rubin, 1998) sont naturellement probabilistes. C'est ainsi que d'autres travaux ont étendu les modèles possibilistes à la Dolev-Yao avec des notions de probabilité. Dans cette catégorie, nous pouvons citer les travaux de Shmatikov sur des

protocoles d'anonymat et de signature des contrats dans lesquels les protocoles et leurs requis sont modélisés par des chaînes de Markov (Norman et Shmatikov, 2002b;Shmatikov, 2004b), les rendant faciles à analyser par des model-checkers probabilistes. D'autres travaux récents de cette catégorie sont ceux de Palamidessi sur l'anonymat (Chatzikokolakis et Palamidessi, 2005; Chatzikokolakis et Palamidessi, 2007) dans lesquels les requis de sécurité sont exprimés grâce à une version probabiliste de la sémantique de test ou ceux de Aldini *et al.*, 2002 sur la non-interférence probabiliste. Les protocoles censés protéger contre les attaques de déni de service (DoS), constituent un autre domaine où les considérations quantitatives deviennent pertinentes. Récemment, quelques travaux commencent à investiguer les moyens d'appliquer des méthodes formelles à l'analyse des protocoles qui doivent vérifier des propriétés quantitatives. Nous pouvons citer les travaux de Meadows, 2001, sur le DoS où les propriétés sont spécifiées en termes des comparaisons entre les ressources dépensées par le répondeur versus celles dépensées par l'initiateur.

CHAPITRE 2

L'INTERFÉRENCE ADMISSIBLE

2.1 Introduction

La non-interférence, initialement proposée par Goguen et Meseguer, 1982, joue un rôle central dans la formalisation des propriétés de sécurité de flux d'information. Elle a pour but de caractériser l'absence complète de fuite d'information dans un système de sécurité multi-niveaux. Comme l'ont signalé plusieurs auteurs par le passé (Bossi *et al.*, 2004; Backes et Pfitzmann, 2003; Lowe, 2002; Mantel, 2001; Mullins, 2000; Roscoe et Goldsmith, 1999; Ryan et Schneider, 1999; Pinsky, 1992), l'absence totale de fuite d'information entre deux niveaux de sécurité, communement appelés *haut niveau* et *bas niveau*[1], est une exigence très forte difficilement réalisable pour un système réel. En particulier pour les protocoles cryptographiques pour lesquels, par exemple, la réception d'un cryptogramme révèle l'existence d'un message secret sans pour autant révéler le secret en question. Ce type de flux d'information est non seulement admissible mais surtout souhaitable pour la plupart des protocoles cryptographiques puisque, d'après le *principe de Kerckhoffs*, leur sécurité doit être basée uniquement sur le secret des clés et non sur celui de la nature des protocoles eux-mêmes. Ainsi, lorsqu'on conçoit ou analyse un vrai protocole de sécurité, on est très souvent amené à inclure quelques mécanismes de *déclassification* qui permettent de contrôler le flux d'information qu'on veut admettre.

[1] L'interprétation de ces niveaux de sécurité dans le cadre des protocoles cryptographiques évoluant dans un environnement hostile sera introduite dans la section 2.3.

2.1.1 Travaux similaires

Le problème de la détection de flux d'information non admissible remonte aux travaux de Goguen et Meseguer, 1984, qui ont introduit la notion de la *non-interférence conditionnelle* qui admet des flux d'information de haut vers le bas niveau à travers des canaux contrôlés. La non-interférence conditionnelle fût mieux formalisée plus tard par Haigh et Young, 1987. Ensuite, Lincoln et Rushby, 1993, développèrent une théorie formelle de la déclassification pour les systèmes déterministes basée sur la *non-interférence intransitive* où le flux admissible du haut vers le bas niveau transite à travers une tierce partie de confiance qui contrôle le flux d'information tandis que tout flux direct du haut vers le bas niveau est interdit. Les concepts standard de la non-interférence et de l'interférence intransitive ont été unifiés dans (Pinsky, 1992).

Toutes les approches ci-dessus se limitent aux systèmes déterministes et ne sont pas applicables aux systèmes distribués tels que les protocoles cryptographiques évoluant dans un environnement hostile. Pour y remédier, Roscoe et Goldsmith, 1999, proposa une formalisation de la non-interférence transitive pour le CSP[2] (Hoare, 1978) limitée à une certaine forme de non-déterminisme. Le premier modèle qui prend en compte les systèmes non-déterministes est probablement celui de Mantel, 2001. Une généralisation de la caractérisation statique (SNNI) de la non-interférence forte de Focardi et Gorrieri, 2001, est proposée par Mullins, 2000. Il y présente une notion d'*interférence admissible non-déterministe* (NAI) dans le contexte de l'algèbre de processus CCS[3] (Milner, 1989).

Toutefois, les modèles Roscoe et Goldsmith, 1999; Mantel, 2001; Mullins, 2000, se basent tous sur l'équivalence de trace et ne permettent pas de détecter un flux non admissible dû au fait qu'une partie peut bloquer ou non le système. Pour cela Ryan

[2]Communicating Sequential Processes
[3]Calculus of Communicating Systems

et Schneider, 1999, proposèrent quelques généralisations de la non-interférence pour les processus CSP dans le but de formaliser la *non-interférence conditionnelle et partielle.* Dans (Lafrance et Mullins, 2002), les auteurs proposent la notion de la *bisimulation-basée non-déterministe interférence admissible* (BNAI) généralisant la BSNNI de Focardi et Gorrieri, 2001. Dans (Bossi *et al.*, 2004), les auteurs proposent un modèle général de spécification de la non-interférence pour les processus SPA (Focardi et Gorrieri, 2001) qui permet de prendre en compte la déclassification.

Enfin, signalons qu'une notion de la *non-interférence transitive probabiliste* est proposée dans (Backes et Pfitzmann, 2003). Et plus récemment, Hadj-Alouane *et al.*, 2005, proposèrent une notion de flux d'information temps réel (dense) et posent le problème de synthèse de contrôleur des systèmes de sécurité temps réel à flux d'information admissible. Il y est étudié le problème de décidabilité de la vérification et de la synthèse de contrôleurs de diverses propriétés de flux d'information temps réel.

2.1.2 Notre contribution

Dans le cadre de nos travaux au laboratoire de Conception et de Réalisation des Application Complexes (CRAC) de l'École Polytechnique de Montréal nous avons réussi à exprimer la plupart des propriétés de l'information transmise par un protocole cryptographique comme propriété de flux d'information, considéré comme admissible, du domaine privé vers le domaine public (Lafrance et Mullins, 2002; Lafrance et Mullins, 2003a; Lafrance et Mullins, 2003b; Brlek *et al.*, 2003; Lafrance, 2004; Brlek *et al.*, 2005; Bastien *et al.*, 2006). L'interférence admissible apparaît donc comme une *primitive* très naturelle à partir de laquelle on peut exprimer la plupart des propriétés de sécurité des systèmes à flux d'information admissible.

Stéphane Lafrance et John Mullins ont également établi une méthode de vérification de ces propriétés, dans le cadre de SPPA, d'une nouvelle algèbre de processus com-

municants avec passage de paramètres et primitives cryptographiques (Lafrance et Mullins, 2002). Cependant, la sémantique de l'algèbre de processus SPPA traite les primitives cryptographiques (i.e. les calculs internes du système) comme étant des actions du système de transitions engendré par la sémantique du protocole. Ce qui a comme conséquence de faire exploser rapidement le nombre d'états des systèmes exprimés dans ce modèle et ainsi de les rendre difficilement vérifiables. En effet, bien qu'en général le nombre des communications sur les canaux publics d'un protocole donné soit très réduit, celui des calculs internes (chiffrement, déchiffrement, pairages, décompositions des termes, génération des nonces et clés de session, ...) peut être énorme même pour un protocole à quelques passes.

Pour résoudre ce problème d'explosion combinatoire, notre *première contribution* dans ce chapitre consiste à proposer un modèle de spécification des protocoles cryptographiques dénommé *Cryptographic Security Process Algebra with Downgrading* ou CS-PAD en abrégé. C'est une algèbre de processus à la CCS avec passage de paramètres par valeur qui étend l'algèbre de processus SPA (Focardi et Gorrieri, 2001), d'une part syntaxiquement pour prendre explicitement en compte la spécification des primitives cryptographiques et d'autre part sémantiquement pour prendre en compte la spécification des niveaux de sécurité avec mécanisme de déclassification. Les calculs internes du système (i.e. les calculs des primitives cryptographiques) sont considérés comme des gardes et n'interfèrent pas avec les actions sur les canaux de communication.

D'autre part, l'interférence admissible, telle qu'exprimée dans (Mullins, 2000;Lafrance et Mullins, 2002), ainsi que dans les différentes approches ci-dessus, se heurte au problème de quantification sur tous les attaquants possibles contre le protocole : une source d'indécidabilité. Pour surmonter cette difficulté, notre première solution consista à analyser les systèmes de sécurité contre le plus grand attaquant, c'est-à-dire, l'attaquant capable de simuler tout autre attaquant du modèle (Brlek *et al.*, 2003;Brlek *et al.*, 2005). Mais un tel attaquant n'existe que pour les relations de préordre telles que l'équivalence de trace et la simulation (co-simulation).

Aussi, notre *deuxième contribution* dans ce chapitre consiste à surmonter cette difficulté en proposant une approche qui ne nécessite ni d'exhiber un attaquant explicite ni l'existence d'un plus grand attaquant. Inspirés par Boreale *et al.*, 1999, qui caractérisent l'équivalence de *may test* et de la *congruence barbue* du spi-calcul au moyen d'équivalence de trace et de la bisimulation construites sur des systèmes de transitions contextuels, nous proposons une sémantique contextuelle de notre modèle CSPAD. La sémantique contextuelle ainsi obtenue engendre des systèmes de transitions enrichis de la forme suivante : les états sont des *configurations* $\phi \rhd P$ où P est un processus et ϕ est la connaissance actuelle de l'environnement et les transitions $\phi \rhd P \xrightarrow{\alpha} \psi \rhd Q$ qui signifient que le processus P interagissant avec le contexte de connaissance ϕ peut faire l'action α et évoluer comme Q interagissant avec un contexte de connaissance ψ. Ces transitions sont contraintes par la connaissance acquise par l'environnement, c'est-à-dire des attaquants éventuels, à la réception de chaque message du protocole.

En vue de vérifier des propriétés des systèmes à flux d'information admissible, notre troisième contribution consiste à définir une équivalence de trace et une bisimulation faible contextuelles à partir desquelles nous avons défini de nouvelles caractérisations de l'interférence admissible. Nous avons montré que notre sémantique contextuelle préserve les équivalences de préordre telles que les équivalences basées sur l'équivalence de trace. Enfin, nous avons montré que les équivalences basées sur la bisimulation raffinent celles basées sur l'équivalence de trace.

2.1.3 Organisation du chapitre

La section 2.2 présente notre modèle de calcul. Les définitions formelles de l'interférence admissible sont données dans la section 2.3. La sémantique contextuelle de notre modèle ainsi que les nouvelles caractérisations statiques de l'interférence admissible sont présentées dans la section 2.4.

2.2 Le modèle CSPAD

L'étude des protocoles cryptographiques commence par une première étape de modélisation. Nous définissons dans cette section notre modèle de calcul. C'est une algèbre de processus à la CCS (Milner, 1989) avec passage de paramètres par valeur appelée CSPAD (*Cryptographic Security Process Algebra with Dawngrading*), qui étend l'algèbre de processus SPA (Focardi et Gorrieri, 2001) syntaxiquement, pour prendre explicitement en compte la spécification des primitives cryptographiques, et sémantiquement, pour prendre en compte la spécification des niveaux de sécurité avec mécanisme de déclassification.

2.2.1 Syntaxe

Termes

CSPAD utilise une algèbre de messages basée sur les catégories syntaxiques des messages de base (identificateurs des principaux et les nombres dénotés respectivement par les ensembles $\mathcal{I}$ et $\mathcal{N}$) et des variables dénotées par l'ensemble $\mathcal{V}$. L'ensemble des *termes* $\mathcal{T}$ est construit comme suit :

$$\begin{array}{rcllllll} t & ::= & m & (message) & \mid x & (variable) & \mid (t,\ldots,t) & (tuple) \mid \\ & \mid & \{t\}_t & (cryptogramme) & \mid h(t) & (hachage) & & \end{array}$$

Pour tout terme t, on dénote par $fv(t)$ l'ensemble des variables dans t. Un *message* est un terme clos (i.e. ne contenant pas de variables). L'ensemble des termes clos est dénoté par $\mathcal{M}$ et est défini comme la clôture transitive de $\mathcal{I} \cup \mathcal{N}$ sous les règles d'inférence du Tableau 2.1. Par souci de simplicité, nous distinguerons un sous-ensemble $\mathcal{K} \subseteq \mathcal{M}$ de messages qui peuvent être utilisé comme clé de chiffrement. Pour prendre en compte le chiffrement à clé publique, nous utilisons un opérateur idem-

potent $[-]^{-1} : \mathcal{K} \rightarrow \mathcal{K}$ tel que a^{-1} dénote la clé privée correspondant à la clé publique a, et inversement. Dans le cas symétrique, il suffit de poser $a^{-1} = a$. On suppose chiffrement et hachage parfaits. Ce système d'inférence permet de combiner plusieurs messages $m_1, m_2, \cdots, m_n$ pour obtenir le n-tuplet $(m_1, m_2, \cdots, m_n)$ (règle $\vdash_{pair}$). Inversement, il permet d'extraire chaque composante m_i du n-tuplet $(m_1, m_2, \cdots, m_n)$ grâce à la projection p_i (règle $\vdash_{p_i}$). Étant donné un message m et une clé de chiffrement, la règle $\vdash_{enc}$ permet d'inférer le cryptogramme $\{m\}_K$. Inversement, la règle $\vdash_{dec}$ permet d'inférer le message m à partir du cryptogramme $\{m\}_K$ et de la clé de déchiffrage K^{-1}. Enfin la règle $\vdash_{hash}$ permet d'inférer le haché (condensé) de tout message m.

TABLEAU 2.1 *Règles d'inférence des messages* CSPAD.

$$\vdash_{pair} \frac{m_1 \quad m_2 \quad \cdots \quad m_n}{(m_1, m_2, ..., m_n)} \qquad \vdash_{p_i} \frac{(m_1, ..., m_n)}{m_i} \quad i \leq n$$

$$\vdash_{enc} \frac{m \quad K}{\{m\}_K} \qquad \vdash_{dec} \frac{\{m\}_K \quad K^{-1}}{m}$$

$$\vdash_{hash} \frac{m}{h(m)}$$

Processus

On considère un ensemble dénombrable $\mathcal{C}$ de *canaux publics*. Les canaux publics sont utilisés pour spécifier les échanges de messages entre les principaux. Chaque canal public c a un domaine prédéterminé $\mathrm{dom}(c)$ de messages qui peuvent y transiter. Les *processus* CSPAD sont construits comme suit :

$$\begin{aligned} P \ ::= \ & \mathbf{0} \ \mid \ c(x).P \ \mid \ \overline{c}(m).P \ \mid \ P + P \ \mid \ P|P \ \mid \ P\backslash\Gamma \ \mid \\ & \mid \ [t = t]P \ \mid \ [< t_1, \ldots, t_n > \vdash_{op} x]P \ \mid \ A(t_1, \ldots, t_n) \end{aligned}$$

où Γ est un ensemble d'actions visibles (définie dans la section 2.2.2 qui suit), $op \in \{pair, p_i, enc, dec, hash\}$ et $t_1, \ldots, t_n$ des termes. Étant donné un processus P,

l'ensemble des *variables libres*, (dénoté $fv(P)$) est l'ensemble des variables x de P qui ne sont dans la portée d'aucun préfixe d'entrée (de la forme $c(x)$) ou d'inférence (de la forme $[< t_1, \ldots, t_n > \vdash_{op} x]$). Un processus clos est un processus sans variable libre et l'ensemble des processus clos est dénoté par $\mathcal{P}roc$. Dans la suite de ce chapitre, tous les processus considérés sont des processus clos.

2.2.2 Sémantique opérationnelle

Soit $m \in \mathcal{M}$ un message, les *actions* de CSPAD sont définies comme suit :

$$\begin{array}{rcll} \alpha & ::= & \overline{c}(m) & (output) \\ & | & c(m) & (input) \\ & | & \tau & (action\ silencieuse)\ . \end{array}$$

$\mathcal{A}ct$ dénote l'ensemble de toutes les actions de CSPAD et $\mathcal{V}is = \mathcal{A}ct \setminus \{\tau\}$ l'ensemble des actions visibles.

La sémantique opérationnelle de CSPAD est un système de transitions $(\mathcal{E}, \mathcal{E} \times \mathcal{A}ct \times \mathcal{E}, E_0)$ engendré par les règles d'inférence du Tableau 2.2 où $\mathcal{E} \subseteq \mathcal{P}roc$ est l'ensemble des sommets, $\mathcal{E} \times \mathcal{A}ct \times \mathcal{E}$ l'ensemble des transitions et $E_0 \in \mathcal{P}roc$ le sommet initial. Nous dénotons par $P \xrightarrow{\alpha} P'$ la transition (P, α, P') du sommet P au sommet P' pour signifier que le processus P peut faire l'action α et évoluer comme le processus P'. C'est une extension de SPA (Focardi et Gorrieri, 2001) avec des primitives cryptographiques et un mécanisme de passage de paramètres par valeur. Nous l'esquissons brièvement dans ce paragraphe. Le mécanisme d'output permet à un principal A d'envoyer un message sur les canaux publics (règle `Output`). Dualement, le mécanisme d'input doit prévoir la réception de n'importe quel message sur un canal public (règle `Input`). L'appel des fonctions cryptographiques et le retour de valeur circulant sur des canaux privés sont modélisés par la règle `Infer`.. Les opérateurs de choix non-déterministe

TABLEAU 2.2 *Sémantique des processus* CSPAD

$$\texttt{Output}\quad \frac{m \in \mathrm{dom}(c)}{\overline{c}(m).P \xrightarrow{\overline{c}(m)} P} \qquad \texttt{Input}\quad \frac{m \in \mathrm{dom}(c)}{c(x).P \xrightarrow{c(m)} P[m/x]}$$

$$\texttt{Match}\quad \frac{m=m' \quad P \xrightarrow{\alpha} P'}{[m=m']P \xrightarrow{\alpha} P'} \qquad \texttt{Rest.}\quad \frac{P \xrightarrow{\alpha} P' \quad \alpha \notin \Gamma}{P \backslash \Gamma \xrightarrow{\alpha} P' \backslash \Gamma}$$

$$\texttt{Par.}\quad \frac{P_1 \xrightarrow{\alpha} P_1'}{P_1|P_2 \xrightarrow{\alpha} P_1'|P_2} \qquad \texttt{Sum}\quad \frac{P_i \xrightarrow{\alpha} P_i'}{P_1+P_2 \xrightarrow{\alpha} P_i'}\ i=1,2$$

$$\frac{P_2 \xrightarrow{\alpha} P_2'}{P_1|P_2 \xrightarrow{\alpha} P_1|P_2'} \qquad \texttt{Sync.}\quad \frac{P_1 \xrightarrow{\overline{c}(a)} P_1' \quad P_2 \xrightarrow{c(a)} P_2'}{P_1|P_2 \xrightarrow{\tau} P_1'|P_2'}$$

$$\texttt{Inf.}\quad \frac{<m_1,\ldots m_n> \vdash_{op} m \quad P[m/x] \xrightarrow{\alpha} P'}{[<m_1,\ldots m_n> \vdash_{op} x]P \xrightarrow{\alpha} P'}$$

$$\texttt{Def.}\quad \frac{P[m_1/x_1,\ldots m_n/x_n] \xrightarrow{\alpha} P' \quad A(x_1,\ldots,x_n)=P}{A(m_1,\ldots,m_n) \xrightarrow{\alpha} P'}$$

et de parallélisme ainsi que la conditionnelle (règles `Sum.`, `Par.` et `Match`) sont définies de façon habituelle. Le processus $P \setminus \Gamma$ (où Γ est un ensemble d'actions visibles *fermé*[4]) se comporte comme le procesus P restreint aux actions qui ne sont pas dans Γ (règle `Restr.`). Soit $C \subseteq \mathcal{C}$ un ensemble des canaux. Nous utilisons la notation $P \setminus C$ pour restreindre toute communication publique sur les canaux dans C. Formellement, $P \setminus C \equiv P \setminus \Gamma$ où $\Gamma = \{\overline{c}(m) \mid c \in C,\ m \in dom(c)\} \cup \{c(m) \mid c \in C,\ m \in dom(c)\}$. Ainsi, nous pouvons modéliser l'exécution d'un processus dans un environnement sûr ou en isolation (à travers des canaux privés), en restreignant les canaux de communication publics.

[4]C'est-à-dire si $\alpha \in \Gamma$ alors $\overline{\alpha} \in \Gamma$.

2.2.3 Exemple de modélisation : le protocole de Needham-Schroeder

La modélisation du protocole de Needham-Schroeder décrit dans le Tableau 1 est illustrée par le Tableau 2.3. Nous spécifions chaque principal par un processus qui modélise son comportement aux différentes étapes du protocole. Nous désignons par $C = \{c_i / 1 \leq i \leq 3\}$ l'ensemble des canaux publics du protocole : le canal c_i correspond à l'étape (i) du protocole. Nous supposons que chaque processus utilise des variables différentes (dont la portée se limite au processus en question). Chaque processus A^i (respectivement B^i) correspond au comportement de l'initiateur (respectivement répondeur) à l'intérieur du protocole à l'étape où il se trouve. Par exemple, le processus B signifie qu'après la réception d'un message y_1 par le canal public c_1 (Message 1 du protocole), B doit décomposer le message en trois messages y_2, y_3 et y_4. Il doit ensuite vérifier que y_3 correspond bien à son nom, avant de déchiffrer y_4 en y_5 avec sa clé privée. Enfin, il doit vérifier que la deuxième composante de y_5 (i.e. y_7) correspond bien à y_2 avant de se comporter comme le processus B^2 qui prépare sa réponse. Ainsi, le protocole est simplement spécifié par la composition parallèle des principaux A et B, c'est-à-dire, par le processus $A|B$.

TABLEAU 2.3 *Spécification du protocole de Needham-Schroeder en* CSPAD

$$
\begin{array}{ll}
A ::= & [((n_A, A), k_B) \vdash_{enc} x_1][(A, B, x_1) \vdash_{pair} x_2]\overline{c_1}(x_2).A^1 \\
A^1 ::= & c_2(x_3).[x_3 \vdash_{p_1} x_4][x_3 \vdash_{p_2} x_5][x_3 \vdash_{p_3} x_6][x_4 = B][x_5 = A] \\
 & [(x_6, k_A^{-1}) \vdash_{dec} x_7][x_7 \vdash_{p_1} x_8][x_7 \vdash_{p_2} x_9][x_8 = n_A]A^2 \\
A^2 ::= & [(x_9, k_B) \vdash_{enc} x_{10}][(A, B, x_{10}) \vdash_{pair} x_{11}]\overline{c_3}(x_{11}).\mathbf{0}
\end{array}
$$

$$
\begin{array}{ll}
B ::= & c_1(y_1).[y_1 \vdash_{p_1} y_2][y_1 \vdash_{p_2} y_3][y_1 \vdash_{p_3} y_4][y_3 = B] \\
 & [(y_4, k_B^{-1}) \vdash_{dec} y_5][y_5 \vdash_{p_1} y_6][y_5 \vdash_{p_2} y_7][y_7 = y_2]B^1 \\
B^1 ::= & [((y_6, n_B), k_{y_2}) \vdash_{enc} y_8][(B, y_2, y_8) \vdash_{pair} y_9]\overline{c_2}(y_9).B^2 \\
B^2 ::= & c_3(y_{10}).[y_{10} \vdash_{p_1} y_{11}][y_{10} \vdash_{p_2} y_{12}][y_{10} \vdash_{p_3} y_{13}][y11 = y_2][y_{12} = B] \\
 & [(y_{13}, k_B^{-1}) \vdash_{dec} y_{14}][y_{14} = n_B]\mathbf{0}
\end{array}
$$

2.2.4 Modèle de l'attaquant

Lorsqu'on analyse un protocole cryptographique, nous devons supposer que ce dernier s'exécute dans un environnement hostile, c'est-à-dire, l'existence d'un attaquant externe qui exerce un contrôle total sur le réseau de communication. Typiquement, cet attaquant est un processus qui essaie d'attaquer le protocole en interagissant avec ce dernier à travers ses canaux de communication publics. Il peut donc intercepter et modifier tout message envoyé sur un canal public ou envoyer tout message qu'il connaît ou sait créer. Ainsi, un attaquant est généralement identifié comme étant un processus quelconque du modèle. Mais de tels attaquants peuvent s'avérer trop puissants si on ne leur impose pas certaines restrictions. En effet, aucun attaquant ne devrait connaître initialement un secret avant d'avoir attaqué le protocole. Considérons le protocole d'authentification

$$(\overline{c}(pw)|c(x).c'(y)[y = x].\overline{c'}(ok).\mathbf{0})\backslash\{c\}$$

qui reçoit un mot de passe pw à travers un canal privé c, un message m à travers un canal public c' et transmet ok sur c' si $pw = m$ pour indiquer que l'authentification est un succès. Puisque $\Pi = \overline{c'}(pw).\mathbf{0}$ est un processus du modèle, il pourrait donc être considéré comme un attaquant et ainsi réussir à se faire correctement authentifier. Pour éliminer ces faux "attaquants", une première restriction s'impose : la connaissance initiale de l'attaquant doit se limiter à ses propres données privées, et aux données publiques tels que les noms des agents, les clés publiques, etc. La deuxième contrainte est celle dite de l'hypothèse de *cryptographie parfaite* qui assume que les primitives cryptographiques sont des boites noires, i.e. l'ennemi ne peut pas attaquer un crypto-système par l'entremise de la cryptanalyse. Ainsi, un attaquant qui intercepte un cryptogramme ne peut le déchiffrer que s'il peut déduire de sa base de connaissance la clé de déchiffrement. Le modèle sera paramétré par un ensemble de règles d'inférences qui caractérisent la puissance de l'attaquant, i.e. qui déterminent

comment il pourrait créer ou inférer des messages à partir de sa base de connaissance. Étant donné un système de règles d'inférence (pour ce chapitre nous considérons, à titre d'exemple, le système engendré par les règles du Tableau 2.1), nous dirons qu'un message m est déductible d'un ensemble de messages ϕ, dénoté par $\phi \vdash m$, s'il existe un arbre de preuve de racine m dont toutes les feuilles sont des messages appartenant à ϕ et tel que tout message dans l'arbre de preuve peut être obtenu en appliquant une des règles d'inférence dont les prémisses sont ses descendants direct dans l'arbre. L'ensemble des messages déductibles de ϕ est dénoté par $\mathcal{D}(\phi)$. On suppose $\mathcal{D}(\phi)$ décidable. En effet, pour le système considéré ici (i.e. le Tableau 2.1) $\mathcal{D}(\phi)$ est décidable en temps polynomial (cf. (Abadi et Cortier, 2006)).

Définition 2.2.1 : *Soit* Π *un processus et* $ID(\Pi)$ *l'ensemble des messages (i.e. des termes clos) qui apparaissent syntaxiquement dans* Π. *Le processus* Π *est un attaquant admissible de connaissance initiale* ϕ *si et seulement si* $ID(\Pi) \subseteq \mathcal{D}(\phi)$. *L'ensemble d'attaquants admissibles de connaissance initiale* ϕ *est dénoté par*

$$\mathcal{E}nemy_{\phi} = \{\Pi \in \mathcal{P}roc \mid ID(\Pi) \subseteq \mathcal{D}(\phi)\}$$

Nous pouvons ainsi montrer que le "faux" attaquant $\overline{c'}(pw).\mathbf{0}$ n'est pas admissible pour le protocole d'authentification ci-dessus.

Proposition 2.2.1 *Soit* $\Pi = \overline{c'}(pw).\mathbf{0}$ *un attaquant de connaissance initiale* ϕ. *Alors* Π *n'est pas un attaquant admissible pour le protocole d'authentification*

$$P = (\overline{c}(pw)|c(x).c'(y)[y = x].\overline{c'}(ok).\mathbf{0})\backslash\{c\}.$$

Preuve: D'après la première contrainte, puisque pw est secret, nous devons avoir pour toute connaissance initiale ϕ de l'attaquant : $\phi \nvdash pw$. Or $ID(\overline{c'}(pw).\mathbf{0}) = \{pw\}$.

Nous avons donc $ID(\overline{c'}(pw).\mathbf{0}) \not\subseteq \mathcal{D}(\phi)$, c'est-à-dire $\overline{c'}(pw).\mathbf{0}$ n'est pas un attaquant admissible. ■

2.2.5 Équivalences observationnelles

Nous terminons la présentation du modèle CSPAD par un rappel des définitions de deux équivalences observationnelles classiques qui sont à la base de nos caractérisations de l'interférence admissible. En effet, comme signalé plus haut, l'interférence admissible non-déterministe (NAI) de Mullins, 2000, est basée sur l'équivalence de trace alors que la BNAI de Lafrance et Mullins, 2002, est basée sur la bisimulation faible.

Nous dénotons par $P \stackrel{\alpha}{\Longrightarrow} P'$ la séquence des transitions $P(\stackrel{\tau}{\longrightarrow})^* P_1 \stackrel{\alpha}{\longrightarrow} P_2(\stackrel{\tau}{\longrightarrow})^* P'$. Ainsi, si $\alpha = \tau$ alors $\stackrel{\alpha}{\Longrightarrow}$ désigne une séquence finie non vide d'actions τ. De même, si $\sigma = \alpha_1\alpha_2\cdots\alpha_n$ est une séquence d'actions (non silencieuses), nous désignons par $P \stackrel{\sigma}{\Longrightarrow} P'$ la séquence $P \stackrel{\alpha_1}{\Longrightarrow} P_1 \stackrel{\alpha_2}{\Longrightarrow} \cdots \stackrel{\alpha_n}{\Longrightarrow} P'$. Enfin la notation $P \stackrel{\hat{a}}{\Longrightarrow} P'$ signifie $P \stackrel{a}{\Longrightarrow} P'$ si $a \neq \tau$ et $P(\stackrel{\tau}{\longrightarrow})^* P'$ autrement.

Définition 2.2.2 : *[Equivalence de trace]*
- *L'ensemble des traces observables de P est $Tr(P) = \{\sigma \in \mathcal{V}is^* \mid \exists_{P' \in \mathcal{P}roc}\ P \stackrel{\sigma}{\Longrightarrow} P'\}$*
- *Deux processus P et Q sont trace-équivalents, dénoté par $P \simeq Q$ si et seulement si $Tr(P) = Tr(Q)$.*

Définition 2.2.3 : *[Bisimulation faible] Une relation binaire $\mathcal{R}$ sur l'ensemble des processus est une bisimulation faible si pour tout $(P, Q) \in \mathcal{R}$ on a :*
- *si $P \stackrel{a}{\longrightarrow} P'$, alors il existe un processus Q' tel que $Q \stackrel{\hat{a}}{\Longrightarrow} Q'$ et $(P', Q') \in \mathcal{R}$.*
- *si $Q \stackrel{a}{\longrightarrow} Q'$, alors il existe un processus P' tel que $P \stackrel{\hat{a}}{\Longrightarrow} P'$ et $(P', Q') \in \mathcal{R}$.*

Deux processus P et Q sont (faiblement) bisimulaires, dénoté par $P \approx Q$, si et seulement s'il existe une bisimulation faible $\mathcal{R}$ tel que $(P, Q) \in \mathcal{R}$.

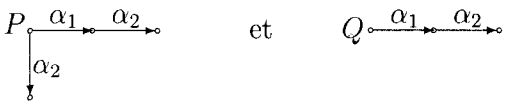

FIGURE 2.1 *Processus* CSPAD P *et* Q.

2.3 Interférence admissible

Nous présentons ici les caractérisations de l'interférence admissible permettant de spécifier les propriétés de sécurité des systèmes de sécurité à flux d'information admissible évoluant dans un environnement hostile. Nous commençons par rappeler les caractérisations de la non-interférence introduite par Focardi et Gorrieri, 2001 et comment elles permettent de manière naturelle de spécifier les propriétés de sécurité des systèmes de sécurité dans un environnement hostile. Ensuite, nous allons étendre ces définitions pour prendre en compte les systèmes de sécurité à flux d'information admissible évoluant dans un environnement hostile.

2.3.1 Non-interférence

Soit un processus P et H et L respectivement les ensembles d'actions de haut et de bas niveaux formant une partition de l'ensemble $\mathcal{V}is = \mathcal{A}ct \setminus \{\tau\}$ des actions visibles, on dit que H cause de l'*interférence* sur L (à l'intérieur de P) s'il existe des actions de H (dans P) qui causent l'occurrence d'actions de L et qui autrement ne se seraient pas produites. Par exemple, dans les processus $P = \overline{c_h}(a).\overline{c_l}(b).\mathbf{0} + \overline{c_l}(b).\mathbf{0}$ et $Q = \overline{c_h}(a).\overline{c_l}(b).\mathbf{0}$ illustrés par la Figure 2.1, en supposant que $\alpha_1 = \overline{c_h}(a) \in H$ et $\alpha_2 = \overline{c_l}(b) \in L$, α_1 cause de l' interférence sur α_2 dans le processus Q, mais pas dans le processus P.

Un processus P est dit de haut (respectivement de bas) niveau si toutes les actions visibles de P sont dans H (respectivement dans L). L'ensemble des processus de haut

(respectivement de bas) niveau est dénoté par $\mathcal{P}roc_H$ (respectivement $\mathcal{P}roc_L$). Ces processus sont évidemment sécuritaires par construction puisque leurs actions sont confinées à l'intérieur d'un même niveau de sécurité. On s'intéresse donc aux processus qui ne sont ni dans l'un ou l'autre de ces deux ensembles.

On peut reformuler la non-interférence (de H sur L à l'interieur de P) en termes d'observations relatives à un niveau : H n'interfère pas sur L (à l'intérieur de P) si tout *comportement observable* de P au niveau L uniquement est un comportement du processus où L et H sont observables. Il existe plusieurs caractérisations intéressantes de la non-interférence dans la littérature. Ici, nous nous intéressons particulièrement à celle donnée par Focardi et Gorrieri, 2001, qui s'exprime comme suit :

$$P \text{ est non-interférent si et seulement si } \forall_{\Pi \in \mathcal{P}roc_H} : (P|\Pi)\backslash H \sim P\backslash H \qquad (2.1)$$

où $\sim$ est une relation d'équivalence de comportements observables[5]. Intuitivement, $P\backslash H$ représente le comportement observable de P isolé au niveau L alors que $(P|\Pi)\backslash H$ représente son comportement au niveau H. L'idée de base est que P est sécuritaire (non-interférent) si le comportement de P en isolation complète au niveau L est le même que celui de P en interaction avec n'importe quel processus de haut niveau. En d'autres termes aucun processus de haut niveau ne peut modifier le comportement de P tel qu'observé au bas niveau.

Pour les systèmes de sécurité évoluant dans un environnement hostile, il est donc naturel de considérer les processus de haut niveau comme étant les processus ennemis puisqu'on voudrait qu'aucun attaquant ne puisse modifier les propriétés du système. Et, puisque les processus ennemis interagissent avec le protocole à travers ses canaux de communication publics, il est naturel de considérer les actions sur les canaux publics comme les actions de haut niveau. Ainsi reformulée, la propriété de la non-

[5]Nous nous limitons aux relations d'équivalences définies dans la section précédente

interférence exprime la sécurité d'un protocole comme suit :

Un protocole est sécuritaire si son "comportement observable" est le même lorsqu'il évolue dans un environnement sûr ou n'importe quel environnement hostile.

Mais, comme un processus est complètement déterminé par ses actions internes (actions τ) et ses actions de communication sur ses canaux publics, alors l'équation 2.1 serait toujours vraie puisque de deux cotés on a des processus réduits aux actions internes. Ainsi, la propriété n'aurait aucun sens. C'est pourquoi la spécification du protocole sera décorée par des actions de bas niveau qui exprime la propriété voulue. Ainsi, par exemple si on considère le protocole d'authentification de Needham-Schroeder (Tableau 2.3), sa spécification sera décorée par les actions de bas niveau $\overline{init}(X,Y)$ et $\overline{commit}(X,Y)$ qui expriment respectivement "le principal X a (ou pense avoir) initié le protocole avec le principal Y" et "X pense avoir authentifié Y" comme suit :

TABLEAU 2.4 *Needham-Schroeder avec actions de bas niveau*

$$
\begin{array}{ll}
A ::= & \overline{init}(A,B).[((n_A,A),k_B) \vdash_{enc} x_1][(A,B,x_1) \vdash_{pair} x_2]\overline{c_1}(x_2).A^1 \\
A^1 ::= & c_2(x_3).[x_3 \vdash_{p_1} x_4][x_3 \vdash_{p_2} x_5][x_3 \vdash_{p_3} x_6][x_4 = B][x_5 = A] \\
 & [(x_6,k_A^{-1}) \vdash_{dec} x_7][x_7 \vdash_{p_1} x_8][x_7 \vdash_{p_2} x_9][x_8 = n_A]\overline{commit}(A,B).A^2 \\
A^2 ::= & [(x_9,k_B) \vdash_{enc} x_{10}][(A,B,x_{10}) \vdash_{pair} x_{11}]\overline{c_3}(x_{11}).\mathbf{0}
\end{array}
$$

$$
\begin{array}{ll}
B ::= & c_1(y_1).[y_1 \vdash_{p_1} y_2][y_1 \vdash_{p_2} y_3][y_1 \vdash_{p_3} y_4][y_3 = B] \\
 & [(y_4,k_B^{-1}) \vdash_{dec} y_5][y_5 \vdash_{p_1} y_6][y_5 \vdash_{p_2} y_7][y_7 = y_2]\overline{init}(B,y_2).B^1 \\
B^1 ::= & [((y_6,n_B),k_{y_2}) \vdash_{enc} y_8][(B,y_2,y_8) \vdash_{pair} y_9]\overline{c_2}(y_9).B^2 \\
B^2 ::= & c_3(y_{10}).[y_{10} \vdash_{p_1} y_{11}][y_{10} \vdash_{p_2} y_{12}][y_{10} \vdash_{p_3} y_{13}][y11 = y_2][y_{12} = B] \\
 & [(y_{13},k_B^{-1}) \vdash_{dec} y_{14}][y_{14} = n_B]\overline{commit}(B,y_2).\mathbf{0}
\end{array}
$$

Nous nous limitons pour l'instant à cet exemple. La spécification et la validation seront abordées en profondeur dans le prochain chapitre.

Maintenant, dans l'équivalence 2.1, si on instancie la relation d'équivalence $\sim$ respectivement par l'équivalence de trace et de la bisimulation on obtient les définitions des caractérisations NDC (Non-Deductible on Composition) et BNDC (Bisimulation-based Non-Deductible on Composition) de la non-interférance. Formellement, on a :

Définition 2.3.1 : *Soit ϕ un ensemble de messages, P un protocole* CSPAD *et H l'ensemble d'actions de haut niveau. Alors*

$$P \in \mathrm{NDC}_\phi \ ssi \ \forall_{\Pi \in \mathcal{E}nemy_\phi} : (P|\Pi)\backslash H \backsimeq P\backslash H$$

Définition 2.3.2 : *Soit ϕ un ensemble de messages, P un protocole* CSPAD *et H l'ensemble d'actions de haut niveau. Alors*

$$P \in \mathrm{BNDC}_\phi \ ssi \ \forall_{\Pi \in \mathcal{E}nemy_\phi} : (P|\Pi)\backslash H \approx P\backslash H$$

Par exemple pour les processus de la Figure 2.1, on a $P \in \mathrm{NDC}_\phi$, $P \in \mathrm{BNDC}_\phi$, $Q \notin \mathrm{NDC}_\phi$ et $Q \notin \mathrm{BNDC}_\phi$ quelque soit ϕ. Par contre le processus $R = \overline{c_h}(a).\mathbf{0}+\overline{c_l}(b).\mathbf{0}$ est NDC_ϕ mais pas BNDC_ϕ. En effet, si on prend l'attaquant $\Pi = \overline{c_h}(x).\mathbf{0}$ alors on a $(R|\Pi)\backslash H \equiv \tau.\mathbf{0} + \overline{c_l}(b).\mathbf{0}$ et $R\backslash H \equiv \overline{c_l}(b).\mathbf{0}$ qui sont trace-équivalents mais non bisimulaires.

2.3.2 Interférence admissible

Soit, maintenant, un ensemble fermé $D \subseteq \mathcal{V}is$ d'actions déclassifiantes t.q. H, L et D forment une partition de $\mathcal{V}is$, l'*interférence admissible* est une propriété de flux d'information qui exige que les seules interférences de H sur L le soient uniquement via D. En d'autres termes un processus P satisfait la propriété d'interférence admissible

si en n'exécutant aucune de ses actions déclassifiantes (et donc se comportant comme $P \setminus D$), il satisfait la non-interférence. Ainsi, en remplaçant le processus P par le processus $P \setminus D$ dans la relation 2.1, on obtient une caractérisation générale de l'interférence admissible comme suit :

$$P \text{ satisfait l'interférence admissible ssi } \forall_{\Pi \in \mathcal{P}roc_H} : ((P \setminus D)|\Pi)\setminus H \sim (P \setminus D)\setminus H \quad (2.2)$$

Puisque $\Pi \in \mathcal{P}roc_H$ alors $((P \setminus D)|\Pi)\setminus H \equiv (P|\Pi)\setminus(D \cup H)$. De même $(P \setminus D)\setminus H \equiv P\setminus(D \cup H)$. La relation 2.2 est donc équivalente à

$$P \text{ satisfait l'interférence admissible ssi } \forall_{\Pi \in \mathcal{P}roc_H} : (P|\Pi)\setminus(D \cup H) \sim P\setminus(D \cup H) \quad (2.3)$$

Comme pour les actions de bas niveau, les actions de déclassifications seront des actions de décorations qui expriment exactement ce qu'on veut admettre et seront donc dépendantes du protocole analysé. En instanciant la relation 2.3 avec l'équivalence de trace et la bisimulation, nous obtenons donc les caractérisations de l'interférence admissible généralisant la NDC et la BNDC de Focardi et Gorrieri, 2001. Nous les dénotons respectivement par NDCIA[6] et BNDCIA et les définissons formellement comme suit :

Définition 2.3.3 : *Soit ϕ un ensemble de messages, P un protocole* CSPAD, *H et D les ensembles d'actions de haut niveau et de déclassification respectivement. Alors*
- $P \in \text{NDCIA}_\phi$ *ssi* $\forall_{\Pi \in \mathcal{E}nemy_\phi} : (P|\Pi)\setminus(D \cup H) \preceq P\setminus(D \cup H)$.
- $P \in \text{BNDCIA}_\phi$ *ssi* $\forall_{\Pi \in \mathcal{E}nemy_\phi} : (P|\Pi)\setminus(D \cup H) \approx P\setminus(D \cup H)$.

Bien que très pratiques pour spécifier les propriétés de correspondance comme nous allons le voir dans le prochain chapitre, ces caractérisations ne permettent pas de détecter un flux d'information non admissible qui se produirait après une déclassification. En effet, considérons le processus $P = \overline{c_h}(a).\overline{c_d}(b).\overline{c_h}(a').\overline{c_l}(b').\mathbf{0}$ où $\overline{c_h}(a), \overline{c_h}(a') \in H$,

[6]pour National Defense Central Inteligence Agency :) !

$\overline{c_d}(b) \in D$ et $\overline{c_l}(b') \in L$. Il est clair que $P \in \text{NDCIA}_\phi$ et $P \in \text{BNDCIA}_\phi$ quel que soit ϕ. Cependant, la deuxième action de haut niveau, i.e. $\overline{c_h}(a')$ cause de l'interférence sur l'action de bas niveau. Pour détecter ce type d'interférence non admissible, il a été proposé dans (Mullins, 2000) d'exprimer cette propriété en termes de la non-interférence de la façon suivante : tout sous-processus P' de P n'exécutant aucune de ses actions déclassifiantes doit satisfaire la non-interférence. Avant de définir formellement cette caractérisation de l'interférence admissible dans le cadre de CSPAD, nous avons besoin des définitions auxiliaires suivantes :

Définition 2.3.4 : *Nous disons qu'un processus P' est dérivable d'un processus P, dénoté $P' \in \mathcal{D}(P)$, s'il existe un chemin σ (éventuellement vide) tel que $P \stackrel{\sigma}{\Longrightarrow} P'$.*

Définition 2.3.5 : *De même, étant donné un ensemble ϕ de messages, nous disons que (P', ϕ') est dérivable de P conformément à ϕ, dénoté $(P', \phi') \in \mathcal{D}_\phi(P)$, s'il existe un chemin $\sigma = \alpha_1 \alpha_2 \cdots \alpha_n$ (éventuellement vide) tels que $P \stackrel{\sigma}{\Longrightarrow} P'$, $\forall_{i \leq n}$ si α_i est de la forme $c(m) \in H$ alors $m \in \mathcal{D}(\phi)$ et $\phi' = \phi \cup \{m : \exists_{i \leq n} \alpha_i = \overline{c}(m) \in H\}$.*

En d'autres termes ϕ' est égale à ϕ augmenté de tous les messages émis sur les canaux publics par P jusqu'à date.

Maintenant, nous sommes en mesure de définir formellement les caractérisations de l'interférence admissible généralisant la SBNDC (Strong BNDC) de Focardi et Gorrieri, 2001.

Définition 2.3.6 : *Soit ϕ un ensemble de messages, P un protocole* CSPAD, *H et D les ensembles d'actions de haut niveau et de déclassification respectivement. Alors*

$$P \in \text{SNDCIA}_\phi \text{ ssi } \forall_{(P',\phi') \in \mathcal{D}_\phi(P)} \forall_{\Pi \in \mathcal{E}nemy_{\phi'}} : (P'|\Pi) \backslash (D \cup H) \backsimeq P' \backslash (D \cup H).$$

Définition 2.3.7 : *Soit ϕ un ensemble de messages, P un protocole* CSPAD, *H et D les ensembles d'actions de haut niveau et de déclassification respectivement. Alors*

$$P \in \text{SBNDCIA}_\phi \text{ ssi } \forall_{(P',\phi')\in\mathcal{D}_\phi(P)}\forall_{\Pi\in\mathcal{E}nemy_{\phi'}} : (P'|\Pi)\backslash(D\cup H) \approx P'\backslash(D\cup H).$$

Nous pouvons ainsi démontrer que l'interférence non admissible dans le processus $P = \overline{c_h}(a).\overline{c_d}(b).\overline{c_h}(a').\overline{c_l}(b').\mathbf{0}$ est détectée aussi bien par la SNDCIA que la SBNDCIA.

Proposition 2.3.1 *Soit P le processus* CSPAD *défini par $\overline{c_h}(a).\overline{c_d}(b).\overline{c_h}(a').\overline{c_l}(b').\mathbf{0}$ et ϕ un ensemble de messages. Alors $P \notin \text{SNDCIA}_\phi$ et $P \notin \text{SBNDCIA}_\phi$.*

Preuve: Considérons le processus $P' = \overline{c_h}(a').\overline{c_l}(b').\mathbf{0}$, l'ensemble $\phi' = \phi \cup \{a\}$ et $\sigma = \overline{c_h}(a)\overline{c_d}(b)$. On a bien $P \stackrel{\sigma}{\Longrightarrow} P'$ et $(P', \phi') \in \mathcal{D}_\phi(P)$. Maintenant considérons le processus ennemi $\Pi = \overline{c_h}(x).\mathbf{0}$, alors on a d'une part

$$(P'|\Pi)\backslash(D\cup H) \equiv \tau.\overline{c_l}(b').\mathbf{0},$$

d'autre part

$$P'\backslash(D\cup H) \equiv \mathbf{0}.$$

Puisque $\tau.\overline{c_l}(b').\mathbf{0} \not\equiv \mathbf{0}$ et $\tau.\overline{c_l}(b').\mathbf{0} \not\approx \mathbf{0}$, on a bien $P \notin \text{SNDCIA}_\phi$ et $P \notin \text{SBNDCIA}_\phi$. ■

Enfin, pour conclure cette sous-section, nous avons le lemme suivant qui classe les différentes caractérisations de l'interférence admissible.

Lemme 2.3.1 *Soit P un processus et ϕ un ensemble de messages. Alors nous avons*

$$P \in \text{SBNDCIA}_\phi \Rightarrow \left\{ \begin{array}{l} P \in \text{BNDCIA}_\phi \\ P \in \text{SNDCIA}_\phi \end{array} \right\} \Rightarrow P \in \text{NDCIA}_\phi.$$

Preuve:

- Les relation $\text{SNDCIA}_\phi \subseteq \text{NDCIA}_\phi$ et $\text{SBNDCIA}_\phi \subseteq \text{BNDCIA}_\phi$ sont évidentes puisque $(P, \phi) \in \mathcal{D}_\phi(P)$.
- Les preuves $\text{SBNDCIA}_\phi \subseteq \text{SNDCIA}_\phi$ et $\text{BNDCIA}_\phi \subseteq \text{NDCIA}_\phi$ sont aussi évidentes puisque la bisimulation raffine l'équivalence de trace.
- Le processus P de la proposition précedente nous permet de conclure que $\text{BNDCIA}_\phi \not\subseteq \text{SNDCIA}_\phi$.
- Enfin, le processus $Q_0 = h.\mathbf{0} + l.\mathbf{0}$ nous permet de conclure que $\text{SNDCIA}_\phi \not\subseteq \text{BNDCIA}_\phi$.

■

2.3.3 Le plus grand attaquant

Comme signalé dans la section 2.1, l'interférence admissible telle que caractérisée dans la section précédente se heurte au problème de quantification sur tous les attaquants possibles. Ici, nous présentons une première solution partielle qui permet de contourner ce problème. C'est une solution partielle, car elle ne s'applique qu'aux caractérisations de l'interférence admissible basées sur l'équivalence de la trace.

Définition 2.3.8 : *Soit ϕ un ensemble de messages représentant la connaissance initiale des attaquants. Le plus grand attaquant de connaissance de base ϕ (PGA_ϕ) est le processus*

$$Top(\phi) = \sum_{c \in \mathcal{C}} c(y).Top(\phi \cup \{y\}) + \sum_{c \in \mathcal{C}} \sum_{m \in \mathcal{D}(\phi)} \overline{c}(m).Top(\phi)$$

La première sommation signifie que si le PGA_ϕ intercepte un nouveau message sur un canal public, il l'ajoute dans sa base de connaissance ϕ et la deuxième sommation lui permet d'envoyer tout message qu'il peut inférer à partir de sa base de connaissance

sur un canal public selon ses besoins. Le lemme qui suit montre que $Top(\phi)$ est le plus grand attaquant de connaissance initiale ϕ pour la trace.

Lemme 2.3.2 $\forall_{\Pi \in \mathcal{E}nemy_\phi}\ Tr(\Pi) \subseteq Tr(Top(\phi))$.

Preuve: Nous allons prouver que $\Re = \{(\Pi', Top(\phi')) : ID(\Pi') \subseteq \mathcal{D}(\phi')\}$ est une simulation[7] contenant la paire $(\Pi, Top(\phi))$. Le lemme résulte alors du fait que la simulation est une relation plus fine que l'équivalence de trace.

- Si $\Pi \xrightarrow{c(m)} \Pi'$ alors $ID(\Pi') = ID(\Pi) \cup \{m\}$ puisque la variable du canal de lecture c de Π a été instencié à m et donc apparaît syntaxiquement dans Π'. Or, d'après la première sommation de $Top(\phi)$, on a $Top(\phi) \xrightarrow{c(m)} Top(\phi')$ avec $\phi' = \phi \cup \{m\}$. Enfin, puisque $ID(\Pi) \subseteq \mathcal{D}(\phi)$ par définition de $\mathcal{E}nemy_\phi$ on a $ID(\Pi') \subseteq \mathcal{D}(\phi')$ et donc $(\Pi', Top(\phi')) \in \Re$.
- Si $\Pi \xrightarrow{\overline{c}(m)} \Pi'$ alors $m \in \mathcal{D}(\phi)$ par définition de $\mathcal{E}nemy_\phi$ et $ID(\Pi') \subseteq ID(\Pi)$. On a donc $ID(\Pi') \subseteq ID(\Pi) \subseteq \mathcal{D}(\phi)$. Or d'après la deuxième sommation de $Top(\phi)$, on a $Top(\phi) \xrightarrow{\overline{c}(m)} Top(\phi)$ et, par conséquent, $(\Pi', Top(\phi)) \in \Re$.
- Si $\Pi \xrightarrow{\tau} \Pi'$ alors $ID(\Pi') = ID(\Pi) \subseteq \mathcal{D}(\phi)$. Or $Top(\phi) \stackrel{\hat{\tau}}{\Longrightarrow} Top(\phi)$ et on a bien $(\Pi', Top(\phi)) \in \Re$.

■

Maintenant, nous pouvons prouver qu'il suffit de composer un protocole avec le PGA_ϕ pour vérifier s'il assure ou non l'interférence admissible non-déterministe.

Théorème 2.3.1 *Soit ϕ un ensemble de messages et P un processus. Alors nous avons les deux résultats suivants :*

1. $P \in \text{NDCIA}_\phi \Leftrightarrow (P|Top(\phi))\backslash(D \cup H) \backsimeq P\backslash(D \cup H)$.
2. $P \in \text{SNDCIA}_\phi \Leftrightarrow \forall_{(P',\phi') \in \mathcal{D}_\phi(P)}\ (P'|Top(\phi'))\backslash(D \cup H) \backsimeq P'\backslash(D \cup H)$.

Preuve: Nous allons prouver le premier point. La preuve du deuxième résultat est

[7]Nous référons le lecteur à l'article (Milner, 1989) pour les notions de simulation

similaire.

- $P \in \text{NDCIA}_\phi \Rightarrow ((P\backslash D)|Top(\phi))\backslash H \backsimeq P\backslash(D \cup H)$ est immédiat par définition de NDCIA_ϕ puisque $Top(\phi) \in \mathcal{E}nemy_\phi$.
- $P \in \text{NDCIA}_\phi \Leftarrow (P|Top(\phi))\backslash(D \cup H) \backsimeq P\backslash(D \cup H)$

  Puisque l'inclusion de trace est une précongruence pour les processus CSPAD, i.e. si $Tr(P) \subseteq Tr(Q)$, alors $\forall_{R \in \mathcal{P}roc}\ Tr(P|R) \subseteq Tr(Q|R)$ et $\forall_{\Gamma \in \mathcal{V}is} Tr(P\backslash\Gamma) \subseteq Tr(Q\backslash\Gamma)$, d'après le Lemme 2.3.2 on a :

$$\forall_{\Pi \in \mathcal{E}nemy_\phi}\ Tr((P|\Pi)\backslash(D \cup H)) \subseteq Tr((P|Top(\phi))\backslash(D \cup H))$$

  Or, par hypothèse, on a $Tr((P|Top(\phi))\backslash(D \cup H)) = Tr(P\backslash(D \cup H))$. D'où

$$Tr((P|\Pi)\backslash(D \cup H)) \subseteq Tr(P\backslash(D \cup H)).$$

  L'inclusion inverse étant évidente, on a bien

$$\forall_{\Pi \in \mathcal{E}nemy_\phi}\ Tr((P|\Pi)\backslash(D \cup H)) = Tr(P\backslash(D \cup H)),$$

  c'est-à-dire $P \in \text{SNDCIA}_\phi$.

■

2.4 Caractérisations statiques de l'interférence admissible

Les caractérisations de l'interférence admissible présentées dans la section précédente souffrent du problème de quantification sur tous les attaquants. Ce qui fait que ces caractérisations ne sont pas des méthodes de vérification effectives. Nous avons vu qu'on peut résoudre ce problème pour l'équivalence de trace grâce au plus grand attaquant. Même dans ce cas, on se ramasse avec un processus énorme et, surtout, infini. Ici, nous adaptons les techniques de Boreale *et al.*, 1999, qui permettent de représenter de

manière symbolique l'ensemble des attaquants ayant comme base de connaissance ϕ. Nous définissons des nouvelles caractérisations de l'interférence admissible basées sur l'équivalence de trace et de la bisimulation construites sur des systèmes de transitions contextuels engendrés par la sémantique contextuelle CSPAD qui suit.

2.4.1 Sémantique contextuelle

Les contraintes imposées à la puissance des attaquants nous permettent d'abstraire ces attaquants par l'ensemble ϕ de leurs connaissances et l'ensemble des règles (Tableau 2.1) qui définissent comment ils peuvent engendrer de nouveaux messages à partir de ϕ. Puisque nous considérons que l'ennemi contrôle totalement le réseau de communication, tout message envoyé sur un canal public est appris par l'attaquant et augmente donc sa base de connaissances. De même, tout message reçu, provient nécessairement de l'attaquant et doit être donc déductible de ϕ. Enfin, les communications privées (i.e. sur des canaux restreints), les actions de bas niveau et de déclassification n'affectent pas l'environnement dans lequel évolue le protocole. Nous obtenons donc le nouveau système de transitions défini par les règles du Tableau 2.5, où l'expression $\phi \rhd P \xrightarrow{\alpha} \phi' \rhd P'$ signifie que le processus P évoluant dans un environnement hostile de connaissance ϕ peut faire l'action α et atteindre le processus P' évoluant dans l'environnement de connaissance ϕ'. Cette façon statique de définir l'attaquant nous permet de définir des nouvelles caractérisations de l'interférence admissible qui évitent d'exhiber un quelconque attaquant ou de quantifier sur tous les attaquants possibles (il y en a une infinité !).

2.4.2 Equivalences contextuelles

Maintenant, nous définissons l'équivalence de trace et la bisimulation faible sur les systèmes de transitions engendrés par cette sémantique contextuelle. Nous déno-

TABLEAU 2.5 *Sémantique contextuelle de* CSPAD

$$\texttt{Output}\ \frac{P \xrightarrow{\overline{c}(m)} P' \quad \overline{c}(m) \in H}{\phi \rhd P \xrightarrow{\overline{c}(m)} \phi \cup \{m\} \rhd P'} \qquad \texttt{Input}\ \frac{P \xrightarrow{c(m)} P' \quad \phi \vdash m \quad c(m) \in H}{\phi \rhd P \xrightarrow{c(m)} \phi \rhd P'}$$

$$\texttt{Low}\ \frac{P \xrightarrow{\alpha} P' \quad \alpha \in L}{\phi \rhd P \xrightarrow{\alpha} \phi \rhd P'} \qquad \texttt{Down}\ \frac{P \xrightarrow{\alpha} P' \quad \alpha \in D}{\phi \rhd P \xrightarrow{\alpha} \phi \rhd P'}$$

$$\texttt{Tau}\ \frac{P \xrightarrow{\tau} P'}{\phi \rhd P \xrightarrow{\tau} \phi \rhd P'}$$

tons par $\phi \rhd P \stackrel{\alpha}{\Longrightarrow} \phi' \rhd P'$ la séquence des transitions $\phi \rhd P \ (\xrightarrow{\tau})^* \ \phi \rhd P_1 \xrightarrow{\alpha} \phi' \rhd P_2 \ (\xrightarrow{\tau})^* \ \phi' \rhd P'$. Ainsi, si $\alpha = \tau$ alors $\stackrel{\alpha}{\Longrightarrow}$ désigne une séquence finie non vide d'actions τ. De même, si $\sigma = \alpha_1 \alpha_2 \cdots \alpha_n$ est une séquence d'actions (non silencieuses), nous désignons par $\phi \rhd P \stackrel{\sigma}{\Longrightarrow} \phi' \rhd P'$ la séquence $\phi \rhd P \stackrel{\alpha_1}{\Longrightarrow} \phi_1 \rhd P_1 \stackrel{\alpha_2}{\Longrightarrow} \cdots \stackrel{\alpha_n}{\Longrightarrow} \phi' \rhd P'$. Enfin, la notation $\phi \rhd P \stackrel{\hat{\alpha}}{\Longrightarrow} \phi' \rhd P'$ signifie $\phi \rhd P \stackrel{\alpha}{\Longrightarrow} \phi' \rhd P'$ si $\alpha \neq \tau$ et $\phi \rhd P \ (\xrightarrow{\tau})^* \ \phi' \rhd P'$ autrement. L'ensemble de toutes les configurations de la forme $\phi \rhd P$ où $\phi \subseteq \mathcal{M}$ et $P \in \mathcal{P}roc$ est dénoté par $\mathcal{C}onf$.

Définition 2.4.1 : *[Equivalence de trace]*

- *Soit $\phi \rhd P$ une configuration. L'ensemble des traces visibles de la configuration $\phi \rhd P$ est*

$$Tr(\phi \rhd P) = \{\sigma \in \mathcal{V}is^* \mid \exists_{\phi' \rhd P' \in \mathcal{C}onf} \ \phi \rhd P \stackrel{\alpha}{\Longrightarrow} \phi' \rhd P'\}$$

- *Deux configurations $\phi \rhd P$ et $\psi \rhd Q$ sont trace-équivalentes, que nous dénotons par $\phi \rhd P \ \backsimeq \ \psi \rhd Q$ si et seulement si $Tr(\phi \rhd P) = Tr(\psi \rhd Q)$.*

Définition 2.4.2 : *[Equivalence de trace sous un environnement] Deux processus P et Q sont trace-équivalents sous l'environnement ϕ, dénoté par $P \ \backsimeq_\phi \ Q$ ssi $Tr(\phi \rhd P) = Tr(\phi \rhd Q)$.*

Définition 2.4.3 : *[Bisimulation contextuelle] Une relation binaire $\mathcal{R}$ sur l'ensemble des configurations est une bisimulation faible si pour tout $(\phi \rhd P, \psi \rhd Q) \in \mathcal{R}$ on a :*
- *si $\phi \rhd P \xrightarrow{\alpha} \phi' \rhd P'$, alors il existe une configuration $\psi', \rhd Q'$ telle que $\psi \rhd Q \stackrel{\hat{\alpha}}{\Longrightarrow} \psi' \rhd Q'$ et $(\phi' \rhd P', \psi' \rhd Q') \in \mathcal{R}$.*
- *si $\psi \rhd Q \xrightarrow{\alpha} \psi' \rhd Q'$, alors il existe une configuration $\phi', \rhd P'$ telle que $\phi \rhd P \stackrel{\hat{\alpha}}{\Longrightarrow} \phi' \rhd P'$ et $(\phi' \rhd P', \psi' \rhd Q') \in \mathcal{R}$.*

On dit que deux configurations $\phi \rhd P$ et $\psi \rhd Q$ sont (faiblement) bisimulaires, dénoté par $\phi \rhd P \approx \psi \rhd Q$, si et seulement s'il existe une bisimulation faible $\mathcal{R}$ tel que $(\phi \rhd P, \psi \rhd Q) \in \mathcal{R}$.

Définition 2.4.4 : *[Bisimulation sous un environnement] Deux processus P et Q sont (faiblement) bisimulaires dans l'environnement ϕ, dénoté par $P \approx_\phi Q$, si et seulement s'il existe une bisimulation faible $\mathcal{R}$ tel que $(\phi \rhd P, \phi \rhd Q) \in \mathcal{R}$.*

2.4.3 Interférence admissible sous un environnement ϕ

Maintenant, nous partons des caractérisations statiques SNNI (*Strong Non-deterministic Non Interference*) et BSNNI (*Bisimulation-based SNNI*) de la non-interférence de Focardi et Gorrieri, 2001, pour les étendre aux systèmes à flux d'information admissible dans le cadre de CSPAD. Avant de donner les définitions formelles de la non-interférence, nous avons besoin de définir l'opérateur de masquage P/Γ qui masque toutes les actions de P qui sont dans Γ où $\Gamma \subseteq \mathcal{V}is$.

Définition 2.4.5 : *Soit $\Gamma \subseteq \mathcal{V}is$ un ensemble d'actions visibles et P un processus. Le processus P/Γ est défini comme suit :*

$$P \xrightarrow{\alpha} P' \Rightarrow \begin{cases} P/\Gamma \xrightarrow{\alpha} P'/\Gamma & si\ \alpha \notin \Gamma \\ P/\Gamma \xrightarrow{\tau} P'/\Gamma & sinon \end{cases}$$

Comme pour l'opérateur de restriction, nous utilisons la notation P/C pour masquer toutes les actions visibles sur les canaux dans C. Avec cet opérateur, la SNNI et la BSNNI sont formellement définies dans le cadre de CSPAD comme suit :

Définition 2.4.6 : *Soit ϕ un ensemble de messages, P un protocole* CSPAD *et H l'ensemble d'actions de haut niveau. Alors*

- $P \in \text{SNNI}_\phi$ *ssi* $P/H \backsimeq_\phi P\backslash H$.
- $P \in \text{BSNNI}_\phi$ *ssi* $P/H \approx_\phi P\backslash H$.

Notons qu'ici également, pour les processus de la Figure 2.1, on a $P \in \text{SNNI}_\phi$, $P \in \text{BSNNI}_\phi$, $Q \notin \text{SNNI}_\phi$ et $Q \notin \text{BSNNI}_\phi$ quelque soit ϕ. Par contre le processus $R = \overline{c_h}(a).\mathbf{0} + \overline{c_l}(b).\mathbf{0}$ est SNNI_ϕ, mais pas BSNNI_ϕ. En effet, $R/H \equiv \tau.\mathbf{0} + \overline{c_l}(b).\mathbf{0}$ et $R\backslash H \equiv \overline{c_l}(b).\mathbf{0}$ sont trace ϕ-équivalents mais non ϕ-bisimulaires.

Puisqu'un processus P satisfait la propriété d'interférence admissible si $P \setminus D$ (i.e. P n'exécutant aucune de ses actions déclassifiantes) satisfait la non-interférence, nous avons les généralisations des SNNI et BSNNI dénotées respectivement par NAI (*Non-deterministic Admissible Interference*) et par BNAI (*Bisimulation-based* NAI) suivantes :

Définition 2.4.7 : *Soit ϕ un ensemble de messages, P un protocole* CSPAD, *H et D les ensembles d'actions de haut niveau et de déclassification respectivement. Alors*

$$P \in \text{NAI}_\phi \textit{ ssi } (P\backslash D)/H \backsimeq_\phi P\backslash(D \cup H).$$

Définition 2.4.8 : *Soit ϕ un ensemble de messages, P un protocole* CSPAD, *H et D les ensembles d'actions de haut niveau et de déclassification respectivement. Alors*

$$P \in \text{BNAI}_\phi \textit{ ssi } (P\backslash D)/H \approx_\phi P\backslash(D \cup H).$$

Comme dans le cas de la sémantique concrète, ces nouvelles caractérisations dans une sémantique contextuelle ne détectent que les premières interférences non admissibles, mais toutes celles qui ont lieu au delà d'une première déclassification ne sont pas détectées. Aussi, nous donnons les versions persistantes de ces caractérisations qui exigent que la propriété soit vérifiée dans tout état accessible du processus. En d'autres termes, tous les sous-processus de P doivent satisfaire la propriété. Nous les dénotons respectivement par SNAI (*Strong* NAI) et SBNAI (*Strong* BNAI)[8].

Définition 2.4.9 : *Soit ϕ un ensemble de messages, P un protocole* CSPAD, *H et D les ensembles d'actions de haut niveau et de déclassification respectivement. Alors*

$$P \in \mathrm{SNAI}_\phi \textit{ ssi } \forall_{\phi' \triangleright P' \in \mathcal{D}(\phi \triangleright P)} \ (P' \backslash D)/H \backsimeq_{\phi'} P' \backslash (D \cup H).$$

Définition 2.4.10 : *Soit ϕ un ensemble de messages, P un protocole* CSPAD, *H et D les ensembles d'actions de haut niveau et de déclassification respectivement. Alors*

$$P \in \mathrm{SBNAI}_\phi \textit{ ssi } \forall_{\phi' \triangleright P' \in \mathcal{D}(\phi \triangleright P)} \ (P' \backslash D)/H \approx_{\phi'} P' \backslash (D \cup H).$$

Le lemme qui suit classe les différentes caractérisations statiques de l'interférence admissible. Sa demonstration est identique à celle du Lemme 2.3.1.

[8] À noter que dans tous les articles (Mullins, 2000; Lafrance et Mullins, 2002; Lafrance et Mullins, 2003a; Lafrance et Mullins, 2003b; Brlek *et al.*, 2003; Lafrance, 2004; Brlek *et al.*, 2005; Bastien *et al.*, 2006), NAI et BNAI désignent respectivement ce que nous dénotons ici par SNAI et SBNAI. Mais, par souci de conformité avec les notations introduites dans la section précédente, nous avons souhaité adopter cette fois-ci ces notations.

Lemme 2.4.1 *Soit P un processus et ϕ un ensemble de messages. Alors, nous avons*

$$P \in \mathrm{SBNAI}_\phi \Rightarrow \left\{ \begin{array}{l} P \in \mathrm{BNAI}_\phi \\ P \in \mathrm{SNAI}_\phi \end{array} \right\} \Rightarrow P \in \mathrm{NAI}_\phi.$$

Enfin, pour conclure cette section, nous avons l'important résultat qui prouve que la sémantique contextuelle préserve les caractérisations de l'interférence admissible basées sur l'équivalence de trace, mais pas celles basées sur la bisimulation. Il permet également de classer toutes les différentes caractérisations de l'interférence admissible introduites dans ce chapitre (grâce aux Lemmes 2.3.1 et 2.4.1).

Théorème 2.4.1 *Soit P un processus et ϕ un ensemble de messages. Alors nous avons les résultats suivants :*

1. $P \in \mathrm{NDCIA}_\phi \Leftrightarrow P \in \mathrm{NAI}_\phi$.
2. $P \in \mathrm{SNDCIA}_\phi \Leftrightarrow P \in \mathrm{SNAI}_\phi$.
3. $P \in \mathrm{BNDCIA}_\phi \Rightarrow P \in \mathrm{BNAI}_\phi$.
4. $P \in \mathrm{SBNDCIA}_\phi \Rightarrow P \in \mathrm{SBNAI}_\phi$.

Preuve: Nous allons prouver les items 1, 3 et 4. La preuve de 2 découle de 1. Soit ϕ un ensemble de messages et P un processus.

1. D'après le Théorème 2.3.1, $P \in \mathrm{NDCIA}_\phi \Leftrightarrow (P|Top(\phi))\backslash(D \cup H) \backsimeq P\backslash(D \cup H)$. Il suffit donc de prouver que

$$(P|Top(\phi))\backslash(D \cup H) \backsimeq P\backslash(D \cup H) \Leftrightarrow (P\backslash D)/H \backsimeq_\phi P\backslash(D \cup H).$$

Commençons par prouver l'implication ($\Rightarrow$). Pour cela, il suffit de montrer que

$$\Re_1 = \{(P'|Top(\phi'), \phi' \rhd P') \mid P'|Top(\phi') \in \mathcal{D}(P|Top(\phi))\}$$

est une relation de simulation contenant $(P|Top(\phi), \phi \rhd P)$.

- $(P'|Top(\phi'))\backslash(D \cup H) \xrightarrow{\alpha} (P''|Top(\phi''))\backslash(D \cup H)$ et $\alpha \in L$. Dans ce cas, $Top(\phi'') = Top(\phi')$ puisque c'est un processus de haut niveau. On a donc $P' \xrightarrow{\alpha} P''$. D'après la sémantique contextuelle de CSPAD, on alors $\phi' \rhd ((P'\backslash D)/H) \xrightarrow{\alpha} \phi' \rhd ((P''\backslash D)/H)$ et on a bien $(P''|Top(\phi'), \phi' \rhd P'') \in \Re_1$.
- $(P'|Top(\phi'))\backslash(D \cup H) \xrightarrow{\tau} (P''|Top(\phi''))\backslash(D \cup H)$ avec $Top(\phi') \xrightarrow{\overline{c}(m)} Top(\phi'')$ et $P' \xrightarrow{c(m)} P''$. Dans ce cas, par définition de $Top(\phi')$, on a $m \in \mathcal{D}(\phi')$ et $\phi'' = \phi'$. Ainsi, d'après la sémantique contextuelle nous avons $\phi' \rhd (P'\backslash D) \xrightarrow{c(m)} \phi' \rhd (P''\backslash D)$. Enfin, comme $c(m) \in H$ nous avons $\phi' \rhd ((P'\backslash D)/H) \xrightarrow{\tau} \phi' \rhd ((P''\backslash D)/H)$ et on a bien $(P''|Top(\phi'), \phi' \rhd P'') \in \Re_1$.
- $(P'|Top(\phi'))\backslash(D \cup H) \xrightarrow{\tau} (P''|Top(\phi''))\backslash(D \cup H)$ avec $Top(\phi') \xrightarrow{c(m)} Top(\phi'')$ et $P' \xrightarrow{\overline{c}(m)} P''$. Dans ce cas, par définition de $Top(\phi')$, on a $\phi'' = \phi' \cup \{m\}$. Ainsi, d'après la sémantique contextuelle, nous avons $\phi' \rhd (P'\backslash D) \xrightarrow{\overline{c}(m)} \phi' \cup \{m\} \rhd (P''\backslash D)$. Enfin, comme $\overline{c}(m) \in H$ nous avons $\phi' \rhd ((P'\backslash D)/H) \xrightarrow{\tau} \phi' \cup \{m\} \rhd ((P''\backslash D)/H)$ et on a bien $(P''|Top(\phi''), \phi'' \rhd P'') \in \Re_1$.

Pour prouver l'implication inverse ($\Leftarrow$), il suffit de montrer que

$$\Re_2 = \{\phi' \rhd P', P'|Top(\phi')) \mid \exists_{\phi' \rhd P' \in Conf}\ \phi' \rhd P' \in \mathcal{D}(\phi \rhd P)\}$$

est une simulation. La preuve est similaire à celle de $\Re_1$.

2. Pour prouver que $\text{BNDCIA}_\phi \subset \text{BNAI}_\phi$, il suffit de remarquer que la relation $\Re_1$ est une bisimulation puisque $\Re_2 = \Re_1^{-1}$ et puisque $P \in \text{BNDCIA}_\phi$ implique $(P|Top(\phi))\backslash(D \cup H) \approx P\backslash(D \cup H)$, le résultat est établi. Enfin, pour voir que l'inclusion est stricte, il suffit de considérer le processus $P = h.h.l.\mathbf{0} + l.\mathbf{0}$ qui est BNAI_ϕ, mais non BNDCIA_ϕ.

■

CHAPITRE 3

VALIDATION DES PROPRIÉTÉS DE SÉCURITÉ

3.1 Introduction

L'interférence admissible, étudiée dans le chapitre précédent, est une notion de sécurité très générale et apparaît être une définition très naturelle et probablement l'une des plus rigoureuses de la sécurité en méthodes formelles. En effet, elle exige qu'en dehors des influences admissibles ou même souhaitables, le système attaqué par n'importe quel intrus se comporte de la même manière que le système en isolation complète. Ces influences admissibles ou même souhaitables peuvent être, par exemple, le fait que l'attaquant peut observer ou intercepter un cryptogramme, connaître la nature du crypto-système ou le fait d'accepter qu'une application externe puisse installer sur notre système des "cookies" inoffensifs mais utiles pour le fonctionnement. Cette définition abstraite présuppose qu'on est capable de définir formellement ces influences ou *interférences admissibles* entre l'attaquant et le système mais également ses *comportements observables corrects ou incorrects.* Comme nous l'avons vu dans le chapitre précédent, les interférences admissibles sont celles à travers des actions de déclassification, alors que les comportements observables sont ceux spécifiables par les actions de bas niveau. Or, dans notre modèle CSPAD et tous les modèles à base d'algèbres de processus, tout système est complètement déterminé par ses actions internes et ses communications avec l'environnement externe à travers les canaux publics (ses actions de haut niveau). Ainsi, tout système spécifié dans ces modèles vérifie trivialement n'importe quelle caractérisation de l'interférence admissible telle que définie dans le chapitre précédent. L'interférence admissible serait donc une définition naturelle de la sécurité, mais complètement inutile. Pour contourner cette difficulté,

nous nous inspirons de l'idée des *fonctions de décoration* introduites par Focardi, 2001. Aussi, nous allons décorer les protocoles avec des actions de déclassification et de bas niveau pour exprimer respectivement ce qu'on veut admettre et les comportements observables relativement à la propriété vérifiée. Cette décoration dépend donc non seulement du protocole analysé, mais surtout de la propriété vérifiée. Aussi, le but principal de ce chapitre est d'établir formellement les fonctions de décoration pour les propriétés de sécurité des protocoles cryptographiques telles que la confidentialité, l'authentification, l'intégrité et la non-répudiation.

3.1.1 Notre contribution

La contribution de ce chapitre comporte trois volets. Premièrement, partant de l'idée des fonctions de décoration introduite par Focardi, 2001, pour la non-interférence, nous avons défini formellement les contraintes que doivent vérifier les fonctions de décoration pour les systèmes de sécurité à flux d'information admissible. En effet, comme leur nom l'indique, ces fonctions dont le rôle est purement décoratif, ne doivent pas modifier les propriétés du système, i.e. ses comportements. Cette définition formelle nous a permis d'étendre nos caractérisations de l'interférence admissible à une notion plus générale d'interférence admissible à une décoration près. Bien que la décoration soit une technique utilisée dans des études de cas illustrant des approches formelles à flux d'information admissible (voir par exemple (Bossi *et al.*, 2004)), à notre connaissance, aucune définition formelle des fonctions de décoration pour des systèmes à flux d'information admissible n'existe dans la littérature.

Comme signalé ci-dessus, le but d'une fonction de décoration est "d'insérer" dans la spécification du protocole des actions de déclassification pour exprimer ce que l'on souhaite déclassifier ou admettre et des actions de bas niveau pour exprimer la propriété de sécurité vérifiée. Aussi, notre deuxième contribution, et de loin la plus

importante dans ce chapitre, est de définir formellement les fonctions de décoration pour les requis de sécurité suivants : *le secret et la correspondance.*

En effet, pour le requis de secret, donc la propriété de confidentialité, nous avons défini d'abord une nouvelle notion d'idéal d'un ensemble de messages secrets relativement à un environnement donné. Brièvement, l'idée est qu'un idéal d'un ensemble de messages secrets $\mathcal{S}$ est l'ensemble des messages qui, s'ils sont envoyés tels quels sur un canal public, révéleront à coup sûr à l'attaquant un secret conformément à sa base de connaissance actuelle. Par exemple, chiffrer un message secret par une clé déjà connue par l'attaquant et envoyer le cryptogramme sur un canal public révélera à coup sûr le secret. Grâce à cet idéal, nous avons défini une fonction qui à chaque état détermine si un message est potentiellement dangereux, i.e. dans l'idéal de $\mathcal{S}$ (donc non déclassifiable) ou non. Nous avons ainsi pu exprimer le secret en termes d'interférence admissible et, surtout, montrer la solidité de cette définition relativement à la définition classique du secret en termes de non-atteignabilité d'états non sécuritaires. À notre connaissance, c'est la première fois que l'équivalence entre la spécification du secret comme propriété de flux d'information et sa spécification classique en termes de propriété d'atteignabilité est formellement établie. De plus, cette approche basée sur l'idéal de $\mathcal{S}$ peut être facilement étendue pour tenir compte des secrets partiels, i.e. d'un ensemble des messages secrets $\mathcal{S}$ qui peut évoluer dynamiquement.

Nous avons ensuite défini formellement pour chacune des propriétés de correspondance (authentification, intégrité et non-répudiation) les fonctions de décoration correspondantes et avons pu ainsi les exprimer comme instances de l'interférence admissible. Notre approche simplifie grandement la formulation et la vérification de ces propriétés et paraît plus naturelle comparée à l'approche adoptée par Focardi, 2001. En effet, pour établir la correspondance entre deux ensembles d'événements E_1 et E_2, nous spécifions la réalisabilité des événements E_1 par des actions de déclassification et celle des événements E_2 par des actions de bas niveau et exigeons que les seules interférences admissibles sur le bas niveau ne soient que celles causées à travers des actions

de déclassification. Autrement dit, l'intrus ne peut interférer avec le bas niveau qu'à travers des actions admissibles.

Enfin, la troisième contribution de ce chapitre est l'analyse d'une extension du protocole SET (Secure Electronic Transaction) de Visa et MasterCard, dénommé ASET pour Anonymous and Secure Electronic Transaction (Brlek *et al.*, 2003; Brlek *et al.*, 2005). C'est un protocole qui en plus de corriger les défaux du protocole SET (Brlek *et al.*, 2006a; Brlek *et al.*, 2006b) assure l'anonymat du client envers le marchand. Nous avons montré que l'anonymat dans ce protocole peut se réduire à une confidentialité et donc être une instance de l'interférence admissible.

3.1.2 Organisation du chapitre

Le reste du chapitre se compose comme suit : la Section 3.2 présente notre généralisation de la notion d'interférence admissible. Ensuite nous présentons la spécification du requis de secret en termes d'interférence admissible dans la Section 3.3. Les requis de correspondance sont traités dans la Section 3.4. Nous terminons le chapitre par une importante étude de cas dans la Section 3.5.

3.2 Généralisation de la notion d'interférence admissible

Comme nous l'avons déjà signalé dans le chapitre précédent, un protocole est généralement complètement déterminé par ses actions internes (actions τ) et ses actions de communication sur ses canaux publics, donc des **actions de haut niveau**. De tels protocoles qui n'ont ni d'actions de déclassification ni d'actions de bas niveau vérifient trivialement n'importe quelle caractérisation de l'interférence admissible telle que définie dans le chapitre précédent. Pour réduire les propriétés de sécurité à l'interférence admissible et appliquer l'une de ses caractérisations, nous devons donc "décorer" le

protocole avec des actions de déclassification et de bas niveau. Cette décoration doit être faite de telle sorte que l'observation des actions de bas niveau exprime la propriété de sécurité vérifiée et que les actions de déclassification expriment exactement ce que l'on veut admettre relativement à la propriété vérifiée. La décoration dépend donc de la propriété vérifiée ainsi que du protocole analysé.

3.2.1 Fonctions de décoration

Une fonction de décoration ne doit pas modifier le comportement du système. Son rôle, comme indique son nom, est de décorer uniquement la spécification du système par des actions de déclassification et de bas niveau pour exprimer la propriété de sécurité vérifiée. Aussi, nous formalisons cette dernière contrainte sur les fonctions de décoration comme suit :

Définition 3.2.1 : *Une fonction* $\gamma : \mathcal{P}roc \longrightarrow \mathcal{P}roc$ *est une fonction de décoration si et seulement si*

$$\forall_{P \in \mathcal{P}roc_H} \ \gamma(P)/(D \cup L) \sim P$$

où $\sim \in \{\backsimeq, \approx\}$ *selon que la propriété vérifiée est basée sur l'équivalence de trace ou la bisimulation.*

En d'autres termes, si on masque les actions de décorations ajoutées (D et L), le processus décoré est équivalent au processus initial n'ayant aucune action visible autre que les actions de haut niveau.

Maintenant, nous allons étendre la notion d'interférence admissible à une notion plus générale d'*interférence admissible à une fonction de décoration* γ *près.*

Définition 3.2.2 : *Soit* γ *une fonction de décoration,* P *un processus et* ϕ *un environnement. Nous disons que*

TABLEAU 3.1 *Le protocole Wide Mouthed Frog*

(1)	$\mathbf{A} \rightarrow \mathbf{S}$:	$A, B, \{K_{AB}\}_{K_{AS}}$
(2)	$\mathbf{S} \rightarrow \mathbf{B}$:	$\{A, K_{AB}\}_{K_{BS}}$
(3)	$\mathbf{A} \rightarrow \mathbf{B}$:	$\{M\}_{K_{AB}}$

- $P \in \_NDCIA_{\phi}^{\gamma}$ *si* $\gamma(P) \in \_NDCIA_{\phi}$.
- $P \in \_NAI_{\phi}^{\gamma}$ *si* $\gamma(P) \in \_NAI_{\phi}$.

où $\_ \in \{\epsilon, S, B, SB\}$ *avec* ϵ *dénotant le mot vide.*

Évidemment, γ et la caractérisation d'interférence admissible doivent être basées sur la même équivalence. Par exemple, nous disons que $P \in NDCIA_{\phi}^{\gamma}$ si $\gamma(P) \in NDCIA_{\phi}$ et $P \in BNAI_{\phi}^{\gamma}$ si $\gamma(P) \in BNAI_{\phi}$.

3.2.2 Protocole de la grenouille à grande gueule

Pour illustrer les différentes propriétés de sécurité vérifiées dans ce chapitre, nous allons considérer la version simplifiée (Table 3.1) du protocole de la grenouille à grande gueule WMF (Wide Mouthed Frog) de (Burrows *et al.*, 1990), qui a pour but d'établir une clé de session pour deux participants, A et B, avec l'aide d'un serveur S. Bien que son objectif soit d'assurer la confidentialité de la clé de session, il permet également d'illustrer les autres propriétés de sécurité telles que l'authentification d'entité et l'intégrité de messages.

Dans le message (1), A envoie à S son nom, le nom du destinataire B, et une clé de session K_{AB} fraîche chiffrée avec la clé K_{AS}, partagée par A et S. En (2), S transmet la clé et le nom de la source à B, chiffrés avec la clé K_{BS}, partagée par B et S. Finalement, A envoie à B le message M chiffré avec la clé K. Au final, K_{AB} et M doivent demeurer secrets pour l'intrus. La description du protocole dans le modèle CSPAD est donné dans le Tableau 3.2.

TABLEAU 3.2 *Le protocole WMF en* CSPAD

$A ::= [(K, K_{AS}) \vdash_{enc} x_0][(A, B, x_0) \vdash_{pair} x_1]\overline{c_1}(x_1).[(M, K) \vdash_{enc} x_2].\overline{c_3}(x_2).\mathbf{0}$
$B ::= B_1 \mid B_2$ où $B_1 ::= c_2(y_0).[(y_0, K_{BS}) \vdash_{dec} y_1][y_1 \vdash_{p_1} y_2][y_1 \vdash_{p_2} y_3]$ $B_2 ::= c_3(y_4).[(y_4, y_3) \vdash_{dec} y_5].\mathbf{0}$
$S ::= c_1(z_0).[z_0 \vdash_{p_1} z_1][z_0 \vdash_{p_2} z_2][z_0 \vdash_{p_3} z_3][(z_3, K_{z_1 S}) \vdash_{dec} z_4][(z_1, z_4) \vdash_{pair} z_5]$ $[(z_5, K_{z_2 S}) \vdash_{enc} z_6].\overline{c_2}(z_6).\mathbf{0}$

3.3 Requis de secret : la confidentialité

Cette section a pour but d'établir une nouvelle caractérisation de la propriété de confidentialité comme propriété de flux d'information : plus particulièrement en terme d'interférence admissible. Pour cela, nous allons d'abord définir la confidentialité comme problème d'atteignabilité d'états non sécuritaires, i.e. des états dans lesquels un secret est déductible de la base de connaissances de l'attaquant. Ensuite, nous allons établir notre nouvelle caractérisation et prouver qu'elle est équivalente au problème d'atteignabilité.

3.3.1 Confidentialité en terme de non-atteignabilité

La propriété de confidentialité est généralement spécifiée en terme de non-atteignabilité d'états non sécuritaires (Meadows, 1996). Nous la reformulons ici dans le contexte de notre modèle CSPAD.

Définition 3.3.1 : *Soit $\mathcal{S}$ un ensemble des messages secrets*[1]*. Un protocole P assure la confidentialité des messages $\mathcal{S}$ contre tout attaquant de connaissance initiale ϕ si et seulement si*

$$\forall_{(P',\phi')\in\mathcal{D}_\phi(P)}\ \mathcal{S}\cap\mathcal{D}(\phi')=\emptyset.$$

Nous avons immédiatement le lemme suivant qui dit que si un protocole P assure la confidentialité des messages $\mathcal{S}$ contre tout attaquant de connaissance initiale ϕ alors ce dernier ne connaît initialement aucun message secret.

Lemme 3.3.1 *Si P assure la confidentialité de $\mathcal{S}$ contre tout attaquant de connaissance initiale ϕ alors $\forall_{s\in\mathcal{S}}\ \phi \not\vdash s$.*

Preuve: La preuve est immédiate, puisque $(P,\phi)\in\mathcal{D}_\phi(P)$ on a donc $\mathcal{S}\cap\mathcal{D}(\phi)=\emptyset$, c'est-à-dire, $\forall_{s\in\mathcal{S}}\ \phi \not\vdash s$. ■

Si la sémantique considérée est la sémantique contextuelle alors la Définition 3.3.1 se reformule naturellement comme suit :

Lemme 3.3.2 *Un protocole P assure la confidentialité des messages $\mathcal{S}$ dans un environnement hostile de connaissance initiale ϕ si et seulement si*

$$\forall_{\phi'\triangleright P'\in\mathcal{D}(\phi\triangleright P)}\ \mathcal{S}\cap\mathcal{D}(\phi')=\emptyset.$$

3.3.2 Confidentialité en terme d'interférence admissible

Pour réduire la confidentialité à l'interférence admissible et appliquer la Définition 3.2.2, comme signalé plus haut, on doit décorer sa spécification par des actions de déclassification et de bas niveau. Cette décoration doit être faite attentivement

[1] *$\mathcal{S}$ contient toutes les clés privées et les clés symétriques des principaux honnêtes.*

TABLEAU 3.3 *Attaque de confidentialité contre le protocole WMF*

(1)	$\mathbf{A} \rightarrow \mathbf{I(S)}$:	$A, B, \{K_{AB}\}_{K_{AS}}$
(1')	$\mathbf{I(A)} \rightarrow \mathbf{S}$:	$A, I, \{K_{AB}\}_{K_{AS}}$
(2)	$\mathbf{S} \rightarrow \mathbf{I}$:	$\{A, K_{AB}\}_{K_{IS}}$
(3)	$\mathbf{A} \rightarrow \mathbf{I(B)}$:	$\{M\}_{K_{AB}}$

pour que l'interférence admissible caractérise la propriété de confidentialité telle que exprimée dans la Définition 3.3.1. Les définitions qui suivent dans cette sous-section, serviront à atteindre cet objectif. Nous commençons par les actions de déclassification.

Notre approche consiste à déterminer s'il existe des actions des principaux honnêtes qui peuvent entraîner la divulgation d'un message confidentiel. Par exemple si l'intrus intercepte un cryptogramme $\{m\}_K$ (m et K^{-1} sont supposés non connus par l'intrus), alors la seule manière pour lui de déduire le message m est de "forcer" un principal (voir l'attaque à la Table 3.3) soit :

- à lui envoyer la clé de déchiffrage K^{-1}. C'est le cas du cryptogramme $\{M\}_{K_{AB}}$ pour lequel le serveur envoie à l'intrus la clé K_{AB} chiffrée avec la clé de l'intrus, ce qui est équivalent à lui envoyer la clé secrète en clair (conformément à la Définition 3.3.3).
- ou à déchiffrer le cryptogramme et lui envoyer le message comme ce fut le cas du cryptogramme $\{K_{AB}\}_{K_{AS}}$ que le serveur déchiffra et envoya le message (i.e. la clé K_{AB}) à l'intrus.

Ainsi, nous considérons l'envoi du cryptogramme $\{K_{AB}\}_{K_{AS}}$ comme une action saine (étant donné que K_{AS}^{-1} n'est pas connu par l'intrus), donc déclassifiable, alors que les envois de $\{K_{AB}\}_{K_{IS}}$ et $\{M\}_{K_{AB}}$ comme des actions dangereuses (non déclassifiables), puisque leurs clés de déchiffrage sont connues par l'intrus au moment de l'envoi. En effet, chaque fois qu'un protocole émet sur un canal public un message contenant (en clair ou non) un message confidentiel, alors un flux d'information (contenant le message confidentiel) du domaine secret vers le domaine public doit avoir eu lieu (puisque

initialement l'intrus ne connaît aucun message confidentiel). Ainsi, nous devons nous assurer que ce flux d'information s'est déroulé de manière saine, par exemple si le message émis est chiffré avec une clé dont la clé de déchiffrage correspondante n'est pas connue par l'intrus, i.e. non déductible de sa base de connaissance.

Actions de déclassification.

Définition 3.3.2 : *Soient m et m' deux messages. Nous définissons la relation* contient *sur l'ensemble des messages $\mathcal{M}$ (notée $m \prec m'$: lire m' contient m) comme suit :*

$$\frac{}{m \prec m} \qquad \frac{m \prec m'}{m \prec (\ldots, m', \ldots)}$$

$$\frac{m \prec m'}{m \prec h(m')} \qquad \frac{m \prec m'}{m \prec \{m'\}_K} \; K \in \mathcal{K}$$

Définition 3.3.3 : *Soient m et m' deux messages. Nous définissons la relation* contient en clair *(notée $m \prec^{\phi}_{clear} m'$) relativement à un ensemble de messages ϕ par les règles suivantes :*

$$\frac{}{m \prec^{\phi}_{clear} m}$$

$$\frac{m \prec^{\phi}_{clear} m'}{m \prec^{\phi}_{clear} (\ldots, m', \ldots)} \qquad \frac{m \prec^{\phi}_{clear} m' \quad K^{-1} \in \phi}{m \prec^{\phi}_{clear} \{m'\}_K}$$

La dernière règle de $\prec^{\phi}_{clear}$ exprime le fait que chiffrer un message avec une clé connue de l'intrus est "inutile" si on cherche à assurer la confidentialité du message. Le message chiffré est "équivalent" au message en clair (relativement à la connaissance de l'intrus). Notons que chacune de ces deux relations ($\prec$ et $\prec^{\phi}_{clear}$) définit un ordre partiel sur l'ensemble $\mathcal{M}$.

Définition 3.3.4 : *Soit $\mathcal{S}$ l'ensemble des messages confidentiels échangés dans le protocole. Nous définissons l'ensemble des* messages secrets *relativement aux connaissances actuelles ϕ de l'intrus, appelé* idéal de $\mathcal{S}$ relativement à ϕ *et dénoté $I_\phi(\mathcal{S})$, par l'ensemble*

$$I_\phi(\mathcal{S}) = \{m' \in \mathcal{M} \mid \exists_{m \in \mathcal{S}}\ m \prec^{\phi}_{clear} m'\}$$

des messages qui contiennent en clair (relativement à ϕ) un message confidentiel.

$I_\phi(\mathcal{S})$ définit l'ensemble des *messages dangereux*, i.e. des messages qui, s'ils sont envoyés tels quels révéleront un secret à l'attaquant. C'est une propriété locale puisque définie relativement à la connaissance actuelle de l'attaquant. Nous nous en servirons dans la définition qui suit pour déterminer à chaque état quels sont les messages déclassifiables, i.e. non dangereux.

Définition 3.3.5 : *Soit $\mathcal{S}$ un ensemble des messages secrets, d un canal de déclassification fixé et ϕ un ensemble des messages. La* fonction de déclassification sûre $\gamma^{\mathcal{S}}_{\phi} : \mathcal{P}roc \longrightarrow \mathcal{P}roc$ *relativement à $\mathcal{S}$ et ϕ, est la fonction définie par*

$$\forall_{P \in \mathcal{P}roc}\ P \xrightarrow{\alpha} Q \Rightarrow \begin{cases} \gamma^{\mathcal{S}}_{\phi}(P) \xrightarrow{\alpha} \overline{d}(m).\gamma^{\mathcal{S}}_{\phi \cup \{m\}}(Q) \\ \qquad si\ \alpha \in H\ est\ de\ la\ forme\ \overline{c}(m)\ et\ m \notin I_{\phi \cup \{m\}}(\mathcal{S}) \\ \gamma^{\mathcal{S}}_{\phi}(P) \xrightarrow{\alpha} \gamma^{\mathcal{S}}_{\phi \cup \{m\}}(Q) \\ \qquad si\ \alpha \in H\ est\ de\ la\ forme\ \overline{c}(m)\ et\ m \in I_{\phi \cup \{m\}}(\mathcal{S}) \\ \gamma^{\mathcal{S}}_{\phi}(P) \xrightarrow{\alpha} \gamma^{\mathcal{S}}_{\phi}(Q) \qquad sinon \end{cases}$$

Essentiellement, la fonction $\gamma^{\mathcal{S}}_{\phi}$ déclassifie les outputs du protocole sur un canal public dont les messages m émis ne présentent aucun danger, i.e. ne sont pas dans l'idéal $I_{\phi \cup \{m\}}(\mathcal{S})$. Notons que, quand le protocole émet un message "dangereux", $\gamma^{\mathcal{S}}_{\phi}$ ne le déclassifie pas, mais met à jour la connaissance de l'attaquant.

Actions de bas niveau.

Maintenant que nous savons comment décorer un protocole par des actions de déclassification, il nous reste à déterminer les actions de bas niveau et comment décorer le protocole par ces dernières. Aussi, nous allons augmenter la spécification du protocole par le processus $c_{\mathcal{S}}(x).\overline{l_{\mathcal{S}}}(x).\mathbf{0}$ où $c_{\mathcal{S}}$ est canal public de domaine $\mathcal{S}$, **ne faisant pas partie des canaux public du protocole** et $l_{\mathcal{S}}$ un canal de bas niveau de domaine $\mathcal{S}$. Ainsi, puisque $c_{\mathcal{S}}$ n'est pas un canal du protocole, tout message lu par ce canal provient nécessairement de l'intrus. Son émission sur le canal de bas niveau $l_{\mathcal{S}}$ permet à l'observateur de bas niveau de savoir que le secret émis (puisque $c_{\mathcal{S}}$ et $l_{\mathcal{S}}$ n'acceptent que les messages secrets $\mathcal{S}$) est connu par l'attaquant.

Nous sommes maintenant suffisamment outillés pour exprimer la propriété de confidentialité en termes de l'interférence admissible. Soit $\mathrm{Spec}_{\mathcal{S}} : \mathcal{P}roc \longrightarrow \mathcal{P}roc$ la fonction qui étant donné un processus P, le décore avec les actions de bas niveau comme décrit ci-dessus, i.e.

$$\forall_{P \in \mathcal{P}roc}\ \mathrm{Spec}_{\mathcal{S}}(P) = P | c_{\mathcal{S}}(x).\overline{l_{\mathcal{S}}}(x).\mathbf{0}.$$

Nous avons alors cet important résultat :

Théorème 3.3.1 *Soit P un protocole* CSPAD *et $\mathcal{S}$ un ensemble des messages secrets. P assure la confidentialité de $\mathcal{S}$ contre tout attaquant de connaissance initiale ϕ si et seulement si*

$$\mathrm{Spec}_{\mathcal{S}}(P) \in SNAI_{\phi}^{\gamma_{\phi}^{\mathcal{S}}}$$

Preuve:

$$\begin{aligned}
\mathrm{Spec}_{\mathcal{S}}(P) \in SNAI_{\phi}^{\gamma_{\phi}^{\mathcal{S}}} &\iff P|c_{\mathcal{S}}(x).\overline{l_{\mathcal{S}}}(x).\mathbf{0} \in SNAI_{\phi}^{\gamma_{\phi}^{\mathcal{S}}} \\
&\iff \gamma_{\phi}^{\mathcal{S}}(P|c_{\mathcal{S}}(x).\overline{l_{\mathcal{S}}}(x).\mathbf{0}) \in SNAI_{\phi} \\
&\iff \forall_{\psi \rhd Q \in \mathcal{D}(\phi \rhd \gamma_{\phi}^{\mathcal{S}}(P|c_{\mathcal{S}}(x).\overline{l_{\mathcal{S}}}(x).\mathbf{0}))}\ Q \in NAI_{\psi} \\
&\iff \forall_{\psi \rhd Q \in \mathcal{D}(\phi \rhd \gamma_{\phi}^{\mathcal{S}}(P|c_{\mathcal{S}}(x).\overline{l_{\mathcal{S}}}(x).\mathbf{0}))}\ (Q\backslash D)/H \backsimeq_{\psi} Q\backslash(D \cup H).
\end{aligned}$$

Puisque $c_{\mathcal{S}}(x)$ est un canal d'input et que $\overline{l_{\mathcal{S}}}(x)$ est un canal de bas niveau, toute action sur ces deux canaux ne modifie pas la connaissance de l'environnement. Ainsi, prouver le théorème revient à prouver que

$$\forall_{\psi \rhd Q \in \mathcal{D}(\phi \rhd \gamma_{\phi}^{\mathcal{S}}(P|c_{\mathcal{S}}(x).\overline{l_{\mathcal{S}}}(x).\mathbf{0}))}\ (Q\backslash D)/H \backsimeq_{\psi} Q\backslash(D \cup H) \iff \mathcal{S} \cap \mathcal{D}(\psi) = \emptyset.$$

Nous prouvons d'abord l'implication dans le sens ($\Longleftarrow$). En effet, supposons que

$$\forall_{\psi \rhd Q \in \mathcal{D}(\phi \rhd \gamma_{\phi}^{\mathcal{S}}(P|c_{\mathcal{S}}(x).\overline{l_{\mathcal{S}}}(x).\mathbf{0}))}\ \mathcal{S} \cap \mathcal{D}(\psi) = \emptyset.$$

Alors, on a

$$\forall_{c,m} \text{ t.q. } \overline{c}(m) \in H \text{ et } \psi \rhd Q \xrightarrow{\overline{c}(m)} \psi \cup \{m\} \rhd Q',\ m \notin I_{\psi \cup \{m\}}(\mathcal{S}).$$

En effet, si $m \in I_{\psi \cup \{m\}}(\mathcal{S})$ alors $\mathcal{S} \cap \mathcal{D}(\psi \cup \{m\}) \neq \emptyset$ dans l'état $\psi \cup \{m\} \rhd Q'$, ce qui contredirait notre hypothèse. Donc, Q n'a aucune action de haut niveau non déclassifiable, i.e. $(Q\backslash D)/H = Q\backslash(D \cup H) \Longrightarrow (Q\backslash D)/H \backsimeq_{\psi} Q\backslash(D \cup H)$.

Pour prouver l'implication inverse ($\Longrightarrow$), nous allons procéder par induction sur la longeur des chemins du sommet initial $\phi \rhd \gamma_{\phi}^{\mathcal{S}}(P|c_{\mathcal{S}}(x).\overline{l_{\mathcal{S}}}(x).\mathbf{0})$ aux sommets atteignables $\psi \rhd Q$. Supposons que

$$\forall_{\psi \rhd Q \in \mathcal{D}(\phi \rhd \gamma_{\phi}^{\mathcal{S}}(P|c_{\mathcal{S}}(x).\overline{l_{\mathcal{S}}}(x).\mathbf{0}))}\ (Q\backslash D)/H \backsimeq_{\psi} Q\backslash(D \cup H).$$

[Base d'induction :] puisque la configuration initiale

$$\phi \rhd Q = \phi \rhd \gamma_\phi^{\mathcal{S}}(P|c_{\mathcal{S}}(x).\overline{l_{\mathcal{S}}}(x).\mathbf{0}) \in \mathcal{D}(\phi \rhd \gamma_\phi^{\mathcal{S}}(P|c_{\mathcal{S}}(x).\overline{l_{\mathcal{S}}}(x).\mathbf{0})),$$

alors, on a $\forall_{m \in \mathrm{dom}(c_{\mathcal{S}})}$, (i.e. $m \in \mathcal{S}$) $\phi \rhd Q \not\xrightarrow{c_{\mathcal{S}}(m)}$, ce qui implique $\mathcal{S} \cap \mathcal{D}(\phi) = \emptyset$. En effet, dans le cas contraire, i.e. s'il existe $m \in \mathcal{S}$ t.q.

$$\phi \rhd Q \xrightarrow{c_{\mathcal{S}}(m)} \phi \rhd Q',$$

alors, puisque $c_{\mathcal{S}}(m)$ n'est pas déclassifiable (ce n'est même pas un output !), on a d'une part

$$\phi \rhd (Q \backslash D)/H \xrightarrow{\tau} \phi \rhd (Q' \backslash D)/H \xrightarrow{\overline{l_{\mathcal{S}}}(m)} \phi \rhd Q''$$

et d'autre part

$$Q \backslash (D \cup H) \backsimeq_\phi \mathbf{0},$$

ce qui contrédirait notre hypothèse.

[Étape d'induction :] supposons que

$$\psi \rhd Q \in \mathcal{D}(\phi \rhd \gamma_\phi^{\mathcal{S}}(P|c_{\mathcal{S}}(x).\overline{l_{\mathcal{S}}}(x).\mathbf{0})) \text{ t.q. } \mathcal{S} \cap \mathcal{D}(\psi) = \emptyset.$$

Nous allons prouver que pour tout état $\psi' \rhd Q'$ successeur de la configuration $\psi \rhd Q$, on a $\mathcal{S} \cap \mathcal{D}(\psi') = \emptyset$.

Soit alors $\alpha \in \mathcal{A}ct$ t.q. $\psi \rhd Q \xrightarrow{\alpha} \psi' \rhd Q'$.

- Si $\alpha \notin H$, alors $\psi' = \psi$ et donc $\mathcal{S} \cap \mathcal{D}(\psi') = \mathcal{S} \cap \mathcal{D}(\psi) = \emptyset$.
- Si $\alpha \in H$ est de la forme $\overline{c}(m)$ et que $m \notin I_{\psi \cup \{m\}}(\mathcal{S})$, alors $\psi' = \psi \cup \{m\}$ et $\mathcal{S} \cap \mathcal{D}(\psi') = \emptyset$.
- Enfin, si $\alpha \in H$ est de la forme $\overline{c}(m)$ et que $m \in I_{\psi \cup \{m\}}(\mathcal{S})$, alors $\psi' = \psi \cup \{m\}$ et $\mathcal{S} \cap \mathcal{D}(\psi') \neq \emptyset$. Nous allons prouver que compte tenu de nos hypothèses, ce

dernier cas n'est pas possible. En effet, d'une part, si le canal d'input $c_{\mathcal{S}}(x)$ est encore disponible alors le même raisonnement que celui de la base de l'induction permet de conclure que $Q'\backslash D)/H \not\approx_{\phi'} Q'\backslash(D \cup H)$ contredisant notre hypothèse initiale. D'autre part, si $c_{\mathcal{S}}(x)$ n'est plus disponible, alors il existe un état $\psi_0 \rhd Q_0$ précédant $\psi' \rhd Q'$ et un message $m_0 \in \mathcal{S}$ t.q. $\psi_0 \rhd Q_0 \xrightarrow{c_{\mathcal{S}}(m_0)}$. Mais, puisque le prédécesseur immédiat de $\psi' \rhd Q'$ est $\psi \rhd Q$ alors $\psi_0 \subseteq \psi$, ce qui implique que $m_0 \in \mathcal{D}(\psi)$ contrédisant l'hypothèse de base $\mathcal{S} \cap \mathcal{D}(\psi) = \emptyset$.

Donc, dans tous les cas, nous avons bien $\mathcal{S} \cap \mathcal{D}(\psi') = \emptyset$. ■

Corollaire 3.3.1 *Soit P un protocole* CSPAD *et $\mathcal{S}$ un ensemble des messages secrets. P assure la confidentialité de $\mathcal{S}$ contre tout attaquant de connaissance initiale ϕ si et seulement si*

$$\mathrm{Spec}_{\mathcal{S}}(P) \in SNDCIA_{\phi}^{\gamma_{\phi}^{\mathcal{S}}}$$

Preuve: Découle directement du Théorème 2.4.1. ■

Corollaire 3.3.1 *Soit $\circ$ l'opérateur de composition des fonctions. P assure la confidentialité de $\mathcal{S}$ contre tout attaquant de connaissance initiale ϕ si et seulement si*

$$P \in SNAI_{\phi}^{\gamma_{\phi}^{\mathcal{S}} \circ \mathrm{Spec}_{\mathcal{S}}} \iff P \in SNDCIA_{\phi}^{\gamma_{\phi}^{\mathcal{S}} \circ \mathrm{Spec}_{\mathcal{S}}}$$

Preuve: D'après la Définition 3.2.2 on a :

$$\begin{aligned} P \in SNAI_{\phi}^{\gamma_{\phi}^{\mathcal{S}} \circ \mathrm{Spec}_{\mathcal{S}}} &\iff \gamma_{\phi}^{\mathcal{S}} \circ \mathrm{Spec}_{\mathcal{S}}(P) \in SNAI_{\phi} \\ &\iff \gamma_{\phi}^{\mathcal{S}}(\mathrm{Spec}_{\mathcal{S}}(P)) \in SNAI_{\phi} \\ &\iff \mathrm{Spec}_{\mathcal{S}}(P) \in SNAI_{\phi}^{\gamma_{\phi}^{\mathcal{S}}}. \end{aligned}$$

■

Par exemple, si m est un message secret, c un canal public, K une clé de chiffrement et ϕ la base de connaissance initiale de l'attaquant telle que $\{m, K^{-1}\} \cap \mathcal{D}(\phi) = \emptyset$. Alors, on a :

- Le processus $P = \overline{c}(m).\mathbf{0}$ qui émet le secret sur le canal public n'assure pas la confidentialité de m.
- Le processus $Q = \overline{c}(\{m\}_K).\mathbf{0}$ qui émet le cryptogramme du secret sur le canal public assure la confidentialité de m.

En effet, dans le premier cas, on a $\mathcal{S} = \{m\}$ et puisque la seule action d'output de P est l'émission de m qui est dans l'idéal $I_{\phi \cup \{m\}}(\mathcal{S})$, nous avons

$$\gamma^{\mathcal{S}}_{\phi_0}(\mathrm{Spec}_{\mathcal{S}}(P)) = \overline{c}(m).\mathbf{0} | c_{\mathcal{S}}(x).\overline{l_{\mathcal{S}}}(x).\mathbf{0},$$

qui n'est pas $SNAI_\phi$ sécuritaire. Dans le deuxième cas $\mathcal{S} = \{m, K^{-1}\}$ et la seule action d'output de Q est l'émission de $\{m\}_K$ qui n'est pas dans l'idéal $I_{\phi \cup \{m\}_K}(\mathcal{S})$. Nous avons donc

$$\gamma^{\mathcal{S}}_{\phi_0}(\mathrm{Spec}_{\mathcal{S}}(P)) = \overline{d}(\{m\}_K).\overline{c}(\{m\}_K).\mathbf{0} | c_{\mathcal{S}}(x).\overline{l_{\mathcal{S}}}(x).\mathbf{0},$$

qui est $SNAI_\phi$ sécuritaire.

Maintenant, nous pouvons montrer que l'interférence admissible permet de détecter la faille de confidentialité dans le protocole de la grenouille à grande gueule. Soit $WMF = A|B|S$ le processus qui spécifie le protocole WMF, $\phi_0 = \{A, B, I, K_{IS}\}$ l'ensemble des connaissances initiales de l'intrus et $\mathcal{S} = \{M, K_{AB}, K_{AS}, K_{BS}\}$ l'ensemble des messages confidentiels. Alors, nous avons la proposition suivante qui prouve que P n'assure pas la propriété de confidentialité.

Proposition 3.3.1

$$\mathrm{Spec}_{\mathcal{S}}(WMF) \notin SNAI^{\gamma^{\mathcal{S}}_{\phi_0}}_{\phi_0}$$

Preuve: En effet $\phi \rhd Q \in \mathcal{D}(\phi_0 \rhd \gamma^{\mathcal{S}}_{\phi_0}(\mathrm{Spec}_{\mathcal{S}}(WMF)))$ avec

$$Q = \mathbf{0}|B|\overline{c_2}(\{(A, K_{AB})\}_{K_{IS}}).\mathbf{0}|c_{\mathcal{S}}(x).\overline{l_{\mathcal{S}}}(x).\mathbf{0},\ \phi = \{A, B, I, K_{IS}, \{K_{AB}\}_{KAS}, \{M\}_{K_{AB}}\}$$

Mais, Q ne satisfait pas NAI_ϕ. La configuration[2] $\phi \rhd Q$ désigne l'état atteint quand A a envoyé ses deux messages interceptés par l'intrus qui transmet le premier message au serveur après avoir remplacé le nom de B par le sien. Le serveur est prêt à envoyer la clé K_{AB} à l'intrus à travers le canal c_2. Q ne satisfait pas NAI_ϕ à cause du fait que le processus $Q\backslash(D \cup H) \backsimeq_\phi \mathbf{0}$, alors que masquer les actions de haut niveau (processus $(Q\backslash D)/H$) permettra d'observer les actions de bas niveau $\overline{l_{\mathcal{S}}}(K_{AB})$ et $\overline{l_{\mathcal{S}}}(M)$. Donc les deux processus ne sont pas trace-équivalent dans l'environnement ϕ. ■

3.4 Requis de correspondance

La correspondance décrit les dépendances entre les événements qui se produisent dans le protocole. Elle est habituellement utilisée pour exprimer les propriétés d'authentification, d'intégrité et de non-répudiation. L'idée est que certains événements observables sont des conséquences d'autres événements. L'idée de la dépendance est que ces événements observables ne se réalisent que seulement si les événements qui les causent se sont préalablement réalisés, condition nécessaire mais non suffisante. Ce sont des propriétés de sûreté que nous allons pouvoir exprimer comme propriété de flux d'information. Notre approche consiste à spécifier la réalisabilité d'un événement observable par des actions de bas niveau et celle d'un événement causal par des actions de déclassification. Ainsi, les événements observables (actions de bas niveau) sont des conséquences des actions de déclassification. Autrement dit, si on supprime

[2] À noter que $\gamma^{\mathcal{S}}_{\phi_0}$ ne modifie pas les actions de B pour la simple raison que ce dernier n'a aucune action d'outpout sur un canal public ; ce qui explique la présence de B dans le processus Q. Noter également que l'action $\overline{c_2}(\{(A, K_{AB})\}_{K_{IS}})$ n'est pas déclassifiée, car $\{(A, K_{AB})\}_{K_{IS}}$ n'est pas dans $I_{\phi \cup \{(A.K_{AB})\}_{K_{IS}}}(\mathcal{S})$.

les actions de déclassification, aucun événement observable ne devrait avoir lieu. Par exemple, dans le cas d'une authentification de A par B, il suffit d'ajouter dans la spécification du protocole l'action de bas niveau $\overline{auth_B}(A)$ qui exprime que "B pense avoir authentifier A" et l'action de déclassification $\overline{init_A}(B)$ qui exprime que "A a initié ou pense avoir initié le protocole avec B". Ainsi, si B pense avoir authentifié A, alors forcement au préalable, A a initié le protocole avec B.

Ces propriétés de correspondance exigent donc que toute occurrence d'une action de bas niveau soit précédée par l'occurrence d'au moins une action de déclassification. En d'autres termes, les traces observables doivent garantir cette précédence des actions de déclassification sur celles de bas niveau. Pour exprimer cette précédence, (Focardi, 2001) a dû introduire un nouveau processus énorme et surtout infini (du genre du plus grand attaquant PGA_ϕ défini dans la Section 2.3.3) dans sa formulation de ces propriétés de correspondance en termes de la non-interférence. Ici, notre approche basée sur l'interférence admissible simplifie grandement la formulation de ces propriétés, comme nous allons le voir. Tout d'abord, nous allons imposer une contrainte supplémentaire pour les fonctions de décoration du requis de correspondance. En effet, nous devons nous assurer que la correspondance entre ces deux ensembles d'événements est bien établie par la décoration dans un environnement sûr (i.e. pour tout système non attaqué). Cette contrainte est formellement établie dans la définition suivante :

Définition 3.4.1 : *Soit E_D et E_L deux ensembles d'événements. Une fonction de décoration γ est admissible pour établir la dépendance des événements E_L par rapport aux événement E_D si et seulement si*

$$\forall_{P \in \mathcal{P}roc_H} \ (\gamma(P) \backslash D)/H \backsimeq P/H$$

et que l'occurrence des actions de déclassification et de bas niveau de $\gamma(P)$ expriment respectivement la réalisabilité des événements E_D et E_L.

Puisque les actions visibles de $P \in \mathcal{P}roc_H$ sont toutes des actions de haut niveau, alors l'ensemble des traces observables de P/H est vide. Par conséquent, l'équivalence $(\gamma(P)\backslash D)/H \backsimeq P/H$ nous garantit que si on supprime les actions de déclassification, alors aucune action de bas niveau n'est réalisable, ce qui établit la correspondance entre les deux. Cette contrainte est nécessaire pour établir la correspondance. En effet, considérons le processus $P = \overline{c_h}(a).\mathbf{0}$ et une fonction de décoration γ telle que

$$\gamma(P) = \overline{c_h}(a).\overline{c_d}(a).\overline{c_l}(b).\mathbf{0} + \overline{c_l}(b).\mathbf{0}.$$

γ n'est pas admissible pour établir la correspondance entre les actions de déclassification et celles de bas niveau, puisque la seule action de déclassification $\overline{c_d}(a)$ n'interfère pas avec l'unique action de bas niveau $\overline{c_l}(b)$.

Au contraire de la confidentialité pour laquelle nous avons exigé que la déclassification soit contrôlée pour tout état atteignable, ici, nous ne pouvons pas exiger que les conditions de déclassification (événements causes) soient vérifiées pour tout état atteignable. Ainsi, la réalisabilité d'une condition de déclassification déclassifie tout ce qui vient après. Aussi, les requis de correspondance vont être exprimés en terme de la NAI_ϕ. Formellement nous exprimons les requis de correspondance comme suit :

Définition 3.4.2 : *Soit P un protocole* CSPAD, *ϕ un environnement, E_D et E_L deux ensembles d'événements. Le protocole P assure la dépendance*[3] *des événements E_L par rapport aux événements E_D dans l'environnement ϕ si et seulement s'il existe une fonction de décoration γ admissible pour E_D et E_L telle que $P \in NAI_\phi^\gamma$.*

Évidemment, les événements E_D et E_L dépendent non seulement des propriétés de

[3] *Nous supposons que la dépendance est globale, i.e. la réalisabilité d'un événement dans E_L est conditionnée à la réalisabilité d'un événement quelconque de E_D. S'il existe un sous-ensemble $E_{L_1} \subset E_L$ dont la dépendance est uniquement liée à un sous-ensemble $E_{D_1} \subset E_D$, il suffit de considérer une fonction de décoration qui exprime exactement la dépendance entre ces deux sous-ensembles.*

sécurité vérifiées (authentification, intégrité et non-répudiation), mais aussi du protocole analysé. Dans les sous-sections qui suivent, nous allons examiner chacune des trois propriétés et donner quelques exemples de décoration.

3.4.1 Authentification d'entité

Soient A et B deux prétendants légitimes, C un tiers et I un opposant (espion). Les principaux objectifs des protocoles d'authentification d'entité (prétendant) sont :

Non-transférabilité : le prétendant légitime A (respectivement B) ne peut réutiliser un échange d'authentification avec B(respectivement A) pour se faire passer pour B(respectivement A) auprès d'un tiers C.

Imposture : il est pratiquement impossible que l'opposant I puisse effectuer le protocole avec A (respectivement B) et réussir à le convaincre en se faisant passer pour B (respectivement A).

L'authentification est unilatérale ou mutuelle selon l'application. Dans ce qui suit, nous allons illustrer l'authentification unilatérale, i.e. dans le cas où un principal A veut se faire authentifier par un principal B. Pour l'authentification mutuelle, il suffit de faire l'analyse dans les deux sens : B authentifie A et ensuite A authentifie B. Les événements causes E_D expriment généralement l'intention de l'initiateur du protocole de se faire authentifier par un principal particulier. Cette intention est généralement identifiable grâce au message envoyé pour initier l'authentification : les identités des principaux contenues dans le message, la clé utilisée pour chiffrer, etc. Les événements de bas niveau E_L expriment plutôt la croyance de l'authentificateur. Cette croyance est généralement identifiable par un ensemble d'états acceptants, i.e. quand le système arrive dans un état où l'authentificateur B pense avoir identifié le prétendant A. Un état acceptant peut être atteint à la suite de la réception par B d'un message particulier ou à la suite d'un certain nombre de défi-réponses entre les deux prétendants dépendamment de l'application. Aussi, nous supposons à partir de maintenant

qu'il existe une fonction de décoration γ_{Auth} qui, étant donné un processus le décore avec les actions de bas niveau et de déclassification nécessaires. Étant donné un protocole P, il revient donc au concepteur ou celui qui fait l'analyse de déterminer le(s) message(s) qui exprime l'intention de A de se faire authentifier par un principal en particulier et l'état (ou les états) acceptant qui exprime la croyance de l'authentificateur. Une fois ce choix fait, l'analyse lui permet de déterminer si le protocole assure l'authentification de A par B dans ces conditions. À titre d'illustration, nous allons examiner le protocole WMF utilisé comme protocole d'authentification de A par B.

En effet, nous pouvons essayer d'utiliser le protocole WMF pour faire de l'authentification d'entité comme suit : l'identité de B contenue dans le premier message $(A, B, \{K\}_{K_{AS}})$ de A exprime le fait que A veut se faire authentifier par B et la réception à la fois des messages $\{A, K\}_{K_{BS}}$ et $\{M\}_K$ permet à B de croire qu'il communique bien avec A. En effet, la réception du message $\{A, K\}_{K_{BS}}$ lui assure que, d'après le serveur, en qui il a confiance, la clé secrète K provient bien de A et la réception d'un message chiffré par cette clé finit de le convaincre qu'il parle bien à A. Aussi, pour s'assurer que la protocole assure la "non-transférabilité", nous allons légèrement modifier la spécification du prétendant A pour lui permettre d'initier le protocole de manière légitime avec n'importe quel principal X comme suit :

$$A(X) ::= [(K, K_{AS}) \vdash_{enc} x_0][(A, X, x_0) \vdash_{pair} x_1]\overline{c_1}(x_1).[(M, K) \vdash_{enc} x_2].\overline{c_3}(x_2).\mathbf{0}$$

La décoration $\gamma_{Auth}(WMF)$ du protocole d'authentification WMF dans le cas d'une *vérification de l'authentification du prétendant A par le principal B* est illustrée dans le Tableau 3.4. La spécification de A est décorée par l'action de $\overline{init_A}(B)$ qui déclassifie tout ce qui vient après à condition que la garde $[X = B]$, qui exprime l'intention de A de se faire authentifier par B, soit vérifiée. La spécification de B est décorée par l'action de bas niveau $\overline{auth_B}(A)$ qui exprime que B a reçu à la fois le message

TABLEAU 3.4 *Spécification de la décoration d'authentification du protocole WMF*

$A(X) ::=$	$[(K, K_{AS}) \vdash_{enc} x_0][(A, X, x_0) \vdash_{pair} x_1]([X = B]\overline{init_A}(B).A_1 + A_1)$ où $A_1 ::= \overline{c_1}(x_1).[(M, K) \vdash_{enc} x_2].\overline{c_3}(x_2).\mathbf{0}$
$B' ::=$	$B_1 \mid B_2$ où $B_1 ::= c_2(y_0).[(y_0, K_{BS}) \vdash_{dec} y_1][y_1 \vdash_{p_1} y_2][y_1 \vdash_{p_2} y_3]$ $B_2 ::= c_3(y_4).[(y_4, y_3) \vdash_{dec} y_5][y_2 = B]\overline{auth_B}(A).\mathbf{0}$
$S ::=$	$c_1(z_0).[z_0 \vdash_{p_1} z_1][z_0 \vdash_{p_2} z_2][z_0 \vdash_{p_3} z_3][(z_3, K_{z_1 S}) \vdash_{dec} z_4][(z_1, z_4) \vdash_{pair} z_5]$ $[(z_5, K_{z_2 S}) \vdash_{enc} z_6].\overline{c_2}(z_6).\mathbf{0}$

$\{A, K\}_{K_{BS}}$ et un message chiffré par la clé K, ce qui le fait croire qu'il communique avec B.

Évidemment, le protocole WMF ne garantit pas l'authentification de A par B pour la simple raison qu'il n'assure pas la non-transférabilité. En effet, lorsque le prétendant A initie le protocole de manière légitime avec un principal malhonnête I, ce dernier peut utiliser le message pour se faire passer pour A auprès de B (et du serveur). L'attaque d'authentification contre le protocole WMF est illustrée dans le Tableau 3.5. Le protocole n'est pas NAI_ϕ sécuritaire[4] pour la simple raison qu'en supprimant à la fois l'action de déclassification et les actions de haut niveau dans $\gamma_{Auth}(WMF)$, l'action de bas niveau ne pourra pas s'exécuter, alors que supprimer l'action de déclassification et masquer les actions de haut niveau permet à l'action de bas niveau de s'exécuter comme l'illustre l'attaque.

[4] ϕ ne contenant pas bien sûr les clés K_{AS}, K_{BS} et K.

TABLEAU 3.5 *Attaque d'authentification contre le protocole WMF*

(1)	$A \rightarrow I(S)$:	$A, I\{K\}_{K_{AS}}$
(1')	$\mathbf{I(A)} \rightarrow S$:	$A, B\{K\}_{K_{AS}}$
(2)	$\mathbf{S} \rightarrow B$:	$\{A, K\}_{K_{BS}}$
(3)	$A \rightarrow I(B)$:	$\{M\}_K$
(3')	$I(A) \rightarrow B$:	$\{M\}_K$

3.4.2 Intégrité

L'intégrité est une propriété de sécurité qui permet de garantir qu'une information ne soit modifiée que par les utilisateurs habilités, dans les conditions normalement prévues. C'est une propriété des systèmes de sécurité à flux d'information admissible qui permet d'assurer que toute modification apportée à un message reçu par un principal a été faite dans des conditions légales, i.e. aucune altération non admissible de l'information n'a été faite. Nous pouvons l'exprimer grâce à l'interférence admissible. En effet, soit un principal A qui envoie un message M à un principal B, nous voulons que B "accepte" le message M seulement si, effectivement, A l'a envoyé au préalable. La dépendance qu'on veut capturer, ici, est celle qu'il y a entre la réception du message et celui de son envoi. Idéalement, on aimerait vérifier la dépendance entre l'envoi et la réception du même message entre deux principaux, mais en pratique, le message dont on veut assurer l'intégrité peut être une composante des messages envoyés et reçus. En effet, dans le cas d'un système de paiement électronique, le protocole doit assurer l'intégrité des données bancaires du client envoyées à la banque. Or, le message original, contenant ces données bancaires et envoyé par le client, est généralement différent de la demande de payement (contenant elle aussi ces données bancaires) envoyé à la banque par le marchand.

Soit P un protocole, A et B deux principaux du protocole et M un message. Nous disons que P assure l'intégrité du message M entre l'émetteur A et le receveur B

si, pour toute exécution du protocole où B termine correctement le protocole en acceptant M, A a effectivement envoyé le message. Aussi, nous allons définir deux fonctions de décoration γ_d^M qui déclassifient les outputs de l'émetteur A et γ_l^M qui décore la spécification du receveur B avec l'action de bas niveau $\overline{accept}(m)$. Soit M un message, P un processus et $\text{Terminal}_M(P)$ l'ensemble d'états terminaux de P acceptant M. Formellement, nous avons

Définition 3.4.3 : *Soit M un message. Alors, les fonctions de décoration d'intégrité γ_d^M et γ_l^M, respectivement de déclassification et de bas niveau sont définies comme suit :*

$$\forall_{P\in\mathcal{P}roc}\ P\xrightarrow{\alpha}Q \Rightarrow \begin{cases} \gamma_d^M(P)\xrightarrow{\overline{d}(m)}\alpha.\gamma_d^M(Q) \\ \qquad\qquad si\ \alpha\in H\ est\ de\ la\ forme\ \overline{c}(m)\ et\ M\prec m \\ \gamma_d^M(P)\xrightarrow{\alpha}\gamma_d^M(Q) \qquad sinon \end{cases}$$

et

$$\forall_{P\in\mathcal{P}roc}\ P\xrightarrow{\alpha}Q \Rightarrow \begin{cases} \gamma_l^M(P)\xrightarrow{\alpha}\overline{accept}(M).\gamma_l^M(Q) & si\ Q\in \text{Terminal}_M(P) \\ \gamma_l^M(P)\xrightarrow{\alpha}\gamma_d^M(Q) & sinon \end{cases}$$

Maintenant, nous pouvons définir la fonction de décoration $\gamma_{Integ}^{M(A,B)}$ d'intégrité du message M entre l'émetteur A et le receveur B formellement comme suit :

Définition 3.4.4 : *Soit $P\equiv A|B|Q$ un protocole, A et B deux principaux du protocole et M un message. Alors, la fonction de décoration $\gamma_{Integ}^{M(A,B)}$ d'intégrité est définie comme suit :*

$$\gamma_{Integ}^{M(A,B)}(P)=\gamma_d^M(A)|\gamma_l^M(B)|Q.$$

Ainsi, la propriété d'intégrité d'un message peut s'énoncer comme suit :

TABLEAU 3.6 *Spécification de la décoration d'intégrité du protocole WMF*

$\gamma_d^M(A) ::=$	$[(K, K_{AS}) \vdash_{enc} x_0][(A, B, x_0) \vdash_{pair} x_1]\overline{c_1}(x_1).[(M, K) \vdash_{enc} x_2]\overline{d}(x_2).\overline{c_3}(x_2).\mathbf{0}$
$\gamma_l^M(B) ::=$	$B_1 \mid B_2$ où $B_1 ::= c_2(y_0).[(y_0, K_{BS}) \vdash_{dec} y_1][y_1 \vdash_{p_1} y_2][y_1 \vdash_{p_2} y_3]$ $B_2 ::= c_3(y_4).[(y_4, y_3) \vdash_{dec} y_5][y_5 = M]\overline{accept}(M).\mathbf{0}$
$S ::=$	$c_1(z_0).[z_0 \vdash_{p_1} z_1][z_0 \vdash_{p_2} z_2][z_0 \vdash_{p_3} z_3][(z_3, K_{z_1 S}) \vdash_{dec} z_4][(z_1, z_4) \vdash_{pair} z_5]$ $[(z_5, K_{z_2 S}) \vdash_{enc} z_6].\overline{c_2}(z_6).\mathbf{0}$

Proposition 3.4.1 *Soit $P \equiv A|B|Q$ un protocole, A et B deux principaux du protocole et M un message. Alors P assure l'intégrité du message M de A vers B dans un environnement ϕ si et seulement si $P \in NAI_\phi^{\gamma_{Integ}^{M(A,B)}}$.*

À titre d'illustration, vérifions si le protocole WMF assure l'intégrité de tout message M envoyé par A à B. La réponse est non, comme illustré par l'attaque décrite dans le Tableau 3.7. En effet, un principale malhonnête I qui exécute une session légitime (session 1 de l'attaque) peut se servir de la clé légitimement acquise dans cette session pour remplacer tout message M d'une session ultérieure entre A et B par tout autre message M' qu'il connaît. En effet, dans la session 2 de l'attaque, sa base de connaissance contient les messages $\{K_1\}_{K_{AS}}$ et K_1 obtenus grâce à la session légitime 1. Ces messages lui permettent ensuite d'envoyer les messages des étapes (1') et (3'). Les messages légitimes (étapes (1) et (3)) interceptés par l'attaquant sont simplement détruits. Puisque le seul état terminal acceptant pour B est $\mathbf{0}$, nous avons

$$\text{Terminal}_M(B) = \begin{cases} \{\mathbf{0}\} & \text{si } y_5 = M \\ \emptyset & \text{sinon} \end{cases}$$

Nous avons donc la spécification de $\gamma_{Integ}^{M(A,B)}(WMF)$ donnée dans le Tableau 3.6 qui n'est pas NAI_{ϕ_2} sécuritaire.

TABLEAU 3.7 *Attaque d'intégrité par enregistrement reprise contre le protocole WMF*

Session 1		
(1)	$A \rightarrow S$:	$A, I\{K_1\}_{K_{AS}}$
(2)	S $\rightarrow I$:	$\{A, K_1\}_{K_{IS}}$
(3)	$A \rightarrow I$:	$\{M_1\}_{K_1}$
Session 2		
(1)	$A \rightarrow \mathbf{I(S)}$:	$A, B\{K\}_{K_{AS}}$
(1')	$\mathbf{I(A)} \rightarrow \mathbf{S}$:	$A, B\{K_1\}_{K_{AS}}$
(2)	$\mathbf{S} \rightarrow \mathbf{B}$:	$\{A, K_1\}_{K_{BS}}$
(3)	$A \rightarrow I(B)$:	$\{M\}_K$
(3')	$\mathbf{I(A)} \rightarrow B$:	$\{M'\}_{K_1}$

3.4.3 Non-répudiation

La non-répudiation est une propriété de sécurité qui cherche à assurer que :

> *Si un principal A envoie un message m à un principal B, alors A peut prouver que B a bien reçu m (non-répudiation de réception ou NRR) et que B peut prouver que A a effectivement envoyé m (non-répudiation d'origine ou NRO).*

NRR et NRO nécessitent respectivement *l'évidence de réception (EOR)* et *l'évidence d'origine (EOO)* et tous les principaux doivent s'accorder que ces évidences constituent des preuves valides qu'un événement particulier de réception ou de transmission a eu lieu. Un juge doit pouvoir vérifier les évidences EOR et EOO en cas de dispute. EOR (resp. EOO) est un ensemble de messages que A (resp. B) acquièrt au cours de l'exécution du protocole. Par exemple, dans le protocole WMF, B pourrait utiliser

l'ensemble $\{\{A, K\}_{K_{BS}}, \{M\}_K\}$ comme évidence d'origine du message M. En effet, le fait que le serveur ait inclu l'identité de A dans le message $\{A, K\}_{K_{BS}}$ signifie que la clé provient de A (d'après le serveur !) et que le succès du déchiffrage du cryptogramme $\{M\}_K\}$ par K signifie que le message provient de A. Mais, d'après l'attaque (Tableau 3.7), on sait que $\{\{A, K\}_{K_{BS}}, \{M\}_K\}$ ne constitue pas une preuve valide de l'origine du message. Avant de formaliser la propriété d'intégrité, examinons rapidement un vrai protocole dont l'objectif principal est la garantie de la non-répudiation.

Protocole de non-répudiation de Zhou-Gollmann

Pour illustrer ce qui peut être une preuve d'évidence de réception ou d'origine, considérons le protocole de non-répudiation de Zhou-Gollmann décrit dans le Tableau 3.8 où

$X \leftrightarrow TTP : m$ signifie que le principal X obtient le message m de TTP en utilisant un *ftp-get.*

L une étiquetage unique liant tous les messages du protocole.

f_m est un "flag" qui indique le rôle du message m.

$[m]_{S_X}$ est la signature du principal X sur le message m (n'inclut pas m).

$EOO\_C = [f_{EOO}, B, L, \{M\}_K]_{S_A}, \{M\}_K$.

$EOR\_C = [f_{EOR}, A, L, \{M\}_K]_{S_B}, \{M\}_K$.

$sub\_K = [f_{SUB}, B, L, K]_{S_A}, K$.

$con\_K = [f_{CON}, A, B, L, K]_{S_{TTP}}, K$.

Le but du protocole est de transmettre un message d'un principal A à un autre principal B, de fournir à B la preuve que le message provient de A, et inversement, de fournir à A la preuve que B a reçu le message. Un tiers de confiance TTP en ligne est impliqué dans le protocole. L'idée est de diviser l'envoi du message M de A vers B en deux étapes. D'abord, A envoi un cryptogramme $\{M\}_K$ du message et B lui

TABLEAU 3.8 *Protocole de non-répudiation de Zhou-Gollmann*

(1)	A $\rightarrow$ B	:	$f_{EOO}, B, L, \{M\}_K, EOO\_C$
(2)	B $\rightarrow$ A	:	f_{EOR}, A, L, EOR_C
(3)	A $\rightarrow$ TTP	:	f_{SUB}, B, L, K, sub_K
(4)	B $\leftrightarrow$ TTP	:	$f_{CON}, A, B, L, K, con\_K$
(5)	A $\leftrightarrow$ TTP	:	$f_{CON}, A, B, L, K, con\_K$

renvoie un accusé de réception. Ensuite, A met la clé K à la disposition de B par l'intermédiaire d'un tiers de confiance. Le principe est le même que celui du protocole WMF avec la différence notable qu'ici chaque message est accompagné d'une preuve de son authenticité $(EOO\_C, EOR\_C, con\_K, sub\_K)$. Ainsi,

$$EOO = \{EOO\_C, con\_K\} \text{ et } EOR = \{EOR\_C, con\_K\}.$$

En cas de dispute, i.e. lorsque B prétend avoir reçu un message m de A et que ce dernier nie l'avoir envoyé ou A prétend avoir envoyé m à B alors que ce dernier nie l'avoir reçu, EOO et EOR doivent être envoyés à un juge pour vérifier que :

- con_K est la signature de TTP sur (f_{CON}, A, B, L, K), signifiant que le message $f_{CON}, A, B, L, K, con\_K$ est rendu disponible par le tiers de confiance TTP à cause de la réception du message f_{SUB} de A,
- EOO_C est la signature de A sur $(f_{EOO}, B, L, \{M\}_K)$, (EOR_C est la signature de B sur $((f_{EOR}, A, L, \{M\}_K)$, respectivement),
- le message m est égale à $\{M\}_K$.

Non-répudiation d'origine (NRO)

Maintenant, nous allons formaliser la garantie de la non-répudiation d'origine d'un message comme une propriété de correspondance, donc une propriété d'interférence

admissible. La formalisation de la garantie de la non-répudiation de réception en découle.

Soit m un message, P un protocole, A et B deux participants au protocole tels que A est la source du message et B sa destination. P garantit la non-répudiation d'origine de m, dénotée $NRO(m)$, si B peut prouver à un juge, hors de tout doute que A est la source de m. Concrètement, ça veut dire que si $EOO(m) = \{m_1, m_2, \cdots, m_k\}$ constitue une évidence d'origine irréfutable de m pour B, alors, à chaque fois que le système est dans un état où B possède tous les messages $m_1, m_2, \cdots$, et m_k, A a préalablement envoyé le message m à B. Mais, comme l'illustre bien le protocole de Zhou-Gollmann, l'envoi du message peut être divisé en plusieurs parties, i.e. A envoie en k étape les messages $m'_1, m'_2, \cdots$, et m'_k tels que la combinaison de tous ces messages permet de révéler m et que pour tout i $(1 \leq i \leq k)$ m_i est la preuve d'origine de m'_i. Par exemple, pour le protocole de Zhou-Gollmann, on a $EOO(M) = \{EOO\_C, con\_K\}$ et EOO_C est la preuve d'origine du cryptogramme $\{M\}_K$, alors que con_K est la preuve d'origine de la clé K. La dépendance que l'on veut détecter est celle qui existe entre la réception de $EOO(m)$, i.e. l'arrivée du système dans un état où B a reçu tous les messages m_i appelé état acceptant pour $EOO(m)$ et l'envoi au préalable par A des messages m'_i. Soit $\text{Acceptant}_{EOO(m)}(P)$, l'ensemble des états acceptants de l'évidence d'origine du message m dans le protocole P et $\text{Auth}_{EOO} : \mathcal{M} \longrightarrow \mathcal{M}$ la fonction qui, étant donné un message m_i de l'évidence d'origine EOO, retourne le message m'_i dont il est la preuve d'origine. Par exemple, pour le protocole de Zhou-Gollmann, on a :

$$\text{Auth}_{EOO(M)}(EOO\_C) = \{M\}_K \text{ et } \text{Auth}_{EOO(M)}(con\_K) = K.$$

Nous pouvons maintenant définir les fonctions de décoration de déclassification et de bas niveau de la non-répudiation d'origine d'un message.

Définition 3.4.5 : *Soit M un message. Alors, les fonctions de décoration d'intégrité $\gamma_d^{NRO(M)}$ et $\gamma_l^{NRO(M)}$, respectivement de déclassification et de bas niveau, sont définies comme suit :*

$$\forall_{P\in\mathcal{P}roc}\ P\xrightarrow{\alpha}Q \Rightarrow \begin{cases} \gamma_d^{NRO(M)}(P)\xrightarrow{\overline{d}(m)}\alpha.\gamma_d^{NRO(M)}(Q) \\ \qquad si\ \alpha\in H\ est\ de\ la\ forme\ \overline{c}(m)\ et\ \exists_{m'\in EOO(M)}\ m'\prec m \\ \gamma_d^{NRO(M)}(P)\xrightarrow{\alpha}\gamma_d^{NRO(M)}(Q) \qquad sinon \end{cases}$$

et

$$\forall_{P\in\mathcal{P}roc}\ P\xrightarrow{\alpha}Q \Rightarrow \begin{cases} \gamma_l^{NRO(M)}(P)\xrightarrow{\alpha}\overline{evidence_{EOO}}(M).\gamma_l^{NRO(M)}(Q) \\ \qquad si\ Q\in \text{Acceptant}_{EOO(M)}(P) \\ \gamma_l^{NRO(M)}(P)\xrightarrow{\alpha}\gamma_d^{NRO(M)}(Q) \qquad sinon \end{cases}$$

Maintenant, nous pouvons définir la fonction de décoration $\gamma_{NRO}^{M(A,B)}$ de non-répudiation d'origine d'un message M d'une source A vers un receveur B. Formellement :

Définition 3.4.6 : *Soit $P\equiv A|B|Q$ un protocole, A et B deux principaux du protocole et M un message. Alors, la fonction de décoration $\gamma_{NRO}^{M(A,B)}$, de non-répudiation d'origine, est définie comme suit :*

$$\gamma_{NRO}^{M(A,B)}(P)=\gamma_d^{NRO(M)}(A)|\gamma_l^{NRO(M)}(B)|Q.$$

La propriété de non-répudiation d'origine d'un message peut s'énonce alors :

Proposition 3.4.2 *Soit $P\equiv A|B|Q$ un protocole, A et B deux principaux du protocole et M un message. Alors, P assure la non-répudiation d'origine du message M de A vers B dans un environnement ϕ si et seulement si $P\in NAI_\phi^{\gamma_{NRO}^{M(A,B)}}$.*

Non-répudiation de réception (NRR)

La non-répudiation de réception (NRR) est le problème dual de la non-répudiation d'origine. Elle a pour but de garantir que A peut prouver à un juge, hors de tout doute, que B a bien reçu le message. La formulation de la non-répudiation de réception en termes de l'interférence admissible peut être obtenue à partir de celle de la non-répudiation d'origine comme suit. Soit $\text{Acceptant}_{EOR(M)}(P)$, l'ensemble des états acceptants de l'évidence d'origine du message m dans le processus P, et $\text{Auth}_{EOR(M)} : \mathcal{M} \longrightarrow \mathcal{M}$ la fonction qui, étant donné un message de l'évidence de réception EOR, retourne le message dont il est la preuve de réception. Par exemple, pour le protocole de Zhou-Gollmann, pour lequel $EOR(M) = \{EOR\_C, con\_K\}$, on a :

$$\text{Auth}_{EOR(M)}(EOR\_C) = \{M\}_K \text{ et } \text{Auth}_{EOR(M)}(con\_K) = K.$$

Nous obtenons, donc, les fonctions de décoration de déclassification et de bas niveau suivantes :

Définition 3.4.7 : *Soit M un message. Alors, les fonctions de décoration d'intégrité $\gamma_d^{NRR(M)}$ et $\gamma_l^{NRR(M)}$, respectivement de déclassification et de bas niveau, sont définies comme suit :*

$$\forall_{P\in\mathcal{P}roc}\ P\xrightarrow{\alpha}Q \Rightarrow \begin{cases} \gamma_d^{NRR(M)}(P)\xrightarrow{\bar{d}(m)}\alpha.\gamma_d^{NRR(M)}(Q) \\ \qquad si\ \alpha\in H\ est\ de\ la\ forme\ c(m)\ et\ \exists_{m'\in EOR(M)}\ m'\prec m \\ \gamma_d^{NRR(M)}(P)\xrightarrow{\alpha}\gamma_d^{NRR(M)}(Q) \qquad sinon \end{cases}$$

et

$$\forall_{P\in\mathcal{P}roc}\ P\xrightarrow{\alpha}Q \Rightarrow \begin{cases} \gamma_l^{NRR(M)}(P)\xrightarrow{\alpha}\overline{evidence_{EOR}}(M).\gamma_l^{NRR(M)}(Q) \\ \qquad si\ Q\in \text{Acceptant}_{EOR(M)}(P) \\ \gamma_l^{NRR(M)}(P)\xrightarrow{\alpha}\gamma_d^{NRR(M)}(Q) \qquad sinon \end{cases}$$

Définition 3.4.8 : *Soit $P \equiv A|B|Q$ un protocole, A et B deux principaux du protocole et M un message. Alors, la fonction de décoration $\gamma_{NRO}^{M(A,B)}$ de non-répudiation de réception*[5] *est définie comme suit :*

$$\gamma_{NRR}^{M(A,B)}(P) = \gamma_l^{NRR(M)}(A)|\gamma_d^{NRR(M)}(B)|Q.$$

Ainsi, la non-répudiation de réception d'un message peut s'énonce alors :

Proposition 3.4.3 *Soit $P \equiv A|B|Q$ un protocole, A et B deux principaux du protocole et M un message. Alors, P assure la non-répudiation de réception du message M de A vers B dans un environnement ϕ si et seulement si $P \in NAI_\phi^{\gamma_{NRR}^{M(A.B)}}$.*

Nous avons prouvé (cf. (Bastien *et al.*, 2006)) le résultat suivant pour le protocole de Zhou-Gollmann :

Théorème 3.4.1 *Le protocole de Zhou-Gollmann, tel que décrit dans le Tableau 3.8, assure la non-répudiation d'origine et de réception du message M du principal A vers le principal B dans tout environnement ϕ ne connaissant pas les clés privées de signature des participants au protocole*[6]*, i.e.*

$$P \in NAI_\phi^{\gamma_{NRO}^{M(A.B)}} \text{ et } P \in NAI_\phi^{\gamma_{NRR}^{M(A.B)}}.$$

A noter que la décoration de la spécification d'un protocole P par les actions de bas niveau et de déclassification se fait de manière automatique et nécessite très peu

[5] *A noter que, contrairement à la non-répudiation d'origine (Définition 3.4.6), ici, on déclassifie les actions de réception de B et on décore les états acceptants de $EOR(M)$ de A.*

[6] *Evidemment nous supposons que le tiers de confiance* TTP *ne permet pas la réutilisation ni de la clé de session K, ni l'étiquette de la session L, ce qui suppose qu'il garde toutes les sessions antérieures dans une base de données et vérifie que la nouvelle clé et l'étiquette de la session courante n'y figurent pas.*

l'intervention de l'usager. En effet, la vérification se fait sur le système de transitions du protocole décoré. Or ce dernier s'obtient automatiquement de celui de P comme l'indique les définitions de fonctions de décoration (cf. Définitions 3.3.5, 3.4.3, 3.4.5 et 3.4.7). Par exemple, dans le cas de la confidentialité, il suffit à l'usager de fixer l'ensemble de messages secrets $\mathcal{S}$ et le reste se fait automatiquement. Dans le cas de l'intégrité d'un message M, il suffit de fixer qui est le principal source A du message pour la déclassification et le principal destinataire B et les états terminaux $\text{Terminal}_M(B)$ du message, i.e. les états dans les quels B à effectivement reçu le message M.

Pour clore ce chapitre sur la vérification des protocoles cryptographiques, nous présentons dans la section qui suit une importante étude de cas d'un protocole de commerce électronique : le protocole ASET. C'est un nouveau protocole de transaction électronique sécuritaire et, surtout, anonyme par l'association d'un logiciel de cryptage et d'une carte à puce. C'est un protocole initialement développé dans le cadre de nos études de maîtrise (Hamadou, 2002) et amélioré grâce à nos techniques de vérification présentés dans ce chapitre. La sécurité du protocole est assurée par l'utilisation des techniques cryptographiques telles que le chiffrement, la signature électronique et l'authentification. L'anonymat est exprimé comme une propriété de flux d'information. Les principaux résultats présentés ici ont été publiés dans (Brlek *et al.*, 2003) -qui fût primé meilleur article de la conférence- avant d'être étendus dans (Brlek *et al.*, 2005).

3.5 Étude de cas : ASET, Anonymous and Secure Electronic Transaction

Le protocole SSL (Garfinkel et Spafford, 2001) de Netscape est actuellement le système de paiement électronique le plus utilisé. Il consiste en la transmission des coordonnées bancaires du client, de manière sécuritaire, au marchand. Le risque de fraude par le marchand qui détient ces coordonnées bancaires est énorme, sans compter que le

site du marchand peut être un écran virtuel servant à pousser les gens à révéler leurs coordonnées bancaires. C'est pourquoi Visa et Mastercard, en association avec d'autres firmes, ont développé en 1997, le protocole SET, un système de paiement spécialement conçu pour le commerce électronique. Ce système, sensé devenir un standard au niveau planétaire, est une solution purement logicielle. Mais de l'avis général :

> "*La protection par un système purement logiciel ne pourra jamais être totalement fiable. La solution la plus viable à l'heure actuelle semble être l'association d'un logiciel de cryptographie et d'une carte à puce.*" [7]

ASET est un protocole qui, par association des techniques cryptographiques et d'une carte à puce, permet de faire des achats sécuritaires et surtout anonymes sur Internet. Par anonymat, nous entendons la non révélation de l'identité du client au marchand.

3.5.1 Aperçu du protocole ASET

Le protocole ASET est un protocole de paiement électronique dont le but est d'assurer la *confidentialité* des données relatives aux détails de paiement du client, *l'intégrité* et *l'authenticité* des messages, la *non-répudiation* et *l'anonymat* du client auprès du marchand. Pour atteindre ces objectifs, ASET utilise les primitives cryptographiques comme le chiffrement (symétrique et à clé publique) et la signature électronique. En vue d'assurer la "fraîcheur" des messages, l'utilisation des nonces de session a été préférée à l'horodatage qui nécessite la synchronisation des différentes horloges, ce qui n'est pas évident dans un système réparti.

[7] Pierre Bresse et Al, dans (Bresse *et al.*, 1997), p. 37

Déroulement d'une transaction

Le déroulement d'une transaction nécessite l'intervention de quatre agents : le *client*, le *marchand* et leurs *banques*. Nous supposons que le client dispose d'un compte bancaire auprès d'une institution financière qui lui délivre une carte à puce accessible par un mot de passe ou NIP et contenant : un logiciel de signature électronique, une clé privée de signature, une clé privée de chiffrage et les clés publiques de chiffrage et de vérification de signature de sa banque. Le marchand possède une clé privée de signature, une clé privée de chiffrage, un certificat de ses clés publiques de chiffrage et de vérification de sa signature signé par sa banque et les clés publiques de toutes les banques avec lesquelles il fait directement affaire. Les banques possèdent leurs paires de clés de chiffrement et de signature ainsi que les certificats de ces clés publiques. La transaction se déroule en cinq phases :

Initialisation : lorsqu'un client décide de faire un achat, le marchand lui envoie son certificat des clés publiques $cert_M$.

Authentification du marchand : en vue de s'assurer de la validité du certificat, le client ouvre une session (login) auprès de son institution financière. Il s'authentifie, donc, auprès de celle-ci. Il transmet le certificat du marchand à sa banque qui procède à la vérification et lui confirme sa validité.

Ordre d'achat : ayant authentifié le marchand, le client procède à l'achat en envoyant au marchand les données relatives au paiement de telle sorte que ce dernier ne puisse ni l'identifier, ni avoir accès à ses coordonnées bancaires, tout en ayant la garantie d'être payé si la banque valide la transaction. Aux données du paiement, est joint le certificat de la banque du client.

Authentification du certificat de la banque du client : en vue de s'assurer que le client lui a transmis un certificat valide, le marchand s'adresse à sa propre banque pour l'authentifier.

Demande de paiement : maintenant que le marchand est sûr de disposer des clés publiques de la banque du client, il prépare une demande de paiement incluant les détails de paiement envoyés par le client et l'envoie à la banque. Cette dernière vérifie si les deux messages concordent et valide la transaction.

Délivrance d'un reçu : lorsque le marchand a reçu la confirmation de la banque, il transmet un reçu au client accompagné d'une confirmation de la banque qui lui permet de vérifier la décision de la banque. La description complète du protocole à la "*Alice and Bob*" est donnée à la section 3.5.2.

Notations

Dans la description du protocole ci-dessous, les lettres A, B, C et M désignent respectivement la banque du marchand, la banque du client, le client et le marchand. K_{xy} désigne une clé de session (secrète) partagée par les principaux X et Y et K_x la clé publique de X. $\{m\}_K$ représente le cryptogramme obtenu en chiffrant le message m avec la clé K. La signature d'un message y par le principal X sera notée par $[y]_x$, son identifiant par id_x et le certificat de ses clés publiques de chiffrage et de vérification de sa signature obtenu auprès de son institution financière par $cert_x$. Les paramètres n_x et ref_x désignent respectivement un nonce de session et un numéro de référence pour la transaction choisie par le principal X. Enfin, nous désignerons par $listP$ la liste des produits achetés, *receipt* le reçu délivré au client par le marchand, $ le montant de la transaction et par *statut* le statut de la transaction (refusée ou acceptée).

3.5.2 Le protocole ASET

La description du protocole à la *"Alice et Bob"* est la suivante :

Initialisation

(1) $\mathrm{M} \rightarrow \mathrm{C} : \quad cert_m$

Authentification du marchand

(2) $\mathrm{C} \rightarrow \mathrm{B} : \quad \{(x, [x]_c)\}_{K_b}$

où $\quad x = (n_c, id_c, cert_m)$

(3) $\mathrm{C} \leftarrow \mathrm{B} : \quad \{(n_b, ref_b, , [x]_b)\}_{K_c}$

où $\quad x = (n_c, n_b, id_c, cert_m, ref_b)$

Ordre d'achat

(4) $\mathrm{C} \rightarrow \mathrm{M} : \quad (listP, \{(n_c, K_{cm})\}_{K_m})$

(5) $\mathrm{C} \leftarrow \mathrm{M} : \quad \{(n_m, ref_m, \$, [x]_m)\}_{K_{cm}}$

où $\quad x = (n_c, n_m, id_m, ref_m, listP, \$)$

(6) $\mathrm{C} \dashrightarrow \mathrm{M} : \quad (ref_m, \{(n_c, n_m, cert_b, ref_m, ref_b, \$, reqC2B)\}_{K_{cm}})$

où $\quad reqC2B = [(n_c, n_m, n_b, id_m, id_c, ref_m, ref_b, \$)]_c$

Authentification du certificat de la banque du client

(7) $\mathrm{M} \dashrightarrow \mathrm{A} : \quad \{(x, [x]_m)\}_{K_a}$

(8) $\mathrm{M} \leftarrow \mathrm{A} : \quad [x]_a$

où $\quad x = (n'_m, id_m, cert_b)$

Demande de paiement

(9) $\mathrm{M} \dashrightarrow \mathrm{B} : \quad \{(x, [x]_m)\}_{K_b}$ où $\quad x = (n_c, n_m, id_m, ref_m, ref_b, \$, reqC2B)$

(10) $\mathrm{M} \leftarrow \mathrm{B} : \quad \{(ref_m, status, reqB2C, [x]_b)\}_{K_m}$

où $\quad x = (n_c, n_m, id_m, id_b, ref_m, ref_b, \$, status, reqB2C)$

$ReqB2C = [(n_c, n_m, n_b, id_m, id_c, ref_m, ref_b, \$, status)]_b$

Délivrance d'un reçu

(11) $\mathrm{C} \leftarrow \mathrm{M} : \quad \{(receipt, status, reqB2C, [x]_m)\}_{K_{cm}}$

où $\quad x = (n_c, n_m, ref_m, ref_b, receipt, status, reqB2C)$

3.5.3 Modélisation du protocole ASET en CSPAD

Nous spécifions chaque principal X par un processus (un agent clos) qui modélise son comportement aux différentes étapes du protocole et sera noté par la même lettre.

Nous désignons par KS_x (resp. KS_x^{-1}) la clé privée (resp. publique) de signature du principal X et par $\mathcal{C} = \{c_i/1 \leq i \leq 11\}$ l'ensemble des canaux publics. Le canal c_i correspond à l'étape (i) du protocole. En plus des fonctions privées habituelles $\mathcal{F}_{id}$ de chacun des principaux, nous supposerons que les deux banques ont chacune une fonction privée $checkCert_X$ (X = A ou B) qui leur permet de vérifier la validité d'un certificat. Nous supposerons également que chaque processus utilise des variables différentes (dont la portée se limite au processus en question). La modélisation du client est illustrée par la figure 3.9. Chaque processus C^i correspond à son comportement à l'intérieur du protocole à l'étape où il se trouve. Par exemple, le processus C^1 signifie qu'après la réception d'un message x_1 par le canal public c_1, le client doit préparer un message x_2 composé de son nonce de session n_c, son identificateur id_c et du message x_1 $(= cert_M)$. Ensuite, il signe le message, le chiffre avec la clé publique de sa banque, l'envoie par le canal public c_2 et se comporte comme le processus C^2. La spécification du protocole complet est donnée à l'Annexe I. Finalement, le protocole se modélise simplement par la synchronisation de ces différents processus. Ainsi, le protocole ASET sera spécifié par

$$ASET := M|C|B|A$$

Notons que les nonces n_c, n_m et la clé de session K_{cm} peuvent être compromis par un intrus (marchand). L'attaque consiste à intercepter le certificat du marchand au message (1) et à le remplacer par le certificat (valide) de l'intrus qui doit être un marchand connu des banques. L'intrus jouera ainsi le rôle d'un marchand honnête auprès du client et cherchera à tromper le marchand. Or, la banque du client ayant authentifié le certificat de l'intrus, inclura l'identifiant de ce dernier dans le message (10). L'intrus pourra essayer de remplacer son id par celui du marchand, chiffrer le message avec la clé publique du marchand et le lui transmettre. Mais, la vérification de la signature de la banque échouera. Il sera donc obligé de livrer la marchandise au client qui le connaît bien, puisque la banque l'ayant authentifié, il ne pourra faire

TABLEAU 3.9 Spécifications des processus client et sa banque en CSPAD

$$
\begin{array}{rl}
C ::= & c_1(x_1).C^1 \\
C^1 ::= & [(n_c, id_c, x_1) \vdash_{pair} x_2][(x_2, KS_c) \vdash_{enc} x_3] \\
& [((x_2, x_3), K_b) \vdash_{enc} x_4]\overline{c_2}(x_4).C^2 \\
C^2 ::= & c_3(x_5).[(x_5, {K_c}^{-1}) \vdash_{dec} x_6] \prod_{1 \leq i \leq 3} [x_6 \vdash_{p_i} x_{6+i}] \\
& [(n_c, x_7, id_c, x_1, x_8) \vdash_{pair} x_{10}][x_{10} \vdash_{hash} x_{11}] \\
& [(x_9, {KS_b}^{-1}) \vdash_{dec} x_{12}][x_{11} = x_{12}]C^3 \\
C^3 ::= & [((n_c, K_{cm}), K_m) \vdash_{enc} x_{13}][(listP, x_{13}) \vdash_{pair} x_{14}]\overline{c_4}(x_{14}).C^4 \\
C^4 ::= & c_5(x_{15})[(x_{15}, K_{cm}) \vdash_{dec} x_{16}] \prod_{1 \leq i \leq 4} [x_{16} \vdash_{p_i} x_{16+i}] \\
& [(n_c, x_{17}, id_m, x_{18}, listP, x_{19}) \vdash_{pair} x_{21}][x_{21} \vdash_{hash} x_{22}] \\
& [(x_{20}, {KS_m}^{-1}) \vdash_{dec} x_{23}][x_{22} = x_{23}]C^5 \\
C^5 ::= & [(n_c, x_{17}, x_7, id_m, id_c, x_{18}, x_8, x_{19}) \vdash_{pair} x_{24}][(x_{24}, KS_C) \vdash_{pair} x_{25}] \\
& [(n_c, x_{17}, cert_b, x_{18}, x_8, x_{19}, x_{25}) \vdash_{pair} x_{26}][(x_{26}, K_{cm}) \vdash_{enc} x_{27}] \\
& [(x_{18}, x_{27}) \vdash_{pair} x_{28}]\overline{c_6}(x_{28}).C^6 \\
C^6 ::= & c_{11}(x_{29})[(x_{29}, K_{cm}) \vdash_{dec} x_{30}] \prod_{1 \leq i \leq 4} [x_{28} \vdash_{p_i} x_{30+i}] \\
& [(n_c, x_{17}, x_{18}, x_8, x_{31}, x_{32}, x_{33}) \vdash_{pair} x_{35}][x_{35} \vdash_{hash} x_{36}] \\
& [(x_{34}, {KS_m}^{-1}) \vdash_{dec} x_{37}][x_{36} = x_{37}] \\
& [(n_c, x_{17}, x_7, id_m, id_c, x_{18}, x_8, x_{19}, x_{31}) \vdash_{pair} x_{38}][x_{38} \vdash_{hash} x_{39}] \\
& [(x_{33}, {KS_b}^{-1}) \vdash_{dec} x_{40}][x_{39} = x_{40}]\mathbf{0}
\end{array}
$$

croire ni au marchand ni au client que la transaction a été refusée. C'est quand même un cas de bris de confidentialité. Illustrons brièvement comment la propriété de $SNAI$ peut capturer ce bris de confidentialité. Soit

$$\phi_0 = \{KS_e, {K_e}^{-1}, id_E, n_e, K_{em}, cert_e, cert_b, K_b, {KS_b}^{-1}, K_a, {KS_a}^{-1}\}$$

l'ensemble de la connaissance initiale de l'intrus et

$$\mathcal{S} = \{id_c, n_c, n_m, n_b, K_{cm}, KS_m, KS_c, KS_b, KS_a, {K_m}^{-1}, {K_c}^{-1}, {K_b}^{-1}, {K_a}^{-1}\}$$

l'ensemble des messages confidentiels. Considérons le sous-processus

$$Q = \gamma^{\mathcal{S}}_{\phi_0 \cup \{cert_m\}}(M^1|C^3|B^2|A)$$

de $\gamma^{\mathcal{S}}_{\phi_0}(ASET)$ où M^1, C^3 et B^2 sont les processus définis dans les Tableaux I.1 et I.2 de l'Annexe I. Q est l'état atteint après que la banque ait authentifié le certificat de l'intrus et que le client est prêt à lui envoyer la clé de session K_{cm}. Puisque le marchand authentifié par la banque du client est l'intrus, ce dernier utilisera sa clé publique K_e pour chiffrer le message $\{(n_c, K_{cm})\}$, i.e $x_{14} = (listP, \{(n_c, K_{cm})\}_{K_e})$, qui est dans l'idéal $I_{\phi_0 \cup \{cert_m\}}(\mathcal{S})$. L'action de haut niveau $\overline{c_4}(x_{14})$ n'est donc pas déclassifiable. Ainsi, d'une part, en masquant les actions de haut niveau nous avons

$$(Q \backslash D)/H \equiv ((\gamma^{\mathcal{S}}_{\phi_0 \cup \{cert_m\}}(M^1|\tau.C^4|B^2|A)) \backslash D)/H$$

et, en les supprimant,

$$Q \backslash (D \cup H) \equiv (\gamma^{\mathcal{S}}_{\phi_0 \cup \{cert_m\}}(M^1|\mathbf{0}|B^2|A) \backslash (D \cup H).$$

Ces deux processus ne sont évidemment pas trace-équivalents puisque le premier permet à l'action de haut niveau $\overline{c_4}((listP, \{(n_c, K_{cm})\}_{K_e}))$ (masquée par l'action τ précedant C^4) de s'exécuter, ce qui conduit au bris de confidentialité de n_c et K_{cm}, alors que le second ne le permet pas.

3.5.4 Anonymat

Pour analyser l'anonymat en tant que propriété de sécurité, nous devons le définir d'une manière précise. Mais plutôt qu'une simple propriété, il semble qu'il existe plusieurs types d'anonymat et ses requis dépendent du type d'application : *e-mail, vote électronique, paiement électronique, publication électronique, etc.* Une définition for-

melle doit être applicable à toutes ces différentes situations et quelques efforts ont été faits dans cette direction : les techniques de spécification et d'analyse formelles d'anonymat incluent les approches basées sur les algèbres de processus (Schneider et Sidiropoulos, 1996) et celles qui utilisent la logique modale de connaissance (Syverson et Stubblebine, 1999; Hughes et Shmatikov, 2004) pour les protocoles non déterministes, et (Shmatikov, 2004b; Brlek *et al.*, 2007b; Brlek *et al.*, 2007a; Chatzikokolakis et Palamidessi, 2007; Garcia *et al.*, 2007) pour les systèmes probabilistes. Dans ce qui suit, nous ne définissons pas un nouveau cadre formel pour spécifier l'anonymat qui sera applicable à tous les différents types d'anonymat. Nous allons prouver que, pour le protocole ASET, l'anonymat peut se réduire à une question de confidentialité et, donc, pourra être spécifié et analysé formellement en appliquant le Théorème 3.3.1. Pour le protocole ASET, définissons précisément ce que nous entendons par *anonymat.*

Définition 3.5.1 : *Le protocole ASET assure* `l'anonymat du client envers le marchand` *si les points suivants sont établis :*

(i) *il est impossible pour le marchand de retracer l'identité du client au moyen du protocole (*intraçabilité*);*

(ii) *le marchand ne peut pas relier différentes transactions performées par le même client, même si son identité reste cachée (*non-liabilité*);*

(iii) *la divulgation de l'identité pour une transaction ne doit pas entraîner la perte d'anonymat pour les autres transactions effectuées par le même client (*non-liabilité parfait*).*

Conformément à la définition ci-dessus, nous avons la proposition suivante :

Proposition 3.5.1 *Le protocole ASET assure* `l'anonymat du client envers le marchand` *si et seulement s'il assure la* `confidentialité de l'identifiant du client` id_c `(avec le nonce de la banque` n_b`) envers le marchand.`

Preuve: 1. Anonymat $\Rightarrow$ confidentialité de l'id_c est triviale par le point (i).

2. Supposons que la confidentialité de l'id_c et n_b est assurée. Nous devons prouver que l'intraçabilité et la non-liabilité sont vérifiées. Nous le montrons par une induction sur la structure des messages :

Message en clair : m contient l'identifiant chiffré (ou le certificat) de X. Pour le client, ceci est impossible par hypothèse.

Message signé : $m = (m', [m']_X)$ est un message signé par X et la vérification de la signature est un succès. Ce cas nécessite donc l'accès à la signature du client. Mais dans ASET, il y a seulement deux messages signés par le client (étapes (2) et (6) du protocole) et ces messages contiennent tous deux son identifiant id_C, donc le succès de la vérification de sa signature dépend de la connaissance préalable de son identifiant, ce qui est impossible par hypothèse.

Message chiffré avec une clé symétrique : $\{m\}_{K_{xy}}$ est un message chiffré avec la clé secrète K_{xy} partagée par X et Y et Y connaissant l'identité de X. Par exemple, supposons qu'on supprime l'identité de l'expéditeur dans le message (1) du protocole WMF. Quand le serveur reçoit le message, il est supposé ignorer sa provenance. Mais, le serveur peut essayer de déchiffrer le message avec toutes le clés partagées avec les autres principaux et, comme seule la bonne clé, i.e. celle de A, pourra déchiffrer le message, il est en mesure de déterminer la source du message même si le message ne contient pas son identité. Ce cas suppose, donc, que la clé partagée soit liée aux identités des principaux qui la partagent ; ce qui n'est pas le cas de la clé secrète générée aléatoirement par le client dans ASET.

Message chiffré avec une clé publique : $\{m\}_{K_x}$, Y envoie un message m *frais* (par exemple un nonce), qu'il a chiffré avec la clé publique de X, et reçoit, en retour, une information que seule la connaissance du message m pourra inférer, signifiant que X a déchiffré son message et participe au protocole. La fraîcheur de m assure que X participe à la présente session du protocole, i.e. la réponse

reçue par Y n'est pas une reprise d'une ancienne session. Or, dans ASET, la seule fois que le client essaie de déchiffrer un message avec sa clé publique est le message (3) envoyé (en principe !) par sa banque. Il ne continuera le protocole que si ce message contient son identifiant (cf. test $[x_i = id_c]$). Là encore, le succès de l'identification dépend de la connaissance préalable de son id, ce qui est impossible par hypothèse.

Donc, la confidentialité de id_C implique l'intraçabilité de son identité. Pour la non-liabilité, notons que les trois items directement liés à l'identité du client sont son identifiant id_c, ses clés publiques et sa signature. La non-liabilité est assurée par le fait que tout message qui contient l'id_c, ou chiffré avec sa clé publique, ou que le client a signé, est bourré avec le nonce frais n_b, qui est différent (et surtout aléatoire !) pour chaque transaction. Si n_b n'est pas confidentiel, alors la divulgation de l'id_c pour une transaction permettra au marchand de lier cette transaction à toutes les transactions performées par le même client, puisque dans ce cas, il connaît tous les paramètres du message signé $ReqB2C$. La vérification de la signature de la banque permet de déterminer si la transaction est effectuée par ce client en particulier.

■

Pour ce protocole, notre objectif est d'assurer que seule la banque du client peut l'identifier. L'anonymat est donc assuré par la confidentialité de son id et le nonce n_B vis-à-vis du marchand, sa banque et tout intrus (le marchand et sa banque sont donc considérés comme complices de l'intrus). Ainsi, l'ensemble des messages confidentiels pour l'anonymat du client se réduit à :

$$\mathcal{S}_{Anonym} = \{id_c, n_b, K_b^{-1}, K_c^{-1}, KS_b, KS_c\},$$

i.e. l'identifiant du client, le nonce de session de sa banque et leurs clés privées. Bien que le marchand et sa banque soient des complices de l'intrus, nous les considérons

comme des agents honnêtes. La raison est que nous modélisons leur complicité avec l'intrus en ajoutant à sa base de connaissance leurs clés privées. Ainsi, si le marchand ou sa banque peuvent identifier le client, alors l'intrus le pourra aussi. L'anonymat (du client), dans le protocole ASET, s'exprime donc en reformulant le Théorème 3.3.1, i.e.

Corollaire 3.5.0 *[Anonymat] Le protocole ASET préserve l'anonymat du client dans un environnement ϕ si et seulement si*

$$ASET \in SNAI_{\phi}^{\gamma_{\phi}^{\mathcal{S}_{Anonym}}}.$$

Nous avons prouvé (voir la preuve complète dans (Brlek *et al.*, 2005)) que l'anonymat est assuré, c'est-à-dire, nous avons le théorème suivant :

Théorème 3.5.1 *Le protocole* ASET *préserve l'anonymat du client envers le marchand dans tout environnement ne connaissant pas initialement les données confidentielles $\mathcal{S}_{Anonym}$, i.e.*

$$\forall_{\phi \subset \mathcal{M}} \ t.q. \ \mathcal{S}_{Anonym} \cup \mathcal{D}(\phi) = \emptyset, \ ASET \in SNAI_{\phi}^{\gamma_{\phi}^{\mathcal{S}_{Anonym}}}.$$

CHAPITRE 4

UN MODÈLE DE CALCUL PROBABILISTE POLYNOMIAL

4.1 Introduction

Le modèle CSPAD, ainsi que la plupart des modèles de spécification de protocoles cryptographiques, est basée sur l'approche dite possibiliste dans laquelle le non-déterminisme sert à modéliser les mécanismes aléatoires de génération de tous les comportements possibles des systèmes de sécurité. Parmi ces théories, celle de flux d'information présentée dans le chapitre 2 semble être très naturelle. En effet, comme nous l'avons vu dans le chapitre précédent la plupart des propriétés de sécurité des crypto-protocoles sont des instances de la non-interférence. Ces modèles (dérivés du modèle de Dolev-Yao) se basent également sur l'hypothèse de cryptographie parfaite et souffrent de deux limitations importantes. En effet, *l'absence des notions de probabilité* limite l'expressivité des ces modèles aux protocoles non probabilistes. Or, de plus en plus de protocoles de sécurité utilisent des procédés aléatoires ("randomization") pour atteindre certains objectifs de sécurité (que l'on pense au protocole probabiliste de signature de contrat électronique Norman et Shmatikov, 2002a ou le "Crowds system" (Reiter et Rubin, 1998) qui garantit l'anonymat pour les transactions sur le web). La deuxième limitation importante est *l'hypothèse de cryptographie parfaite*, qui est une abstraction très forte limitant énormément les capacités de l'attaquant. En effet, la cryptographie parfaite est une idéalisation qui n'existe pas dans nos modèles de calcul classiques (non quantique !), à l'exception du chiffrement à clé jetable de Vernam (le One-Time Pad), qui est pratiquement inutilisable. La sécurité de ces primitives cryptographiques est une sécurité dite *computationnelle* qui stipule que le crypto-système est sécuritaire contre tout attaquant de capacité de calcul en temps

polynomial. De plus, les crypto-systèmes jugés suffisamment sûrs tels que le One-Time Pad, le chiffrement RSA, le chiffrement symétrique en mode ECB (Electronic Code Bloc), pour ne citer que ceux là, présentent des propriétés algébriques (l'associativité, la commutativité et la nilpotence du ou-exclusif de One-Time Pad, l'homomorphisme du mode ECB et du chiffrement à clé public RSA, etc.), qui peuvent devenir des vulnérabilités exploitables par l'attaquant dans un protocole donné (Ryan et Schneider, 1998). En effet, considérons l'exemple suivant où un usager A envoie un message M signé par l'algorithme RSA à un usager B :

$$A \rightarrow B : RSA_{KS_A}(M)$$

où KS_A désigne la clé privée de signature RSA de A. Notons que nous traitons ici la signature avec recouvrement du message, i.e. signer un message revient à le chiffrer avec sa clé privée. En utilisant la propriété algébrique d'homomorphisme de RSA, i.e. $RSA_K(x \times y) = RSA_K(x) \times RSA_K(y)$, nous pouvons monter une attaque qui n'est pas détectable dans un modèle à la Dolev-Yao. En effet, considérons un attaquant qui intercepte deux sessions différentes de ce protocole où les signatures des messages M_1 et M_2 ont été envoyés. Il peut alors forger une nouvelle session pour la signature du message $M_1 \times M_2$ juste en multipliant modulo n (n étant la valeur publique de la clé RSA) les deux sessions interceptées.

4.1.1 Extensions des modèles possibilistes à la de Dolev-Yao

Ces dernières années, plusieurs approches ont été proposées pour étendre les modèles possibilistes à la Dolev-Yao. Nous présentons dans cette section un bref survol de ces approches.

Théorie équationnelle

Des travaux récents ont proposé les moyens de raffiner les modèles possibilistes en affaiblissant l'hypothèse de cryptographie parfaite. Le but est de prendre en compte les propriétés algébriques des primitives cryptographiques. La méthode consiste à enrichir le système déductif de l'intrus par des équations algébriques vérifiées par les crypto-systèmes. Par exemple, la commutativité d'un chiffrement se définit par l'équation $\{\{x\}_y\}_z = \{\{x\}_z\}_y$. Ainsi, si $\phi \vdash \{\{M\}_K\}_{K'}$, alors on a aussi $\phi \vdash \{\{M\}_{K'}\}_K$. Plus généralement, soient ϕ l'ensemble des connaissances de l'attaquant et $\mathcal{E}$ une classe de théories équationnelles vérifiées par les crypto-systèmes. On définit le système déductif $\phi_{\mathcal{E}}$ comme suit : $\phi_{\mathcal{E}} \vdash M$ si $\phi \vdash M$ ou $\phi \vdash M'$ et $\exists E \in \mathcal{E}$ tel que $M =_E M'$. Plusieurs propriétés algébriques ont été étudiées dans la littérature : *la commutativité* (Chevalier *et al.*, 2005; Rusinowitch et Turuani, 2001; Even *et al.*, 1985), *le ou-exclusif* (Chevalier *et al.*, 2005; Comon-Lundh et Cortier, 2003), *les groupes Abéliens* (Millen et Shmatikov, 2003; Shmatikov, 2004a), *l'homomorphisme* (Comon-Lundh et Treinen, 2003), *prefixe et postfixe* (Kremer et Ryan, 2005), *exponentiation modulaire* (Kapur *et al.*, 2003; Millen et Shmatikov, 2003; Boreale et Buscemi, 2003; Chevalier *et al.*, 2003) et des classes de théories équationnelles plus complexes (Abadi et Cortier, 2006; Delaune et Jacquemard, 2004). Un survol complet de ces équations algébriques est donné dans (Cortier *et al.*, 2006). La robustesse de ces modèles conformément aux modèles computationels à base de machines de Turing chers aux cryptographes est maintenant bien établie (Abadi et Rogaway, 2000; Cortier et Warinschi, 2005; Herzog, 2005; Adão *et al.*, 2005; Janvier *et al.*, 2005).

Le principal avantage de ces techniques est que les preuves de sécurité peuvent être totalement automatisées. Le principal inconvénient est qu'elles exigent de restreindre les primitives cryptographiques utilisables, puisque ces dernières doivent souvent satisfaire des propriétés de sécurité très importantes pour correspondre aux contraintes imposées par le système déductif. De même, les protocoles vérifiables doivent égale-

ment satisfaire certaines conditions telles que l'absence des clés cycliques (Adão *et al.*, 2005). De plus, ce sont tout de même des modèles possibilistes qui se sont attaqués uniquement au problème de cryptographie parfaite.

Modèles probabilistes

D'autres modèles sont purement probabilistes : à chaque étape de l'exécution, le choix de la prochaine action à exécuter se fait selon une distribution probabiliste. Les plus connus, abondamment étudiés et utilisés dans la littérature sont les *chaînes de Markov* (Bremaud, 1999) et ses variantes telles que les *chaînes de Markov étiquetées* (Desharnais *et al.*, 2003), les *processus de décision de Markov* (Feinburg *et al.*, 2001; Puterman, 1994) et les *programmes probabilistes* (Kozen, 2003). Des model-checkers tel que *PRiSM* (Kwiatkowska *et al.*, 2001) ont été développés pour ces modèles et sont utilisés pour la vérification des systèmes probabilistes.

Dans le but de pouvoir modéliser des systèmes qui combinent à la fois le non-déterminisme et les probabilités, des travaux plus récents ont exploré des méthodes pour étendre les modèles (possibilistes) à base d'algèbres de processus tels que le π-calcul et le CSP, en y intégrant des probabilités pour pouvoir modéliser les protocoles naturellement probabilistes comme le "Crowds protocol". On distingue deux types des modèles.

D'une part, on a des modèles qui s'attaquent uniquement au manque des notions de probabilité dans les modèles non-déterministes en y ajoutant des notions de probabilité (Chatzikokolakis et Palamidessi, 2005; Aldini *et al.*, 2002; Bengt *et al.*, 2002), mais abstraient la cryptographie. Ces modèles ne s'attaquent, donc, pas au problème de cryptographie parfaite. Le modèle (Aldini *et al.*, 2002) étend CSP au modèle probabiliste des systèmes *génératifs-réactifs* (van Glabbeek *et al.*, 1995) en considérant les actions d'output comme actions génératives et les inputs comme réactives. Les propriétés de sécurité sont exprimées en termes de non-interférence probabiliste qui

exige que la distribution probabiliste des événements de bas niveaux ne soit pas altérée par la modification de celle des événements de haut niveau. Notons que les systèmes de transitions générés par la sémantique de ces processus sont totalement probabilistes, et donc ne nécessitent pas l'utilisation des ordonnanceurs pour résoudre le non-déterminisme. En effet, ceci est évité en associant une probabilité à la composition parallèle mais complique énormément la sémantique probabiliste. Enfin, signalons que ce modèle permet la lecture multiple, ce qui peut limiter dans certains cas la capacité de l'attaquant. Le modèle (Chatzikokolakis et Palamidessi, 2005) étend le π-calcul en associant une probabilité à l'opérateur de choix alternatif. On obtient un modèle ayant une sémantique probabiliste très simple qui génère des automates probabilistes. Ces automates ont deux types de transitions : des transitions probabilistes représentant les choix alternatifs, et des transitions non-déterministes qui représentent les choix purement non-déterministes engendrés par la composition parallèle. Le non-déterminisme est éliminé en utilisant des ordonnanceurs qui permettent de choisir (avec probabilité 1) une seule transition. Les auteurs y ont développé une version probabiliste de la sémantique de test et s'en servent pour exprimer les propriétés de sécurité.

D'autre part, les modèles (Mitchell *et al.*, 2006; Blanchet, 2006; Laud et Vene, 2005) définissent des modèles computationnels qui éliminent complètement l'hypothèse de cryptographie parfaite. Le modèle (Mitchell *et al.*, 2006) -une extension du modèle CCS avec réplication finie- est un modèle probabiliste polynomial et computationnel. Les primitives cryptographiques sont considérées comme des termes (fonctions) probabilistes polynomiaux. Ces primitives font partie de la syntaxe des processus ; ce qui permet d'écrire directement l'algorithme utilisé, qu'il s'agisse de RSA, El-Gammal, DES, AES, etc., dans la syntaxe du processus. L'analyse permet donc de prendre en compte les caractéristiques spécifiques de ces algorithmes, telles que leurs propriétés algébriques. La sémantique probabiliste du modèle est plus complexe que celle définie dans (Chatzikokolakis et Palamidessi, 2005), mais plus simple comparée à celle

proposée dans (Aldini *et al.*, 2002). Ils y ont défini une équivalence observationnelle asymptotique (en fonction du paramètre de sécurité) et s'en servent pour exprimer les propriétés de sécurité. À notre connaissance, ce sont les seuls modèles à base d'algèbres de processus qui se sont réellement attaqués à la fois aux deux principales limites des modèles possibilistes à la Dolev-Yao : *l'absence des notions de probabilité* et *l'hypothèse de cryptographie parfaite.*

4.1.2 Notre contribution

Il est d'usage d'utiliser les ordonnanceurs pour résoudre le non-déterminisme dans des modèles probabilistes. Et, comme le réseau de communication est généralement considéré sous le contrôle de l'attaquant, l'ordonnancement est alors contrôlé par ce dernier. Ainsi, l'ordonnancement doit être soigneusement défini pour refléter autant que possible la capacité de l'intrus à contrôler le réseau de communication, sans pour autant contrôler la réaction interne du système avec lequel il interagit. En effet, considérons le protocole $\overline{c}(a).\mathbf{0}|\overline{c}(b).\mathbf{0}$, qui peut émettre les messages a et b sur le canal public c, et l'intrus $c(x).\mathbf{0}$, qui écoute une seule fois sur ce canal public. Compte tenu de la nature totalement non-déterministe du protocole, la probabilité que le message écouté par l'intrus soit l'un ou l'autre message doit être la même. Un ordonnanceur qui pourrait attribuer une distribution probabiliste quelconque à ces deux actions pourrait également forcer le protocole à émettre l'un ou l'autre message. Un tel ordonnanceur est donc trop puissant. Toutefois, on doit faire très attention en limitant la puissance des ordonnanceurs, autrement cela pourrait résulter en des attaquants trop faibles. En effet, considérons le processus

$$P = (c(x).\overline{c'}(x).\mathbf{0}|\overline{c'}(1).\mathbf{0}|c'(y).[y = 0]\overline{c}(secret).\mathbf{0})\backslash\{c'\},$$

qui est évidemment non sécuritaire, puisque l'intrus pourrait envoyer 0 à P à travers le canal public c et, ainsi, permettre à P de publier le secret. Une telle faille de sécurité ne peut pas être détectée par une sémantique dont l'ordonnancement donne la priorité aux actions internes.

Aussi, dans ce chapitre, nous proposons une extension probabiliste polynomiale à la PPC (Mitchell *et al.*, 2006) de notre modèle CSPAD. Nous l'avons muni d'une sémantique spécialement définie pour résoudre le problème de caractérisation de la puissance des attaquants. C'est un modèle dénommé ProSPA (*Probabilistic Security Proccess Algebra*), qui étend l'algèbre de processus CSPAD syntaxiquement, pour prendre explicitement en compte la spécification des primitives cryptographiques (probabilistes) polynomiales et les protocoles probabilistes et, sémantiquement, pour prendre en compte l'analyse des protocoles de sécurité dans un environnement hostile. Contrairement à la plupart des modèles probabilistes notre sémantique ne normalise pas les probabilités. La raison est que la normalisation a pour effet d'affaiblir l'attaquant en lui enlevant le contrôle total sur ses propres actions. En effet, considérons le processus $P = \overline{c}(m).Q_1|\overline{c}(m).Q_2$. Dépendamment qu'il représente un protocole ou un attaquant, l'ordonnancement de ses deux composantes doit être respectivement équiprobable et quelconque. Une solution pourrait être de discriminer, au niveau sémantique, entre les actions du protocole analysé et celles de l'attaquant. Mais, cette solution est difficilement praticable puisque les actions de synchronisation peuvent impliquer les deux. Notre solution consiste à partitionner l'ensemble des actions exécutables à chaque état en classes d'actions indistinguables par un attaquant et de forcer les ordonnanceurs à choisir les actions de la même classe équiprobablement. Nous avons démontré la correction de notre approche et la limite des approches utilisées dans d'autres modèles, en particulier celle dans (Mitchell *et al.*, 2006).

En vue de vérifier des propriétés de sécurité, nous avons défini deux équivalences asymptotiques. La première équivalence est *l'équivalence observationnelle asympto-*

tique de Mitchell *et al.*, 2006. C'est une équivalence qui stipule que deux processus sont équivalents, si et seulement si, soumis aux mêmes attaques, ils engendrent "approximativement" les mêmes observations, c'est-à-dire, les mêmes actions visibles avec approximativement les mêmes probabilités. Par approximativement, nous voulons dire *asymptotiquement* proches par rapport au paramètre de sécurité. En effet, notre objectif est de vérifier des protocoles de sécurité dont la sécurité des primitives cryptographiques dépend de la valeur du paramètre de sécurité. Comme c'est souvent le cas dans des modèles computationels, un protocole peut ne pas être sécuritaire pour certaines valeurs du paramètre de sécurité, mais, si on augmente suffisamment le paramètre de sécurité (par exemple la taille des clés), alors aucun adversaire de puissance de calcul en temps polynomial ne peut attaquer le protocole. Bien que pratique, l'équivalence observationnelle s'avère non adaptée pour l'analyse de plusieurs systèmes pour lesquels on est souvent amené à calculer la probabilité d'une suite d'observables comme nous allons le voir dans notre étude de cas. Pour remédier à ce problème, nous l'avons étendu pour inclure toutes les traces observables. Nous avons montré que les deux formulations sont équivalentes.

Enfin, pour démontrer l'utilité de notre approche, nous avons analysé un important protocole de sécurité : le protocole *Dîner des Cryptographes* (Chaum, 1988). Nous avons proposé une version probabiliste de la spécification de la propriété d'*anonymat* due à Schneider et Sidiropoulos, 1996, et une spécification du protocole comportant une faille de sécurité et avons montré qu'une sémantique qui privilégie les actions internes ne peut pas détecter cette faille. Nous avons ensuite montré que, dans une implémentation idéale du protocole, l'anonymat du payeur ne dépend que du "fairness" des jetons. Nous avons également montré que, même si les jetons ne sont pas biaisés, une implémentation réelle du protocole peut ne pas être sécuritaire, ceci étant dû à certaines propriétés mathématiques des crypto-systèmes utilisés ; ce qui montre l'importance d'avoir des modèles qui vont au delà des hypothèses traditionnelles à la Dolev-Yao.

4.1.3 Caractérisation de la puissance de l'intrus et modèles probabilistes

Le problème de caractérisation de la puissance de l'attaquant à été récemment abordé dans d'autres types de modèles (Canetti *et al.*, 2006; Chatzikokolakis et Palamidessi, 2007; Garcia *et al.*, 2007). L'article (Canetti *et al.*, 2006) traite le problème d'ordonnanceurs trop puissants dans le modèle *d'Automate Probabiliste.* Ils limitent la puissance des attaquants en définissant deux niveaux d'ordonnanceurs. Un ordonnanceur de haut niveau appelé *ordonnanceur adversaire* contrôle le réseau de communication. Il a une connaissance limitée du comportement des autres composantes du système. En effet, leurs choix internes et informations secrètes lui sont cachés. D'autre part, un ordonnanceur de bas niveau, dénommé *ordonnanceur des tâches*, résout le non-déterminisme interne du système. Ces *tâches* sont des classes d'équivalence d'actions et sont indépendantes des choix de l'ordonnanceur adversaire.

Le même problème a été abordé dans (Garcia *et al.*, 2007) toujours dans un modèle *d'Automate Probabiliste.* Il y est défini un ordonnanceur probabiliste qui attribue localement une distribution probabiliste aux transitions sortantes de chaque état. Contrairement à notre approche, leur ordonnanceur est historique dépendant, puisqu'il définit, en fait, des chemins équiprobables. De plus, il n'est pas stochastique et peut, donc, arrêter l'exécution à tout moment. Plus spécifiquement, les ordonnanceurs admissibles sont définis relativement à une bisimulation : toutes traces, observationnellement équivalentes, sont équiprobablement choisies et mènent à des états bisimilaires.

Une autre approche très récent sur ce problème est proposée dans (Chatzikokolakis et Palamidessi, 2007). Au contraire de notre approche et celle dans (Garcia *et al.*, 2007), qui définissent l'ordonnancement au niveau sémantique, l'article (Chatzikokolakis et Palamidessi, 2007) propose un modèle (une extension probabiliste, mais non computationnelle, du CCS) dans lequel l'ordonnancement est contrôlé au ni-

veau syntaxique. Le but est de rendre les choix probabilistes internes du protocole transparents à l'attaquant. Notons que ce problème ne se pose pas dans un modèle computationnel comme le notre, dans lequel les choix (probabilistes) internes sont traités comme calculs internes, au même titre que l'évaluation des primitives cryptographiques et, donc, ne sont pas ordonnançables. Toutefois, cette approche peut être complémentaire à notre approche et, ainsi, faciliter la définition de nos classes d'actions stratégiquement équiprobables.

4.1.4 Organistion du chapitre

Le reste du chapitre se présente comme suit : la section 4.2 présente en détail notre modèle. Les propriétés de sécurités sont exprimées en termes d'équivalences asymptotiques dans la section 4.3 et notre étude de cas est présentée dans la section 4.4.

4.2 Modèle de calcul ProSPA

L'algèbre de processus ProSPA (*Probabilistic Security Preces Algebra*) étend l'algèbre de processus CSPAD syntaxiquement, pour prendre explicitement en compte la spécification des primitives cryptographiques (probabilistes) polynomiales et les protocoles probabilistes et sémantiquement, pour prendre en compte l'analyse des protocoles de sécurité dans un environnement hostile.

4.2.1 Fonctions probabilistes polynomiales

Les définitions qui suivent sont standard : voir par exemple (Atallah, 1999) (chapitre 24, pp. 19-28).

Définition 4.2.1 : *Une fonction probabiliste F de X à Y est une fonction $X \times Y \rightarrow [0,1]$ qui satisfait :*
- $\forall x \in X : \sum_{y \in Y} F(x,y) \leq 1$
- $\forall x \in X$, *l'ensemble* $\{y | y \in Y, F(x,y) > 0\}$ *est fini.*

Pour $x \in X$ et $y \in Y$, nous dirons que $F(x)$ s'évalue à y avec la probabilité p, noté $\text{Prob}[F(x) = y] = p$, si $F(x,y) = p$.

Définition 4.2.2 : *La composition $F = F_1 \circ F_2 : X \times Z \rightarrow [0,1]$ de deux fonctions probabilistes $F_1 : X \times Y \rightarrow [0,1]$ et $F_2 : Y \times Z \rightarrow [0,1]$ est la fonction probabiliste :*

$$\forall x \in X. \forall z \in Z : F(x,z) = \sum_{y \in Y} F_1(x,y) \cdot F_2(y,z)$$

Définition 4.2.3 : *Une machine de Turing à oracle est une machine de Turing avec une bande extra à oracle et trois états extra q_{query}, q_{yes} et q_{no}. Quand la machine entre dans l'état q_{query}, le contrôle passe à l'état q_{yes} si le contenu de la bande à oracle appartient à l'ensemble de l'oracle. Sinon le contrôle passe à l'état q_{no}.*

Étant donné une machine de Turing à oracle M, nous notons par $M_\sigma(\overrightarrow{a})$ le résultat de l'application de M à $\overrightarrow{a}$ en utilisant l'oracle σ.

Définition 4.2.4 : *Une machine de Turing à oracle s'exécute en temps polynomial s'il existe un polynôme $q(\overrightarrow{x})$ tel que, pour tout oracle σ, $M_\sigma(\overrightarrow{a})$ s'arrête en temps $q(|\overrightarrow{a}|)$, où $\overrightarrow{a} = (a_1, \cdots, a_k)$ et $|\overrightarrow{a}| = |a_1| + \cdots + |a_k|$.*

Soit M, une machine de Turing à oracle dont le temps d'exécution est borné par le polynôme $q(\overrightarrow{a})$. Puisque $M(\overrightarrow{a})$ ne peut que faire appel à un oracle qu'avec au plus $q(\overrightarrow{a})$ bits, nous avons un ensemble fini $\mathcal{Q}$ d'oracles pour lequels M s'exécute en temps borné par $q(\overrightarrow{a})$.

Définition 4.2.5 : *Nous disons qu'une machine de Turing à oracle M est probabiliste polynomiale et notons* $\text{Prob}[M(\overrightarrow{a}) = b] = p$ *la probabilité que M appliquée à $\overrightarrow{a}$ retourne b est p, ssi en choisissant uniformément un oracle σ dans l'ensemble fini $\mathcal{Q}$, la probabilité que $M_\sigma(\overrightarrow{a}) = b$ est p.*

Définition 4.2.6 : *Une fonction probabiliste F est dite polynomiale si elle est calculable par une machine de Turing probabiliste polynomiale, i.e. pour tout input $\overrightarrow{a}$ et tout output b,* $\text{Prob}[F(\overrightarrow{a}) = b] = \text{Prob}[M(\overrightarrow{a}) = b]$.

4.2.2 Syntaxe de ProSPA

Termes. L'ensemble des termes $\mathcal{T}$ de l'algèbre ProSPA est formé d'un ensemble $\mathcal{V}$ de variables, des nombres $\mathbb{N}$, des couples ainsi qu'un terme spécifique noté $\mathbf{N}$ désignant le paramètre de sécurité. Le paramètre de sécurité peut être vu comme la taille des clés des primitives cryptographiques et peut apparaître dans les fonctions. Formellement, nous avons :

$$t \ ::= \ n \ \ (\textit{entier}) \ \mid \ x \ \ (\textit{variable}) \ \mid \ \mathbf{N} \ \ (\textit{param. sécur.}) \ \mid \ (t,t) \ \ (\textit{couple})$$

Pour tout terme t, on dénote par $fv(t)$ l'ensemble des variables dans t. Un *message* est un terme clos (i.e. ne contenant pas de variables). L'ensemble des messages (i.e. termes clos) est dénoté par $\mathcal{M}$.

Fonctions. Afin de pouvoir modéliser l'appel des primitives cryptographiques telles que la génération de clés ou de nonces, le chiffrage, la déchiffrage, la signature, etc, les règles d'inférence de CryptoSPA sont remplacées ici par l'ensemble Λ des fonctions probabilistes polynomiales de la forme $\lambda : \mathcal{M}^k \to \mathcal{M}$ tel que

$$\forall (m_1, \ldots, m_k) \in \mathcal{M}^k, \ \forall m \in \mathcal{M}, \ \forall \lambda \in \Lambda \ \exists p \in [0,1]$$

$$\text{tel que } \text{Prob}[\lambda(m_1, \cdots, m_k) = m] = p.$$

Nous dénotons par $\lambda(m_1, \cdots, m_k) \hookrightarrow x$ l'affectation de la valeur $\lambda(m_1, \cdots, m_k)$ à la variable x et par $\lambda(m_1, \cdots, m_k) \overset{p}{\hookrightarrow} m$ si $\lambda(m_1, \cdots, m_k)$ s'évalue à m avec la probabilité p. D'après la définition 4.2.1, l'ensemble des messages m tel que $\lambda(m_1, \cdots, m_k)$ s'évalue à m avec une probabilité non nulle est fini. Nous dénotons cet ensemble par $Im(\lambda(m_1, \cdots, m_k))$. Formellement, on a

$$Im(\lambda(m_1, \cdots, m_k)) = \{m | \exists p \in]0..1] \ \lambda(m_1, \cdots, m_k) \overset{p}{\hookrightarrow} m\}.$$

Par exemple, le chiffrement RSA, qui prend comme paramètres le message m à chiffrer et la clé de chiffrage formée par la paire (e, n) et retourne le nombre $m^e \ mod \ n$, est la fonction probabiliste polynomiale $\lambda_{RSA}(m, e, n)$ qui retourne c avec la probabilité 1 si $c = m^e \ mod \ n$ et 0 autrement. De même, nous pouvons modéliser l'attaque à force brute d'un cryto-système par le produit des fonctions $[\text{rand}(1^k) \hookrightarrow key][\text{dec}(c, key) \hookrightarrow x]$ où 1^k est la taille de la clé générée aléatoirement par la fonction `rand` et la fonction de déchiffrage `dec` retourne m avec la probabilité 1 si $c = \{m\}_k$ et $key = k$. Le succès d'une telle attaque est de probabilité $p = \frac{1}{2^{|k|}}$. Nous pouvons également modéliser le jet d'une pièce (biaisée ?) qui retourne pile ou face avec les probabilités respectives p et $1 - p$ par la fonction λ d'arité 1 qui retourne 0 ou 1 avec les probabilités respectives p et $1 - p$. Ces quelques exemples illustrent l'expressivité qu'offrent ces fonctions. Nous nous limitons à des fonctions (probabilistes) polynomiales pour pouvoir modéliser toutes les attaques réalisables (dans le modèle) en temps polynomial.

Processus. Soit $\mathcal{C}$ un ensemble dénombrable de canaux publics. Nous supposons que chaque canal est muni d'une bande passante donnée par la fonction $bw : \mathcal{C} \longrightarrow \mathbb{N}$. Nous dirons qu'un message m appartient au domaine d'un canal c, dénoté par $m \in$

$\mathrm{dom}(c)$, si

$$|m| \leq bw(c).$$

avec $|(m, m')| = |m| + |m'| + |r|$ où r est un entier fixé nous permettant de composer et décomposer deux termes de manière non ambigüe.

Les processus ProSPA sont construits comme suit :

$$\begin{aligned} P \ ::= \mathbf{0} \quad & | \quad c(x).P \quad | \quad \overline{c}(m).P \quad | \quad P|P \quad | \quad P\backslash\Gamma \quad | \quad [t = t]P \quad | \\ & | \quad !_{q(\mathbf{N})}P \quad | \quad [\lambda(t_1, \cdots, t_n) \hookrightarrow x]P \end{aligned}$$

Étant donné un processus P, l'ensemble des *variables libres* $fv(P)$ est l'ensemble des variables x de P qui ne sont dans la portée d'aucun préfixe d'entrée (de la forme $c(x)$) ou d'évaluation probabiliste (de la forme $[\lambda(t_1, \cdots, t_n) \hookrightarrow x]$). Nous dirons qu'un processus est clos s'il ne contient aucune variable libre et dénotons par $\mathcal{P}roc$ l'ensemble des processus clos. Dans la suite de ce document, tous les processus considérés sont clos.

Les mécanismes de lecture, d'émission, la composition parallèle, la restriction, la correspondance et la constante sont classiques. Les seules nouveautés par rapport au modèle CSPAD sont les mécanismes d'appel des fonctions probabilistes polynomiales et la réplication finie. Le mécanisme d'appel des fonctions probabilistes polynomiales nous permet de modéliser les primitives cryptographiques ainsi que la nature probabiliste d'un protocole. C'est la principale source de probabilité dans ce modèle, comme nous allons le voir à la section 4.2.3. La réplication finie $!_{q(\mathbf{N})}P$ est la composition parallèle du processus P avec lui-même $q(\mathbf{N})$ fois où q est un polynôme. On remarquera également que nous n'avons pas pris en considération l'opérateur du choix alternatif "+". Nous l'avons omis pour la simple raison que nous pouvons le simuler grâce à l'opérateur de parallélisme et une fonction probabiliste spéciale.

Exemple 1

Soit $flips$, une fonction probabiliste polynomiale qui, étant donné $coin$ une pièce, retourne $Pile$ avec probabilité p et $Face$ avec probabilité $1 - p$. Alors, le processus Q qui jette la pièce et se comporte comme le processus Q' si le résultat est Pile et comme le processus Q'' sinon, est modélisé comme suit :

$$Q := [flips(coin) \hookrightarrow x]([x = Head]Q'|[x = Tail]Q'')$$

Exemple 2

Soit $Kgen$ une fonction (probabiliste) polynomiale qui, étant donné le paramètre de sécurité $\mathbf{N}$, génère une clé secrète de chiffrage et enc une fonction (probabiliste) de chiffrage qui, étant donné une clé secrète K et un message M, retourne le cryptogramme de M par K. Le processus P, qui reçoit un message à travers un canal public c, génère une clé de chiffrage et retourne le cryptogramme du message à travers le même canal public, peut être modélisé dans ProSPA comme suit :

$$P := c(x).[Kgen(\mathbf{N}) \hookrightarrow y][enc(x, y) \hookrightarrow z]\overline{c}(z).\mathbf{0}$$

4.2.3 Sémantique opérationnelle.

L'ensemble des actions

$$\mathcal{A}ct = \{\overline{c}(m), c(m), \overline{c}(m) \cdot c(m), c(m) \cdot \overline{c}(m), \tau \mid m \in \mathcal{M} \text{ et } c \in \mathcal{C}\}$$

est formé par l'ensemble d'actions d'output et d'input partielles, et l'ensemble d'actions de synchronisation sur les canaux publics et l'action interne τ :

$$\begin{aligned} \mathcal{P}artial &= \{\overline{c}(m), c(m) \mid m \in \mathcal{M} \text{ et } c \in \mathcal{C}\} \\ \mathcal{A}ctual &= \{\overline{c}(m) \cdot c(m), c(m) \cdot \overline{c}(m), \tau \mid m \in \mathcal{M} \text{ et } c \in \mathcal{C}\} \end{aligned}$$

L'ensemble des actions observables est $\mathcal{V}is = \mathcal{A}ct - \{\tau\}$. La sémantique opérationnelle de PPC est un système de transitions probabiliste $(\mathcal{E}, \mathscr{T}, E_0)$ engendré par les règles d'inférence de la Table 4.1 où

- $\mathcal{E} \subseteq \mathcal{P}roc$ est l'ensemble des sommets.
- $\mathscr{T} \subseteq \mathcal{E} \times \mathcal{A}ct \times [0,1] \times \mathcal{E}$ est l'ensemble des transitions.
- $E_0 \in \mathcal{P}roc$ est le sommet initial.

Nous dénotons par $P \xrightarrow{\alpha[p]} P'$ la transition de P à P' pour signifier que le processus P peut faire l'action α avec la probabilité p et évolue comme le processus P'. C'est une extension de la sémantique de CCS avec un mécanisme d'appel des fonctions probabilistes polynomiales. Nous l'esquissons brièvement dans ce paragraphe.

L'appel des fonctions cryptographiques et le retour de valeur circulant sur des canaux privés sont modélisés par la règle `Eval`.. Pour que les calculs internes des ces primitives ne puissent pas interférer avec les actions de communication, en particulier celles sur les canaux publics contrôlés par un attaquant hostile, nous évaluons simultanément toutes les fonctions probabilistes exposées grâce à la fonction *eval* (cf. Définition 4.2.7) ci-dessous, comme illustré par l'Exemple 3 qui suit. Cette étape, dénomée *Réduction*, permet d'obtenir des processus bloqués, i.e. des processus n'ayant aucun calcul interne à faire. L'ensemble des processus bloqués est dénoté par $\mathcal{B}locked$.

Définition 4.2.7 : *La fonction d'évalution* $eval : \mathcal{P}roc \longrightarrow \mathcal{B}locked$ *est une fonction qui, étant donné un processus* P*, évalue toutes les fonctions probabilistes polynomiales exposées de* P*, i.e. tous les calculs internes du processus, pour retourner un processus*

TABLEAU 4.1 *Sémantique opérationnelle du modèle* ProSPA

$$\texttt{Eval.}\quad \frac{eval(P) \xrightarrow{p} P' \quad P \notin Blocked}{P \xrightarrow{\tau[p]} P'}$$

$$\texttt{Output}\quad \frac{m \in \mathrm{dom}(c)}{\overline{c}(m).P \xrightarrow{\overline{c}(m)[1]} P} \qquad \texttt{Input}\quad \frac{m \in \mathrm{dom}(c)}{c(x).P \xrightarrow{c(m)[1]} P[m/x]}$$

$$\texttt{ParL.}\quad \frac{P_1 \xrightarrow{\alpha[p]} P_1' \quad (P_1|P_2) \in Blocked}{P_1|P_2 \xrightarrow{\alpha[p]} P_1'|P_2} \qquad \texttt{ParR.}\quad \frac{P_2 \xrightarrow{\alpha[p]} P_2' \quad (P_1|P_2) \in Blocked}{P_1|P_2 \xrightarrow{\alpha[p]} P_1|P_2'}$$

$$\texttt{SyncL.}\quad \frac{P_1 \xrightarrow{\overline{c}(m)[p_1]} P_1' \quad P_2 \xrightarrow{c(m)[p_2]} P_2'}{P_1|P_2 \xrightarrow{\overline{c}(m).c(m)[p_1.p_2]} P_1'|P_2'} \qquad \texttt{SyncR.}\quad \frac{P_1 \xrightarrow{c(m)[p_1]} P_1' \quad P_2 \xrightarrow{\overline{c}(m)[p_2]} P_2'}{P_1|P_2 \xrightarrow{c(m).\overline{c}(m)[p_1.p_2]} P_1'|P_2'}$$

$$\texttt{Rest}\quad \frac{P \xrightarrow{\alpha[p]} P' \quad \alpha \notin \Gamma \quad P \in Blocked}{P\backslash\Gamma \xrightarrow{\alpha[p]} P'\backslash\Gamma}$$

bloqué prêt à exécuter une action sur ses canaux (publics et privés). Elle est définie, récursivement, comme suit :

$\mathrm{Prob}[eval(\mathbf{0}) = \mathbf{0}] = 1$

$\mathrm{Prob}[eval(\overline{c}(m).P) = \overline{c}(m).P] = 1$

$\mathrm{Prob}[eval(c(x).P) = c(x).P] = 1$

$$\begin{cases} \mathrm{Prob}[eval([m = m']P) = Q] = \mathrm{Prob}[eval(P) = Q] & si\ m = m' \\ \mathrm{Prob}[eval([m = m']P) = \mathbf{0}] = 1 & sinon \end{cases}$$

$\mathrm{Prob}[eval(P|Q) = P'|Q'] = \mathrm{Prob}[eval(P) = P'] \times \mathrm{Prob}[eval(Q) = Q']$

$\mathrm{Prob}[eval(P\backslash\Gamma) = Q\backslash\Gamma] = \mathrm{Prob}[eval(P) = Q]$

$\mathrm{Prob}[eval([\lambda(m_1, \cdots, m_k) \hookrightarrow x]P) = Q] =$

$\quad \sum_{m \in Im(\lambda(m_1,\cdots,m_k))} \mathrm{Prob}[\lambda(m_1, \cdots, m_k) = m] \times \mathrm{Prob}[eval(P[m/x]) = Q]$

Exemple 3

Si les processus Q' et Q'' de l'Example 1 (Section 4.2.2) sont des processus bloqués, alors Q se réduit à Q' avec la probabilité p et à Q'' avec la probabilité $1 - p$. En

d'autres termes, Prob[$eval(Q) = Q'$] = p et Prob[$eval(Q) = Q''$] = $1 - p$.

Le mécanisme d'output permet à un principal A d'envoyer un message sur les canaux publics (règle `Output`). Dualement, le mécanisme d'input doit prévoir la réception de n'importe quel message sur un canal public (règle `Input`). L'opérateur de parallélisme et la restriction (règles `Par.` et `Rest.`) sont définis de façon habituelle. Comme dans le cas du modèle possibiliste CSPAD, nous utilisons la notation $P\backslash C$ pour restreindre les communications sur l'ensemble des canaux publics C. Sa sémantique est donnée dans le Tableau 4.2. Nous avons évidemment Prob[$eval(P\backslash C) = Q\backslash C$] = Prob[$eval(P) = Q$].

TABLEAU 4.2 *Sémantique opérationnelle des canaux privés*

$$\texttt{RestCL.}\quad \frac{P \xrightarrow{\overline{c}(m).c(m)[p]} P' \quad c \in C}{P\backslash C \xrightarrow{\tau[p]} P'\backslash C} \qquad \texttt{RestCR.}\quad \frac{P \xrightarrow{c(m).\overline{c}(m)[p]} P' \quad c \in C}{P\backslash C \xrightarrow{\tau[p]} P'\backslash C}$$

$$\texttt{RestC}\quad \frac{P \xrightarrow{\alpha[p]} P' \quad \alpha \notin \{c(m), \overline{c}(m), \overline{c}(m)\cdot c(m), c(m)\cdot\overline{c}(m) \mid m \in \mathcal{M} \text{ et } c \in C\} \quad P \in \mathit{Blocked}}{P\backslash C \xrightarrow{\alpha[p]} P'\backslash C}$$

Enfin on notera que, contrairement à ce qui est d'usage dans d'autres modèles probabilistes, nous n'avons pas normalisé les probabilités. La raison principale est que, les attaquants ou l'environnement hostile dans lequel évolue le protocole étant représentés par un processus, normaliser les probabilités revient à enlever à l'attaquant le contrôle total sur ses propres actions. En effet, considérons le processus P suivant : $P = \overline{c}(m).Q_1 | \overline{c}(m).Q_2$. Si P représente un protocole évoluant dans un environnement hostile, alors la probabilité que le premier message m émis par le protocole sur le canal public c provienne de sa composante gauche ou de sa composante droite est la même, c'est-à-dire $\frac{1}{2}$. Par contre, si P représente l'attaquant, alors il peut choisir comme il l'entend la composante qui doit s'exécuter en premier selon qu'il veuille utiliser Q_1 ou Q_2 pour continuer son attaque. Ainsi, normaliser nous obligerait soit à affaiblir l'attaquant (comme dans les autres modèles), soit à avoir une sémantique qui doit

distinguer les cas où le processus modélise un attaquant ou un protocole, sans compter que les actions de synchronisation peuvent impliquer les deux à la fois. Pour éviter les maux de tête engendrés par une telle sémantique, nous avons opté pour une autre approche beaucoup plus simple. Notre approche consiste à munir l'attaquant d'une stratégie d'attaque (cf. la section qui suit), c'est-à-dire d'un processus de sélection qui lui permettra de décider à chaque étape d'évaluation quelle action exécuter. Ce processus de sélection sera défini pour refléter le pouvoir décisionnel de l'attaquant sur les actions selon qu'elles impliquent ou non une action du processus modélisant l'attaquant.

4.2.4 Modèle de l'Attaquant

Notons que les systèmes de transitions engendrés par la sémantique opérationelle des processus ProSPA ne sont pas purement probabilistes, puisque la somme des probabilités des transitions sortantes d'un sommet peut être supérieure à 1. En effet, considérons les processus $P = \overline{c_1}(a).\mathbf{0}|\overline{c_2}(b).\mathbf{0}$, dont le système de transitions est illustré par la Figure 4.1 où $\alpha = \overline{c_1}(a)$ et $\beta = \overline{c_2}(b)$. On voit clairement que la somme des probabilités des transitions sortantes du sommet initial est égale à 2. C'est essentiellement dû à la composition parallèle qui introduit du non-déterminisme. Pour résoudre ce non-déterminisme et obtenir des systèmes de transitions purement probabilistes, il est donc essentiel d'ordonnancer les différentes actions disponibles à chaque étape d'évaluation du processus. Mais en pratique, lorsqu'on analyse un protocole cryptographique, nous devons supposer que ce dernier s'exécute dans un environnement hostile, c'est-à-dire l'existence d'un attaquant externe qui exerce un contrôle total sur le réseau de communication. Typiquement, cet attaquant est un processus qui essaie d'attaquer le protocole en interagissant avec ce dernier à travers les canaux de communication publics. Il peut donc intercepter et modifier tout message envoyé sur un canal public ou envoyer tout message qu'il connait ou sait créer.

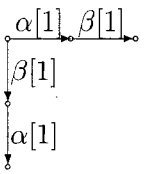

FIGURE 4.1 *Système de transitions probabiliste de* $P = \overline{c_1}(a).\mathbf{0}|\overline{c_2}(b).\mathbf{0}$.

Mais, des tels attaquants peuvent s'avérer trop puissants si on ne leur impose pas certaines restrictions. En effet, les canaux de communication étant supposés être sous le contrôle de l'attaquant, l'ordonnancement des actions visibles doit être contrôlé par ce dernier. Il pourrait donc leur affecter une distribution probabiliste selon ses désirs. Mais, bien qu'ayant un contrôle total sur les canaux de communication, il ne doit être ni capable de modifier les distributions probabilistes des actions internes du processus qui sont indépendantes de la distribution probabiliste des canaux publics, ni de contrôler les réactions internes du protocole à ses requêtes. Puisque nous ne voulons pas définir une stratégie d'attaque en particulier, l'ordonnancement n'est pas inclus dans la sémantique des processus, mais plutôt dans la définition de l'intrus. Ainsi, un intrus est identifié par la paire (Π, S) formée d'un processus Π représentant l'ensemble des actions qu'il peut faire et d'un ordonnanceur, dénoté par S (décrit en détail dans la section 4.2.5 qui suit), représentant sa stratégie d'attaque, i.e. l'ordre dans lequel et les probabilités selon lesquelles ces actions seront exécutées[1].

4.2.5 L'ordonnanceur externe

Dans cette section, nous définissons formellement l'ordonnanceur externe. N'oublions pas que ce dernier est contrôlé par l'attaquant et nous devons faire en sorte que sa définition reflète la capacité réelle de nuisance de l'intrus. En effet, bien que l'attaquant

[1]Par abus de langage, nous identifierons souvent l'attaquant au processus Π.

ait un contrôle total sur les canaux de communication, il ne doit pas être capable de modifier les distributions probabilistes des actions internes du processus qui sont indépendantes de la distribution probabiliste des canaux publics. L'exemple du processus $P = \overline{c}(m).Q_1|\overline{c}(m).Q_2$ qui se comporte différement, selon qu'on le considère comme un intrus ou un système attaqué, nous montre que le processus de sélection des actions, autrement dit l'ordonnencement, doit pouvoir identifier si l'action à choisir implique ou non une action de l'attaquant. Puisqu'un système P attaqué par l'intrus Π est simplement identifié par la composition parallèle $P|\Pi$, nous allons indexer les actions par les positions des composantes qu'elles font intervenir. Par composante, nous voulons dire la composante parallèle. Ainsi, si P et Π ont respectivement n et k composantes parallèles, le produit $P|\Pi$ en aura $n+k$. Par convention, nous supposons que l'attaquant est toujours à droite. Par conséquent, toutes les actions dont les indices sont inférieurs ou égales à n ne font pas intervenir une action de l'intrus, alors que toutes celles dont les indices sont strictement supérieurs à n ne font intervenir que les actions de l'intrus. Une action partielle sera indexée par un entier représentant la composante parallèle dont elle appartient tandis qu'une action de communication sera indexée par une paire d'entiers correspondant aux composantes contenant les actions partielles qu'elle fait intervenir. Soit $\mathcal{I}ndex$, l'ensemble de tous les indices. Formellement, nous avons $\mathcal{I}ndex = \mathbb{N} \cup \mathbb{N}^2$. Nous étendons l'ensemble d'actions $\mathcal{A}ct$ à l'ensemble des actions indexées dénoté par $\mathcal{A}ct \times Ind = \{(\alpha, ID) |\ \alpha \in \mathcal{A}ct \text{ et } ID \in \mathcal{I}ndex\}$.

Étant donné un protocole P attaqué par l'intrus (Π, S), l'évaluation du système $P|\Pi$ selon la stratégie d'attaque S s'éffectue en quatre étapes comme suit :

Réduction : Évaluation de toutes les fonctions probabilistes polynomiales exposées (cf. la Définition 4.2.7) ; ce qui donne le processus bloqué $eval(P|\Pi)$, c'est à dire prêt à exécuter une action s'il en existe au moins une, bien sûr.

Localisation : Indexation des différentes actions exécutables de $eval(P|\Pi)$ grâce à la fonction de localisation χ ci-dessous, qui nous permet de savoir si une action

donnée fait intervenir ou non une action de l'intrus.

Sélection : L'ordonnanceur[2] S (que nous définissons plus loin en détail) choisit une action indexée parmi toutes les actions disponibles.

Exécution : L'action choisie par S est exécutée et le processus recommence jusqu'à ce qu'il n'y ait plus d'actions exécutables.

Soit la fonction support : $\mathcal{A}ct \setminus \{\tau\} \rightarrow \mathcal{C}$ qui, étant donné une action α, retourne le support de l'action, i.e. le canal sur lequel l'action a eu lieu.

Définition 4.2.8 : *La fonction de localisation* χ : $\mathcal{B}locked \longrightarrow 2^{\mathcal{A}ct \times \mathcal{I}nd}$ *est définie récursivement comme suit :*

$$
\begin{aligned}
\chi(\mathbf{0}) &= \emptyset \\
\chi(\alpha.P) &= \{(\alpha, 1)\} \\
\chi(P|Q) &= \chi(P) \cup \{(\alpha, \rho(P) + i) \mid (\alpha, i) \in \chi(Q)\} \\
&\quad \cup \{(\alpha \cdot \overline{\alpha}, i, \rho(P) + j) \mid (\alpha, i) \in \chi(Q) \text{ et } (\overline{\alpha}, j) \in \chi(Q)\} \\
\chi(P \backslash \Gamma) &= \chi(P) \setminus \{(\alpha, id) \in \chi(P) \mid \alpha \in \Gamma\} \\
\chi(P \backslash C) &= (\chi(P) \setminus \{(\alpha, id) \in \chi(P) \mid \text{support}(\alpha) \in C\}) \\
&\quad \cup \{(\tau, i, j) \mid \exists_{\beta \in \mathcal{P}artial}\ (\beta \cdot \overline{\beta}, i, j) \in \chi(P) \text{ et support}(\beta) \in C\}
\end{aligned}
$$

où $\alpha \in \mathcal{P}artial$, $\max(i, (j, k)) = \max(i, j, k)$. *et*

$$
\rho(P) = \begin{cases} \max\{ID \in \mathcal{I}ndex \mid (\beta, ID) \in \chi(P),\ \beta \in \mathcal{V}is\} & \text{si } \chi(P) \neq \emptyset \\ 0 & \text{sinon} \end{cases}
$$

[2]Notons que l'ordonnancement n'est défini que pour les états correspondants aux processus bloqués, puisque dans le cas d'un processus non bloqué, la seule action disponible est l'action interne correspondant à l'évaluation des fonctions.

Actions stratégiquement équiprobables

Maintenant que nous savons comment localiser les différentes actions exécutables à un état donné, il nous reste à définir le processus de sélection lui-même. Mais avant de définir formellement l'ordonnancement, nous allons partitionner l'ensemble d'actions indexées par *classes d'actions stratégiquement équiprobables*, c'est-à-dire les actions qui doivent être choisies de manière uniforme étant donné une stratégie S. Notre approche sera guidée par les principes suivants :

1. Aucune stratégie ne doit distinguer les actions internes d'un protocole. En effet, les actions internes d'un protocole étant invisibles à l'intrus, si la stratégie consiste à laisser évoluer le protocole intérieurement (si c'est possible évidemment), alors si le protocole possède plusieurs actions internes, l'une de ces actions est choisie uniformement et exécutée.
2. Aucune stratégie ne doit permettre à l'intrus d'avoir un contrôle sur la réaction interne du protocole à ses requêtes. En effet, si l'intrus décide, par exemple, d'envoyer un message sur un canal public de lecture du protocole et que le protocole peut le lire à plusieurs positions, alors toutes ces positions doivent avoir la même chance de réagir à la requête de l'attaquant.
3. Quelque soit la stratégie choisie, l'intrus doit avoir un contrôle total sur ses propres actions.

Bien que la fonction χ nous permet de localiser les actions, elle ne permet pas de savoir si une action indexée implique ou non une composante de l'intrus. Or, les trois principes ci-dessus nous montrent que la partition des actions indexées dependra de notre capacité à savoir si une action implique ou non l'intrus. Il nous faut, donc, une information supplémentaire. Pour cela, nous allons définir deux ensembles d'entiers I_1 et I_2 tels que tous les indices correspondant aux positions des composantes du protocole appartiennent à I_1 et ceux correspondant aux composantes de l'intrus

TABLEAU 4.3 *Actions stratégiquement équiprobables*

$$[(\tau,(i,j))]_{I_1\times I_2} = \begin{cases} \{(\tau,(i',j')) \mid i',j' \in I_1\} & \text{si } i,j \in I_1 \\ \{(\tau,(i,j))\} & \text{sinon} \end{cases}$$

$$[(\overline{c}(m),i)]_{I_1\times I_2} = \begin{cases} \{(\overline{c}(m'),i') \mid i' \in I_1 \text{ et } m' \in \mathrm{dom}(c)\} & \text{si } i \in I_1 \\ \{(\overline{c}(m),i)\} & \text{sinon} \end{cases}$$

$$[(c(m),i)]_{I_1\times I_2} = \begin{cases} \{(c(m),j) \mid j \in I_1\} & \text{si } i \in I_1 \\ \{(c(m),i)\} & \text{sinon} \end{cases}$$

$$[(\overline{c}(m)\cdot c(m),(i,j))]_{I_1\times I_2} = \begin{cases} \{(\alpha\cdot\overline{\alpha},(i',j')) \mid i',j' \in I_1 \text{ et } \mathrm{support}(\alpha)=c\} & \text{si } i,j \in I_1 \\ \{(\overline{c}(m')\cdot c(m'),(i',j)) \mid i' \in I_1 \text{ et } m' \in \mathrm{dom}(c)\} & \text{si } i \in I_1 \text{ et } j \in I_2 \\ \{(\overline{c}(m)\cdot c(m),(i,j))\} & \text{sinon} \end{cases}$$

$$[(c(m)\cdot\overline{c}(m),(i,j))]_{I_1\times I_2} = \begin{cases} \{(\alpha\cdot\overline{\alpha},(i',j')) \mid i',j' \in I_1 \text{ et } \mathrm{support}(\alpha)=c\} & \text{si } i,j \in I_1 \\ \{(c(m)\cdot\overline{c}(m),(i',j)) \mid i' \in I_1\} & \text{si } i \in I_1 \text{ et } j \in I_2 \\ \{(c(m)\cdot\overline{c}(m),(i,j))\} & \text{sinon} \end{cases}$$

appartiennent à I_2. Formellement, étant donné un protocole P attaqué par l'intrus (Π, S), nous allons poser

$$I_1 = \begin{cases} \{1,2,\cdots,\rho(eval(P))\} & \text{si } \chi(eval(P)) \neq \emptyset \\ \emptyset & \text{sinon} \end{cases}$$

$$I_2 = \begin{cases} \{\rho(eval(P))+1,\cdots,\rho(eval(P))+\rho(eval(\Pi))\} & \text{si } \chi(eval(\Pi)) \neq \emptyset \\ \emptyset & \text{sinon} \end{cases}$$

conformément à notre convention de placer le processus ennemi à droite. Par exemple, si $P = \overline{c}(m).P_1|\overline{c}(m').P_2$ et $\Pi = c(x).\Pi'$ alors on a :

$\chi(eval(P|\Pi)) =$

$\{(\overline{c}(m),1),(\overline{c}(m'),2),(c(m),3),(c(m'),3),(\overline{c}(m)\cdot c(m),(1,3)),(\overline{c}(m')\cdot c(m'),(2,3))\}$,

$I_1 = \{1,2\}$ et $I_2 = \{3\}$.

Les classes d'équivalence sont données dans le Tableau 4.3 où $[(\alpha, ID)]_{I_1 \times I_2}$ désigne la classe d'équivalence de l'action indexée (α, ID) par rapport aux ensembles I_1 et I_2. Nous les décrivons brièvement ici.

Action interne : Quelle que soit la stratégie choisie, toutes les actions internes d'un protocole sont équiprobables. En effet, la classe $[(\tau, (i,j))]_{I_1 \times I_2}$ nous indique que si l'action τ est indexée par des positions correspondant à celles des composantes du protocole (donc dans I_1), alors elle est équiprobable à toute autre action interne indexée également par des positions référant aux composantes du protocole. Ceci est conforme à notre principe numéro 1. La classe de toute action interne indexée, dont au moins un élément de la paire d'indices ne correspond pas à une position d'une composante du protocole, est réduit à cette dernière. Ce deuxième cas correspond à notre principe numéro 3, puisque forcement les deux éléments de la paire d'indices de l'action τ réfèrent aux positions des composantes de l'intrus, i.e. dans I_2.

Output partiel : Toutes les actions *d'output du protocole sur le même canal public* sont équiprobables. Ceci est conforme à notre deuxième principe qui exige que l'intrus ne puisse pas contrôler les réactions internes du protocole. En effet, dans le cas d'émission, bien que l'attaquant peut choisir le canal public sur lequel écouter (ou simplement observer), il n'a aucun contrôle en ce qui concerne le message qui va être émis sur ce canal. Évidemment, si l'émission est une action de l'attaquant, non seulement il peut choisir le canal, mais il peut également choisir le message qui doit être émis sur ce canal ainsi que la composante (i.e. la position de l'action) qu'il veut utiliser pour monter son attaque. L'exemple de l'intrus $\Pi = \overline{c}(m).\Pi_1 | \overline{c}(m).\Pi_2$ est illustratif. Enfin, un output partiel indexé par une position qui est ni dans I_1 ni dans I_2 est là à titre conventionel pour avoir une relation d'équivalence sur tout l'ensemble $\mathcal{A}ct \times ind$.

Input partiel : Le principe est le même que celui d'output partiel avec la différence notable que, contrairement aux actions d'output partielles, l'attaquant a un

contrôle sur le message reçu par le protocole, puisqu'il provient en principe de "l'extérieur", donc de l'intrus. La seule chose qu'il ne contrôle pas, c'est la position de la composante par laquelle le protocole réagit.

Synchronisation publique : Les mêmes principes gouvernent les actions de synchronisation sur les canaux publics. En effet, l'attaquant peut fixer un canal public du protocole et observer ce qui peut s'y passer. Dans ce cas, puisqu'il ne fait qu'observer sans interagir avec le protocole, toute communication sur ce canal doit faire intervenir deux composantes (output et son input associé) du protocole. Ici évidemment, il n'a aucun contrôle ni sur le contenu du message échangé, ni les positions impliquées. Par contre, si la synchronisation est un output du protocole et un input de l'attaquant, alors il peut décider sur quel canal il écoute ainsi que la position de sa composante d'écoute. Si c'est l'attaquant qui émet et le protocole reçoit, alors non seulement il peut choisir le canal de communication, mais il peut également chosir le message à émettre ainsi que sa composante d'émission. Enfin, si c'est une synchronisation de deux actions de l'intrus, alors il a un contrôle total sur tout, i.e. le canal, le contenu et les composantes impliquées.

L'ordonnanceur externe

Nous sommes maintenant en mesure de définir l'ordonnanceur.

Définition 4.2.9 : /Ordonnenceur] Un ordonnanceur externe est une fonction probabiliste polynomiale *stochastique* $S : 2^{\mathcal{A}ct \times Ind} \times 2^{\mathbb{N}} \times 2^{\mathbb{N}} \rightarrow \mathcal{A}ct \times ind$ satisfaisant pour tout $A \subseteq \mathcal{A}ct \times ind$ non vide, I_1 et $I_2 \subseteq \mathbb{N}$ non tous deux vides et t.q. $\forall i_1 \in I_1, \forall i_2 \in I_2, i_1 < i_2$, les deux conditions suivantes :

1. $\sum_{(\tau,i,j)\in A, i,j\in I_1} \text{Prob}[S(A, I_1, I_2) = (\tau, i, j)] \in \{0, 1\}$.
2. $\forall \alpha, \beta \in A$, $\alpha \in [\beta]_{I_1 \times I_2} \Rightarrow \text{Prob}[S(A, I_1, I_2) = \alpha] = \text{Prob}[S(A, I_1, I_2) = \beta]$.

L'ensemble de tous les ordonnanceurs est dénoté par *Sched.*

Remarque importante. Puisque, étant donné un protocole P attaqué par l'intrus (Π, S), nous savons exactement comment calculer les ensembles d'indexation I_1 et I_2, par soucis d'alléger le texte, nous les supposerons dorénavant implicites dans la suite de ce chapitre. Par conséquent, étant donné un ensemble d'actions indexées A, nous écrivons $S(A)$ simplement, au lieu de $S(A, I_1, I_2)$.

Voici quelques propriétés très intéressantes de l'ordonnancement qui justifient notre manière de définir l'ordonnanceur.

Lemme 4.2.1 *Soit P, un processus tel que $A = \chi(eval(P)) \neq \emptyset$. On a : $\forall S \in Sched$, $\exists \alpha \in A$ une action indexée exécutable t.q.* $\text{Prob}[S(A) = \alpha] \neq 0$. *Autrement dit, l'évaluation du protocole progressera quelque soit l'ordonnanceur choisi.*

La preuve découle de la condition de stochasticité qui est une condition de progression en ce sens qu'à chaque étape d'évaluation quelque chose de positif doit se passer, i.e. une des actions disponibles va être exécutée. C'est une propriété importante en ce sens que l'ordonnancement n'introduit pas de "deadlocks" qui n'ont pas lieu d'être.

Lemme 4.2.2 *L'ordonnancement ne modifie pas les distributions probabilistes des actions internes τ d'un processus.*

La preuve découle des conditions 1 et 2 de la Définition 4.2.9. En effet, d'après la condition 1, on peut permettre à l'attaquant soit, "d'attaquer" le protocole (i.e. d'intercepter un message émis par le protocole, d'émettre un message sur l'un des ses canaux d'écoute, ou tout simplement d'observer un canal public) en choisissant une action visible (disponible bien sûr !) avec une distribution probabiliste conforme à son ordonnanceur. Dans ce cas aucune action interne n'est affectée. Soit de ne rien faire (à condition qu'au moins une action interne soit disponible conformément à la condition

de stochasticité). Dans ce cas, l'une des actions internes est choisie uniformément (condition 2 de la définition) et exécutée sans que leur distribution probabiliste soit modifiée. Ici notre stratégie diffère de celle proposée (Mitchell *et al.*, 2006) qui choisit de privilégier les actions internes au détriment des actions visibles. De ce fait, si le succès d'une attaque dépend de la non réalisation de certaines actions internes, alors leur stratégie ne la détectera pas alors que la notre si. En effet, nous permettons à l'intrus d'attaquer le protocole dès que possible sans tenir compte de l'état interne du processus, alors que leur stratégie ne le lui permet que si le processus ne peut plus évoluer intérieurement. Une telle attaque est possible si, par exemple, son succès dépend d'une variable booléenne du système attaqué et qu'une action interne du système peut modifier la valeur de cette variable permettant ainsi de rendre l'attaque possible ou non. En effet, considérons le processus suivant :

$$P = (c(x).\overline{c'}(x).\mathbf{0}|\overline{c'}(1).\mathbf{0}|c'(y).[y = 0]\overline{c}(secret).\mathbf{0})\backslash\{c'\}.$$

Si nous privilégions les actions internes, alors ce processus ne pourra jamais émettre le secret sur le canal public c alors qu'autrement l'attaquant pourra émettre 0 sur le canal public de lecture $c(x)$ et ainsi permettre au processus de publier le secret. Ce protocole, qui a une faille de sécurité évidente, serait considéré sûr d'après leur stratégie.

Lemme 4.2.3 *L'ordonnancement n'influence que globalement la distribution probabiliste des actions impliquant des outputs du protocole sur un même canal. Autrement dit, l'attaquant peut influer sur la distribution probabiliste des canaux d'output du protocole, mais pas sur leurs contenus.*

La preuve découle de la condition 2 de la Définition 4.2.9. En effet, cette condition exige que si plusieurs actions différentes d'output sur un même canal sont exécutables dans un état donné, alors, si l'ordonnanceur choisit d'écouter sur ce canal selon une

distribution quelconque, cette distribution est redistribuée de la même façon sur toutes les actions d'output sur ce canal. Ceci est tout à fait normal, puisque, dans le cas des actions d'output, l'intrus a certes un contrôle sur les canaux, mais est passif quand à la nature du message que le processus va émettre sur le canal choisi. En effet, considérons le processus $P = \overline{c}(m).\mathbf{0}|\overline{c}(m').\mathbf{0}$ qui envoie sur le canal public c soit le message m suivi de m' soit le message m' suivi de m. Tout attaquant (Π, S) doit être complètement passif devant un tel processus, puisque la seule attaque sur le processus P qu'il peut faire est d'écouter sur le canal c. Tout ordonnanceur, qui pourrait choisir avec des probabilités différentes non nulles les deux actions, serait en dehors de la capacité réelle d'un attaquant. Ici encore, notre stratégie diffère de celle de Mitchell *et al.*. En effet, selon leur stratégie l'attaquant peut décider que, par exemple, dans 75% des cas, le premier message émis sera m et dans 25%, il sera m'. Ce qui est absurde compte tenu de sa passivité totale devant un tel processus. Avec notre stratégie, ce processus émettra l'un ou l'autre message avec la même probabilité quelque soit l'ordonnanceur, ce qui est normal. En effet, nous avons $\{(\overline{c}(m), 1), (\overline{c}(m'), 2)\} \subset A = \chi(eval(P|\Pi))$ et d'après la condition 2, on a : $\text{Prob}[S(A) = (\overline{c}(m'), 2)] = \text{Prob}[S(A) = (\overline{c}(m), 1)]$, puisque les deux actions indexées sont équivalentes. Ce qui reflète bien la nature totalement non déterministe de P.

Lemme 4.2.4 *L'attaquant a une influence totale sur la distribution probabiliste de ses propres actions.*

Ceci est normal compte tenu qu'une stratégie d'attaque ne doit pas être non-déterministe. Un attaquant, composé de plusieurs processus en parallèle, doit pouvoir décider quelle composante privilégier pour monter son attaque. Enfin, pour clore cette section sur l'ordonnanceur, nous avons cet important résultat.

Théorème 4.2.1 *La somme des probabilités des transitions sortantes d'un sommet selon un ordonnanceur quelconque est inférieure ou égale à 1.*

Preuve: Soit P un sommet donné et Exec(P) l'ensemble des transitions sortant de P. Deux cas se présentent :

[$P \notin \mathcal{B}locked$] : Dans ce cas, il n'y a pas d'ordonnencement et

$$\text{Exec}(P) = \{P \xrightarrow{\tau[q]} Q \mid \mathit{eval}(\text{P}) \hookrightarrow^{q} Q\}.$$

La somme des probabilités des transitions sortant de P est

$$\begin{aligned} p &= \sum_{Q \in Im(\mathit{eval}(\text{P}))} \text{Prob}[\mathit{eval}(\text{P}) = Q] \\ &\leq 1 \qquad \text{par définition de } \mathit{eval}, \end{aligned}$$

puisque c'est une fonction probabiliste polynomiale.

[$P \in \mathcal{B}locked$] : Dans ce cas, on a :

$$\text{Exec}(P) = \{t_{ID} = P \xrightarrow{\alpha[q_{ID}]} Q \mid (\alpha, ID) \in \chi(P)\}.$$

La somme des probabilités des transitions sortant de P selon l'ordonnenceur S est

$$\begin{aligned} p &= \sum_{(\alpha,ID)\in\chi(P)} \text{Prob}[S(\chi(P)) = (\alpha, ID)] \times q_{ID} \\ &\leq \sum_{(\alpha,ID)\in\chi(P)} \text{Prob}[S(\chi(P)) = (\alpha, ID)] \qquad \text{puisque } \forall ID \in \mathcal{I}ndex,\ q_{ID} \leq 1 \\ &\leq 1 \qquad \text{par définition de } S. \end{aligned}$$

■

4.2.6 Probabilité cumulative

On voudrait parfois calculer à partir d'un sommet donné, la probabilité d'atteindre un ensemble des sommets. Ainsi, on pourrait être ammené à calculer la somme des probabilités de plusieurs chemins ayant la même source. Pour être sûr d'avoir bien compté tous les chemins, il est indispensable d'étiqueter de manière unique chaque chemin.

Pour les sommets correspondant aux processus bloqués, nous disposons déjà d'un moyen d'étiquetage : la fonction de localisation χ. Il nous reste à trouver un moyen d'étiqueter les transitions sortantes d'un processus non bloqué. Or, si P est non bloqué, on sait qu'il existe un nombre fini $n = |Im(eval(P))|$ des processus Q_i ($1 \leq i \leq n$) t.q. $\exists q_i \neq 0$ et $P \xrightarrow{\tau[q_i]} Q_i$ est une transition sortante de P. Nous allons classer les processus Q_i de 1 à n et utiliser ce classement[3] pour étiqueter les transitions sortantes de P. Ainsi, si la transition $P \xrightarrow{\tau[q_i]} Q_i$ sera étiquetée par la paire $(\tau, (i, i))$ par analogie à l'indexation des actions τ par la fonction χ.

Soit $\sigma = (\alpha_1, id_1)(\alpha_2, id_2)\ldots(\alpha_n, id_n)$, une séquence d'actions indexées. Nous disons que σ est un chemin du processus P au processus Q s'il existe des probabilités non nulles $p_1, p_2, \ldots, p_n$ telles que $P_0 \xrightarrow{(\alpha_1, id_1)[p_1]} P_1 \xrightarrow{(\alpha_2, id_2)[p_2]} \cdots \xrightarrow{(\alpha_n, id_n)[p_n]} P_n$, $P = P_0$ et $Q = P_n$. De même, nous disons que P atteint Q par le chemin σ avec la probabilité p suivant l'ordonnancement S, dénoté par $P \stackrel{\sigma[p]}{\Longrightarrow}_S Q$, si la probabilité que S choisit σ, définie par

$$\text{Prob}[S(\sigma)] = \prod_{1 \leq i \leq n} q_i \text{ où } q_i = \begin{cases} p_i & \text{si } P_{i-1} \notin \mathcal{B}locked \\ \text{Prob}[S(\chi(P_{i-1})) = (\alpha_i, id_i)] \times p_i) & \text{sinon} \end{cases}$$

est égale à p.

[3]Techniquement, ce classement peut être fait grâce aux résultats des fonctions évaluées, la localisation de leurs composantes, ainsi que l'ordre lexicographique

Exemple. Soit P, le processus dont le système de transitions est illustré dans la Figure 4.2 où $P_1 \notin \mathcal{B}locked$ et $\text{Prob}[eval(P_1) = P_i] = p_i$ pour $3 \leq i \leq 5$. Soit S l'ordonnanceur tel que

- $\text{Prob}[S(\{(\alpha_1, 1); (\alpha_2, 2)\}) = (\alpha_1, 1)] = q$
- $\text{Prob}[S(\{(\alpha_1, 1); (\alpha_2, 2)\}) = (\alpha_2, 2)] = 1 - q^4$
- $\text{Prob}[S(\{(\alpha_3, 1)\}) = (\alpha_3, 1)] = 1$ (conformément à la condition de stochasticité).

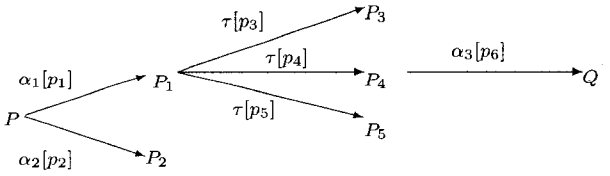

FIGURE 4.2 *Système de transitions de P*

Alors la probabilité que P atteigne Q par le chemin $\sigma = (\alpha_1, 1)(\tau, (2, 2))(\alpha_3, 1)$, selon l'ordonnanceur S, est

$$\text{Prob}[S(\sigma)] = (p_1 \times q) \times p_4 \times (p_6 \times 1).$$

Soit α une action, nous désignons par *α-chemins* l'ensemble des chemins de la forme $(\tau, id_1)(\tau, id_2) \ldots (\tau, id_{n-1})(\alpha, id_n)$, c'est-à-dire un chemin terminant par l'action α précédée d'un nombre fini (éventuellement nul) de τ. La notation $P \stackrel{\alpha[p]}{\Longrightarrow}_S Q$ signifie qu'il existe un α-chemin σ tel que $P \stackrel{\sigma[p]}{\Longrightarrow}_S Q$. De même, $P \stackrel{\hat{\alpha}[p]}{\Longrightarrow}_S Q$ désigne $P \stackrel{\alpha[p]}{\Longrightarrow}_S Q$ si $\alpha \neq \tau$ et $P \stackrel{\hat{\tau}[p]}{\Longrightarrow}_S Q$ sinon, i.e. un chemin (éventuellement vide !) formé uniquement des τ.

Soit un ensemble $\mathcal{E} \subseteq \mathcal{P}roc$ de processus et $Q \in \mathcal{E}$. Soit P un processus et σ un α-chemin de P à Q, nous dirons que σ est *minimal* conformément à $\mathcal{E}$ s'il n'existe

[4]Notons que si $(\alpha_1, 1)$ et $(\alpha_2, 2)$ sont équiprobables alors $q = 1 - q$, i.e. $q = \frac{1}{2}$.

pas un autre α-chemin σ' de P à Q' tel que σ' est un préfixe[5] de σ et $Q' \in \mathscr{E}$.

Définition 4.2.10 : *Soit un ensemble $\mathscr{E} \subseteq \mathcal{P}roc$ de processus, P un processus et α une action. Nous désignons par $Paths(P, \stackrel{\alpha}{\Longrightarrow}, \mathscr{E})$ l'ensemble de tous les α-chemins minimaux de P vers un élément de $\mathscr{E}$.*

Notons que la condition de minimalité ne s'applique qu'aux τ-chemins.

Définition 4.2.11 : *Soit un ensemble $\mathscr{E} \subseteq \mathcal{P}roc$ de processus, P un processus, S un ordonnanceur et α une action. La probabilité totale que le processus P atteigne un processus dans $\mathscr{E}$ par un α-chemin selon l'ordonnanceur S est calculée par la* fonction de probabilité cumulative
$\mu : \mathcal{P}roc \times \mathcal{A}ct \times 2^{\mathcal{P}roc} \times Sched \rightarrow [0,1]$ *définie par*

$$\mu(P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathscr{E}) = \begin{cases} 1 & si\ P \in \mathscr{E}\ et\ \alpha = \tau \\ \displaystyle\sum_{\sigma \in Paths(P, \stackrel{\alpha}{\Longrightarrow}, \mathscr{E})} \mathrm{Prob}[S(\sigma)] & sinon \end{cases}$$

Nous pouvons démontrer (par une induction sur la longueur des α-chemins) le théorème ci-dessous.

Théorème 4.2.2 *La fonction de probabilité cumulative est bien définie, i.e.*

$$\forall_{P,\alpha,\mathscr{E},S} \quad \mu(P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathscr{E}) \leq 1.$$

[5]Le préfixage ne concerne que les actions et non leur indexation. Ainsi, (α_1, id_3) est un préfixe de $(\alpha_1, id_1)(\alpha_2, id_2)$ quelque soit les indexs id_1 et id_3.

4.3 Équivalences asymptotiques

Dans cette section, nous voulons établir une équivalence qui nous permet de dire qu'un protocole P vérifie une propriété de sécurité si et seulement s'il est observationnellement équivalent à une idéalisation Q du protocole qui vérifie la propriété de sécurité par construction. Nous voulons que deux processus soient équivalents, si et seulement si, soumis aux mêmes attaques, ils engendrent "approximativement" les mêmes observations. Par approximativement, nous voulons dire *asymptotiquement* proches par rapport au paramètre de sécurité. En effet, notre objectif est de vérifier des protocoles de sécurité dont la sécurité des primitives cryptographiques dépend de la valeur du paramètre de sécurité. Il est d'usage dans les modèles computationels de considérer qu'un protocole peut ne pas être sécuritaire pour certaines valeurs du paramètre de sécurité, mais si on augmente suffisamment le paramètre de sécurité alors aucun adversaire de puissance de calcul en temps polynomial ne peut attaquer le protocole.

Dans la suite de ce chapitre, pour les besoins de vérification, nous ne considérons que les actions réelles, c'est-à-dire les actions de communication $\mathcal{A}ctual$. En effet, les actions partielles ont été considérées, jusqu'à présent, par souci de cohérence et de complétude de la sémantique, mais en pratique, aucune action partielle n'est exécutable. Ainsi, nous allons restreindre la fonction de localisation χ aux actions $\mathcal{A}ctual$ comme suit : $\overline{\chi} : \mathcal{B}locked \longrightarrow 2^{\mathcal{A}ctual \times Ind}$ avec

$$\overline{\chi}(P) = \chi(P) \cap \mathcal{A}ctual \times Ind.$$

Ainsi donc l'ordonnanceur ne choisira que des actions de réelles dans $\overline{\chi}(P)$.

4.3.1 Équivalence observationnelle asymptotique

Nous allons définir une équivalence observationnelle due à Mitchell *et al.*, 2006, dans le contexte ProSPA.

Nous commençons par définir la probabilité d'observer une action visible qui nous servira à établir l'équivalence observationnelle. Nous allons définir la probabilité d'observer une action visible α comme somme de la probabilité d'observer directement α, i.e. la probabilité cumulative de faire un α-chemin pour atteindre n'importe quel état plus la probabilité de l'observer indirectement, i.e. en observant d'abord des actions visibles différentes de α avant de l'observer. Pour cela, nous avons besoin d'étendre la notion de probabilité cumulative à une notion de probabilité cumulative plus générale que nous appelons *probabilité cumulative "up to H"* où $H \subset \mathcal{A}ctual \setminus \{\tau\}$.

Définition 4.3.1 : *Soit un ensemble $\mathscr{E} \subseteq \mathcal{P}roc$ de processus, P un processus, S un ordonnanceur et H un ensemble d'actions visibles dans $\mathcal{A}ctual \setminus \{\tau\}$. La* probabilité cumulative "up to H" *est définie récursivement comme suit : $\forall \alpha \in \mathcal{A}ctual \setminus H$*

$$\mu(P, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/H}, \mathscr{E}) = \begin{cases} 1 & si\ P \in \mathscr{E}\ and\ \alpha = \tau, \\ \mu(P, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S}, \mathscr{E}) + \sum\limits_{\substack{\beta \in H, \\ Q \in \mathcal{P}roc}} \mu(P, \stackrel{\hat{\beta}}{\Longrightarrow}_{S}, \{Q\}) \mu(Q, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/H}, \mathscr{E}) & sinon \end{cases}$$

Lemme 4.3.1 *La fonction de probabilité cumulative up to H est bien définie, i.e.*

$$\forall_{P,\alpha,\mathscr{E},S,H} \quad \mu(P, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/H}, \mathscr{E}) \leq 1.$$

Preuve: Si $P \in \mathscr{E}$ et $\alpha = \tau$ alors $\mu(P, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/H}, \mathscr{E}) = 1$ et le résultat en découle. Supposons qu'on n'est pas dans ce cas. Soit $\sigma = \tau^* \alpha_1 \cdots \tau^* \alpha_n$, un chemin formé

de n actions réelles visibles séparées par un nombre fini d'actions τ (nous omettons l'indexation des actions), P un processus et S un ordonnanceur. Nous disons que l'action α_i $(1 \leq i \leq n)$ est à une distance i du processus P sur le chemin σ selon l'ordonnanceur S, dénoté par $d_\sigma(P, \alpha_i, S) = i$ s'il existe une probabilité non nulle p et un processus Q tels que $P \stackrel{\sigma[p]}{\Longrightarrow}_S Q$ et $\forall j < i \ \alpha_j \neq \alpha_i$. Autrement dit, $d_\sigma(P, \alpha, S)$ est la première position de l'action visible α sur le chemin σ selon S. De même, étant donné $\alpha \notin H$ une action visible et

$$\Sigma_H(\alpha) = \{\sigma = \tau^*\alpha_1\tau^*\alpha_2 \cdots \tau^*\alpha_n\tau^*\alpha \mid n \in \mathbb{N} \text{ et } \forall i \leq n, \ \alpha_i \in H\},$$

nous disons que l'action α est à une distance maximale k du processus P conformement à l'ensemble H selon l'ordonnanceur S, dénoté par $d_H(P, \alpha, S) = k$ si $\sup_{\sigma \in \Sigma_H(\alpha)}(d_\sigma(P, \alpha, S) = k$. Nous allons prouver le théorème par une induction sur $d_H(P, \alpha, S)$. Soit $\alpha \in \mathcal{A}ctual \setminus (H \cup \{\tau\})$ nous avons :

- *[Base :]* Si $d_H(P, \alpha, S) = 0$ alors

$$\mu(P, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/H}, \mathscr{E}) = \mu(P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathcal{P}roc) \leq 1.$$

  La base d'induction est donc établie.

- *[Etape d'induction :]* Supposons, maintenant, que pour tout P et H satisfaisant $d_H(P, \alpha, S) < n$ on ait $\mu(P, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/H}, \mathscr{E}) \leq 1$. Puisque

$$\mu(P, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/H}, \mathscr{E}) = \mu(P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathscr{E}) + \sum_{\beta \in H, \ Q \in \mathcal{P}roc} \mu(P, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \{Q\})\mu(Q, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/H}, \mathscr{E})$$

  et que l'hyphothèse d'induction nous garantit que $\mu(Q, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/H}, \mathscr{E}) \leq 1$, il nous

reste à montrer que

$$\mu(P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathscr{E}) + \sum_{\beta \in H,\ Q \in \mathcal{P}roc} \mu(P, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \{Q\}) \leq 1$$

Or, nous avons

$$\mu(P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathscr{E}) \ + \sum_{\beta \in H,\ Q \in \mathcal{P}roc} \mu(P, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \{Q\}) \leq \sum_{\beta \in H \cup \{\alpha\}} \mu(P, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \mathcal{P}roc)$$

Nous terminons la preuve en remarquant que $\sum_{\beta \in H \cup \{\alpha\}} \mu(P, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \mathcal{P}roc)$ est inférieur à la somme des probabilités des transitions sortantes de P selon l'ordonnanceur S qui est inférieure à 1 d'après le Théorème 4.2.1.

■

Définition 4.3.2 : *Soit $\alpha \in \mathcal{A}ctual \setminus \{\tau\}$, une action observable et P un processus. La probabilité que P génère α selon l'ordonnanceur S est*

$$\text{Prob}[P \rightsquigarrow_S \alpha] = \mu(P, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/(\mathcal{A}ctual \setminus \{\alpha\})}, \mathcal{P}roc)$$

Avant de définir notre relation d'équivalence observationnelle, nous allons montrer que $\text{Prob}[P \rightsquigarrow_S \alpha]$ est bien une probabilité.

Lemme 4.3.2 *La probabilité d'observer une action visible est bien définie, ie*

$$\forall_{P,\alpha,S} \quad \text{Prob}[P \rightsquigarrow_S \alpha] \leq 1.$$

Preuve: Le résultat découle du lemme 4.3.1. ■

Nous étendons l'observation d'une action visible à celle d'un ensemble d'actions visibles comme suit :

Définition 4.3.3 : *Soit un ensemble* $\Gamma \subset \mathcal{A}ctual \setminus \{\tau\}$ *d'actions visibles et* P *un processus. La probabilité que* P *génère des actions dans* Γ *selon l'ordonnanceur* S *est*

$$\text{Prob}[P \rightsquigarrow_S \Gamma] = \sum_{\alpha \in \Gamma} \mu(P, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/(\mathcal{A}ctual \setminus \Gamma)}, \mathcal{P}roc).$$

Avant de définir notre relation d'équivalence observationnelle, nous allons montrer que $\text{Prob}[P \rightsquigarrow_S \Gamma]$ est bien une probabilité.

Lemme 4.3.3 *La probabilité d'observer un ensemble d'actions visibles est bien définie, i.e.*

$$\forall_{P,\alpha,S} \quad \text{Prob}[P \rightsquigarrow_S \Gamma] \leq 1.$$

Preuve: Soit P, un processus, S un ordonnanceur, un ensemble $\Gamma \subset \mathcal{A}ctual \setminus \{\tau\}$ d'actions visibles et $H = \mathcal{A}ctual \setminus (\Gamma \cup \{\tau\})$. Nous utilisons une induction sur la distance $d_H(P, \alpha, S)$ définie dans la preuve du lemme 4.3.1.

- *[Base :]* Si $d_H(P, \alpha, S) = 0 \ \forall \alpha \in \Gamma$ alors,

$$\begin{aligned}
\text{Prob}[P \rightsquigarrow_S \Gamma] &= \sum_{\alpha \in \Gamma} \mu(P, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/(\mathcal{A}ctual \setminus \Gamma)}, \mathcal{P}roc) \\
&= \sum_{\alpha \in \Gamma} \mu(P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathcal{P}roc) \\
&\leq \sum_{\beta \in \mathcal{A}ct} \mu(P, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \mathcal{P}roc) \\
&\leq 1 \quad \text{, (d'après le Théorème 4.2.1).}
\end{aligned}$$

La base d'induction est donc établie.

- *[Etape d'induction :]* Supposons maintenant que si pour tout $\alpha \in \Gamma$, et tout pro-

cessus Q t.q. $d_H(Q, \alpha, S) < n$ alors $\text{Prob}[Q \rightsquigarrow_S \Gamma] \leq 1$.

Nous avons

$$\begin{aligned}
\text{Prob}[P \rightsquigarrow_S \Gamma] &= \sum_{\alpha \in \Gamma} \mu(P, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/(\mathcal{A}ctual \setminus \Gamma)}, \mathcal{P}roc) \\
&= \sum_{\alpha \in \Gamma} \mu(P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathcal{P}roc) + \\
&\quad \sum_{\alpha \in \Gamma} \sum_{\beta \in H,\ Q \in \mathcal{P}roc} \mu(P, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \{Q\}) \mu(Q, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/H}, \mathcal{E}).
\end{aligned}$$

C'est-à-dire

$$\begin{aligned}
\text{Prob}[P \rightsquigarrow_S \Gamma] &= \sum_{\alpha \in \Gamma} \mu(P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathcal{P}roc) + \\
&\quad \sum_{\beta \in H,\ Q \in \mathcal{P}roc} \mu(P, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \{Q\}) \text{Prob}[Q \rightsquigarrow_S \Gamma] \\
&\leq \sum_{\alpha \in \Gamma} \mu(P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathcal{P}roc) + \\
&\quad \sum_{\beta \in H,\ Q \in \mathcal{P}roc} \mu(P, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \{Q\}) \quad (\text{ par hypothèse d'induction}) \\
&\leq \sum_{\beta \in \mathcal{A}ct} \mu(P, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \mathcal{P}roc) \\
&\leq 1 \quad , (\text{d'après le Théorème 4.2.1}).
\end{aligned}$$

■

Maintenant, nous pouvons définir ce qu'on entend par une observation. Une observation o est tout simplement un canal public et un message, i.e. $o = (c, m)$. Nous dénotons par $\mathcal{O}bs$ l'ensemble de toutes les observations.

Définition 4.3.4 : *Soit $o = (c, m) \in \mathcal{O}bs$, une observation et P un processus. La probabilité que P génère o selon l'ordonnanceur S est*

$$\text{Prob}[P \rightsquigarrow_S o] = \text{Prob}[P \rightsquigarrow_S \{\overline{c}(m) \cdot c(m),\ c(m) \cdot \overline{c}(m)\}].$$

Nous pouvons maintenant définir notre relation d'équivalence observationnelle asymptotique qui stipule que deux processus sont équivalents s'ils engendrent les mêmes observations avec approximativement les mêmes probabilités lorsqu'ils sont attaqués par le même attaquant.

Définition 4.3.5 : *[Équivalence observationnelle asymptotique] Soit $\mathcal{P}oly$, l'ensemble des polynômes positifs de la forme $q : \mathbb{N} \rightarrow \mathbb{R}^+$. Deux processus P et Q sont asymptotiquement équivalents, dénoté par $P \simeq Q$, si et seulement si $\forall_{q \in \mathcal{P}oly}$, $\forall_{o \in \mathcal{O}bs}$, $\forall_{\Pi \in \mathcal{E}nemy}$, $\forall_{S \in Sched}$, $\exists_{i_0}$ t.q. $\forall_{\mathbf{N} \geq i_0}$*

$$|\text{Prob}[P|\Pi \rightsquigarrow_S o] - \text{Prob}[Q|\Pi \rightsquigarrow_S o]| \leq \frac{1}{q(\mathbf{N})}$$

Théorème 4.3.1 *L'équivalence observationnelle asymptotique $\simeq$ est une relation d'équivalence.*

Preuve: La réflexivité et la symétrie sont évidentes. Nous allons donc prouver la transitivité. Soit donc P, Q et R des processus tels que $P \simeq Q$ et $Q \simeq R$. Soit $q \in Poly$, $o \in \mathcal{O}bs$, $\forall \Pi \in \mathcal{E}nemy$ et $S \in Sched$, alors on a :
$P \simeq Q \Rightarrow \exists i_0$ tel que $\forall \mathbf{N} \geq i_0$

$$|\text{Prob}[P|\Pi \rightsquigarrow_S o] - \text{Prob}[Q|\Pi \rightsquigarrow_S o]| \leq \frac{1}{2q(\mathbf{N})},$$

et $Q \simeq R \Rightarrow \exists j_0$ tel que $\forall \mathbf{N} \geq j_0$

$$|\text{Prob}[Q|\Pi \rightsquigarrow_S o] - \text{Prob}[R|\Pi \rightsquigarrow_S o]| \leq \frac{1}{2q(\mathbf{N})}.$$

Nous avons donc $\forall \mathbf{N} \geq k_0 = \max(i_0, j_0)$

$|\text{Prob}[P|\Pi \rightsquigarrow_S o] - \text{Prob}[R|\Pi \rightsquigarrow_S o]|$

$$
\begin{aligned}
&= |\text{Prob}[P|\Pi \rightsquigarrow_S o] \;-\; \text{Prob}[Q|\Pi \rightsquigarrow_S o] \\
&\qquad\qquad\qquad\quad +\; \text{Prob}[Q|\Pi \rightsquigarrow_S o] - \text{Prob}[R|\Pi \rightsquigarrow_S o]| \\
&\leq |\text{Prob}[P|\Pi \rightsquigarrow_S o] \;-\; \text{Prob}[Q|\Pi \rightsquigarrow_S o]| \\
&\qquad\qquad\qquad\quad +\; |\text{Prob}[Q|\Pi \rightsquigarrow_S o] - \text{Prob}[R|\Pi \rightsquigarrow_S o]| \\
&\leq \frac{1}{2q(\mathbf{N})} \;+\; \frac{1}{2q(\mathbf{N})} = \frac{1}{q(\mathbf{N})}.
\end{aligned}
$$

D'où $P \simeq R$. ■

Bien que très pratique, telle que définie, l'équivalence observationnelle s'avère non adaptée pour l'analyse de plusieurs systèmes pour lesquels on est souvent amené à calculer la probabilité d'une suite d'observables comme nous allons le voir à la fin du chapitre dans notre étude de cas. Pour remédier à ce problème, nous allons étendre dans la section qui suit l'équivalence observationnelle pour inclure toutes les traces observables.

4.3.2 Équivalence de trace asymptotique

Cette section a pour but d'établir une équivalence qui nous permet de dire que deux processus sont équivalents si et seulement si, soumis aux mêmes attaques, ils engendrent les mêmes traces observables avec "approximativement" les mêmes probabilités. Par approximativement, nous voulons dire des probabilités *asymptotiquement* proches par rapport au paramètre de sécurité. Nous allons montrer que c'est une reformulation de l'équivalence observationnelle asymptotique, en d'autres termes les deux formulations sont équivalentes. Nous commençons d'abord par définir la probabilité d'observer une suite d'observables qui nous servira à établir notre nouvelle équivalence. Nous définissons la probabilité d'observer une suite d'observables $o_1 o_2 \cdots o_n$

récursivement comme la probabilité d'observer directement o_1 pour atteindre n'importe quel état fois la probabilité que cet état engendre $o_2 \cdots o_n$.

Définition 4.3.6 : *Soit P un processus et $o_1 o_2 \cdots o_n$, une suite d'actions observables t.q $\forall i \leq n, o_i = (c_i, m_i)$. Soit $\alpha_i = \overline{c_i}(m_i) \cdot c_i(m_i)$ et $\beta_i = c_i(m_i) \cdot \overline{c_i}(m_i)$ $\forall 1 \leq i \leq n$. La probabilité que P génère $o_1 o_2 \cdots o_n$ selon l'ordonnanceur S est*

$$\begin{aligned} &\text{Prob}[\ P \ \rightsquigarrow_S^{tr} o_1 o_2 \cdots o_n] \\ &\quad = \sum_{Q \in \mathcal{P}roc} (\mu(P, \stackrel{\hat{\alpha_1}}{\Longrightarrow}_S, \{Q\}) + \mu(P, \stackrel{\hat{\beta_1}}{\Longrightarrow}_S, \{Q\}))\text{Prob}[Q \rightsquigarrow_S^{tr} o_2 \cdots o_n]. \end{aligned}$$

Avant de définir notre équivalence, nous devons nous assurer que la probabilité d'observer une séquence d'actions visibles est bien définie.

Lemme 4.3.4 *La probabilité d'observer une suite d'actions visibles est bien définie, i.e., $\forall P, o_1, o_2, \cdots, o_n, S$ $\text{Prob}[P \rightsquigarrow_S^{tr} o_1 o_2 \cdots o_n] \leq 1$.*

Preuve: Une simple induction sur la longueur des suites et le Théorème 4.2.2. ■

Nous pouvons, maintenant, définir notre relation d'équivalence de trace asymptotique qui stipule que deux processus sont équivalents si et seulement s'ils engendrent les mêmes traces observables avec approximativement les mêmes probabilités lorsqu'ils sont soumis aux mêmes attaques, i.e. lorsqu'ils évoluent dans le même environnement et selon le même ordonnanceur.

Définition 4.3.7 : *[Équivalence de trace asymptotique] Deux processus P et Q sont asymptotiquement trace équivalents, dénoté par $P \simeq^{tr} Q$, si et seulement si $\forall q \in \mathcal{P}oly$, $\forall o_1, o_2, \cdots, o_n \in \mathcal{O}bs$, $\forall \Pi \in \mathcal{E}nemy$, $\forall S \in Sched$, $\exists i_0$ tel que $\forall \mathbf{N} \geq i_0$*

$$|\text{Prob}[P|\Pi \rightsquigarrow_S^{tr} o_1 o_2 \cdots o_n] - \text{Prob}[Q|\Pi \rightsquigarrow_S^{tr} o_1 o_2 \cdots o_n]| \leq \frac{1}{q(\mathbf{N})}.$$

Théorème 4.3.2 *L'équivalence de trace asymptotique $\simeq^{tr}$ est une relation d'équivalence.*

Preuve: La réflexivité et la symétrie sont évidentes. Nous allons donc prouver la transitivité. Soit donc P, Q et R des processus tels que $P \simeq Q$ et $Q \simeq R$. Soit $q \in Poly$, s une séquence d'observations, Π un attaquant et $S \in Sched$. Alors on a : $P \simeq^{tr} Q \Rightarrow \exists i_0$ tel que $\forall \mathbf{N} \geq i_0$

$$|\text{Prob}[P|\Pi \rightsquigarrow_S^{tr} s] - \text{Prob}[Q|\Pi \rightsquigarrow_S^{tr} s]| \leq \frac{1}{2q(\mathbf{N})},$$

et $Q \simeq^{tr} R \Rightarrow \exists j_0$ tel que $\forall \mathbf{N} \geq j_0$

$$|\text{Prob}[Q|\Pi \rightsquigarrow_S^{tr} s] - \text{Prob}[R|\Pi \rightsquigarrow_S^{tr} s]| \leq \frac{1}{2q(\mathbf{N})}.$$

Nous avons donc $\forall \mathbf{N} \geq k_0 = \max(i_0, j_0)$

$|\text{Prob}[P|\Pi \rightsquigarrow_S^{tr} s] - \text{Prob}[R|\Pi \rightsquigarrow_S^{tr} s]|$

$$\begin{aligned}
&= |\text{Prob}[P|\Pi \rightsquigarrow_S^{tr} s] - \text{Prob}[Q|\Pi \rightsquigarrow_S^{tr} s] \\
&\qquad + \text{Prob}[Q|\Pi \rightsquigarrow_S^{tr} s] - \text{Prob}[R|\Pi \rightsquigarrow_S^{tr} s]| \\
&\leq |\text{Prob}[P|\Pi \rightsquigarrow_S^{tr} s] - \text{Prob}[Q|\Pi \rightsquigarrow_S^{tr} s]| \\
&\qquad + |\text{Prob}[Q|\Pi \rightsquigarrow_S^{tr} s] - \text{Prob}[R|\Pi \rightsquigarrow_S^{tr} s]| \\
&\leq \frac{1}{2q(\mathbf{N})} + \frac{1}{2q(\mathbf{N})} = \frac{1}{q(\mathbf{N})}.
\end{aligned}$$

D'où $P \simeq^{tr} R$. ■

Nous avons le résultat suivant, qui établit que l'équivalence de trace est une reformulation de l'équivalence observationnelle.

Théorème 4.3.3 $\forall_{P,Q \in \mathcal{P}roc} P \simeq^{tr} Q \Leftrightarrow P \simeq Q$.

Preuve: ($\Rightarrow$). Soit un ensemble d'actions visibles $H \subset \mathcal{A}ctual \setminus \{\tau\}$, un observable $o = (c, m)$ t.q. $\{\overline{c}(m) \cdot c(m), c(m) \cdot \overline{c}(m)\} \cap H = \emptyset$, S un ordonnanceur et R un processus. Alors, $d_H(R, o, S)$ dénote

$$d_H(R, o, S) = \max(d_H(R, \overline{c}(m) \cdot c(m), S), d_H(R, c(m) \cdot \overline{c}(m), S)).$$

Soit O, un ensemble d'observables t.q. $o \notin O$, et $Tr_k^O(R, o, S)$ l'ensemble

$$Tr_k^O(P, o, S) = \{s = o_1 o_2 \cdots o_k o | \forall i \leq k, o_i \in O \text{ et } \text{Prob}[R \rightsquigarrow_S^{tr} s] \neq 0\}.$$

Alors, pour tout observable o, nous avons

$$\text{Prob}[R \rightsquigarrow_S o] = \sum_{k \geq 0} \sum_{s \in Tr_k^{\mathcal{O}bs \setminus \{o\}}(R,o,S)} \text{Prob}[R \rightsquigarrow_S^{tr} s].$$

Soit P et Q, deux processus t.q. $P \simeq^{tr} Q$, $o = (c, m) \in \mathcal{O}bs$, $(\Pi, S) \in \mathcal{E}nemy$, $q \in Poly$ et

$$H = \{\alpha \mid \exists (c', m') \in \mathcal{O}bs \setminus \{o\} \text{ et } \alpha \in \{\overline{c'}(m') \cdot c'(m'), c'(m') \cdot \overline{c'}(m')\}\}.$$

Nous prouvons le théorème par induction sur

$$\max(d_H(P|\Pi, o, S), d_H(Q|\Pi, o, S)).$$

[Base :] Si $d_H(P|\Pi, o, S) = d_H(Q|\Pi, o, S) = 0$, alors, nous avons :

$$\text{Prob}[P|\Pi \rightsquigarrow_S o] = \mu(P|\Pi, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathcal{P}roc) + \mu(P|\Pi, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \mathcal{P}roc)$$

où $\alpha = \overline{c}(m) \cdot c(m)$ et $\beta = c(m) \cdot \overline{c}(m)$. De même,

$$\text{Prob}[Q|\Pi \rightsquigarrow_S o] = \mu(Q|\Pi, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathcal{P}roc) + \mu(Q|\Pi, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \mathcal{P}roc).$$

$P \simeq^{tr} Q \Rightarrow$

$$\begin{aligned} |\mu(P|\Pi, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathcal{P}roc) &+ \mu(P|\Pi, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \mathcal{P}roc) \\ &- \mu(Q|\Pi, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathcal{P}roc) + \mu(Q|\Pi, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \mathcal{P}roc)| \leq \frac{1}{q(\mathbf{N})} \end{aligned}$$

D'où

$$|\text{Prob}[P|\Pi \rightsquigarrow_S o] - \text{Prob}[Q|\Pi \rightsquigarrow_S o]| \leq \frac{1}{q(\mathbf{N})}.$$

[Induction :] Supposons que $\max(d_H(P|\Pi, o, S), d_H(Q|\Pi, o, S) = n$, et que $\forall_{q \in Poly}$ nous avons

$$\sum_{k<n} \sum_{s \in Tr_k^{\mathcal{O}bs\setminus\{o\}}(P,o,S) \cup Tr_k^{\mathcal{O}bs\setminus\{o\}}(Q,o,S)} |\text{Prob}[P|\Pi \rightsquigarrow_S^{tr} s] - \text{Prob}[Q|\Pi \rightsquigarrow_S^{tr} s]| \leq \frac{1}{2q(\mathbf{N})}.$$

Maintenant, soit $U_k = Tr_k^H(P|\Pi, o, S) \cup Tr_k^H(Q|\Pi, o, S)$, alors nous avons

$$\begin{aligned} |\text{Prob}[P|\Pi \rightsquigarrow_S o] &- \text{Prob}[Q|\Pi \rightsquigarrow_S o]| \\ &= |\sum_{k \leq n} \sum_{s \in U_k} (\text{Prob}[P|\Pi \rightsquigarrow_S^{tr} s] - \text{Prob}[Q|\Pi \rightsquigarrow_S^{tr} s])| \\ &\leq |\sum_{k<n} \sum_{s \in U_k} \text{Prob}[P|\Pi \rightsquigarrow_S^{tr} s] - \text{Prob}[Q|\Pi \rightsquigarrow_S^{tr} s]| \\ &\quad + \sum_{s \in U_n} |\text{Prob}[P|\Pi \rightsquigarrow_S^{tr} s] - \text{Prob}[Q|\Pi \rightsquigarrow_S^{tr} s]| \end{aligned}$$

Par hypothèse d'induction nous avons

$$|\sum_{k<n} \sum_{s \in U_k} \text{Prob}[P|\Pi \rightsquigarrow_S^{tr} s] - \text{Prob}[Q|\Pi \rightsquigarrow_S^{tr} s]| \leq \frac{1}{2q(\mathbf{N})}.$$

Posons $r_n = |Tr_n^H(P|\Pi, o, S)| + |Tr_n^H(Q|\Pi, o, S)|$. $P \simeq^{tr} Q \Rightarrow \forall s \in U_n$

$$|\text{Prob}[P|\Pi \rightsquigarrow_S^{tr} s] - \text{Prob}[Q|\Pi \rightsquigarrow_S^{tr} s]| \leq \frac{1}{r_n \cdot 2q(\mathbf{N})},$$

alors

$$\sum_{s \in U_n} |\text{Prob}[P|\Pi \rightsquigarrow_S^{tr} s] - \text{Prob}[Q|\Pi \rightsquigarrow_S^{tr} s]| \leq \frac{r_n}{r_n \cdot 2q(\mathbf{N})} = \frac{1}{2q(\mathbf{N})}.$$

Nous avons ainsi

$$\begin{aligned} |\text{Prob}[P|\Pi \rightsquigarrow_S o] - \text{Prob}[P|\Pi \rightsquigarrow_S o]| &\leq \frac{1}{2q(\mathbf{N})} + \frac{1}{2q(\mathbf{N})} \\ &\leq \frac{1}{q(\mathbf{N})}. \end{aligned}$$

($\Leftarrow$). Supposons maintenant que $P \simeq Q$, mais $P \not\simeq^{tr} Q$. Alors, nous allons prouver qu'on aboutit à une contradiction. En effet, si $P \not\simeq^{tr} Q$, alors il existe un attaquant (Π, S), une trace observable $o_1 o_2 \cdots o_n$ et un polynome q t.q.

$$\forall_{\mathbf{N}} \; |\text{Prob}[P|\Pi \rightsquigarrow_S^{tr} o_1 o_2 \cdots o_n] - \text{Prob}[Q|\Pi \rightsquigarrow_S^{tr} o_1 o_2 \cdots o_n]| > \frac{1}{q(\mathbf{N})}.$$

Soit $o' = (c', m)$ t.q. c' n'est ni un canal P et Q ni un canal de Π et soit (Π', S') un attaquant qui se comporte exactement comme (Π, S) exepté que chaque fois qu'il observe une trace $o_1 o_2 \cdots o_n$ il produit l'observable o' avec probabilité 1. Alors nous avons

$$\begin{aligned} &|\text{Prob}[\; P|\Pi' \rightsquigarrow'_S o'] - \text{Prob}[Q|\Pi' \rightsquigarrow'_S o']| \\ &\quad = \quad |\text{Prob}[P|\Pi \rightsquigarrow_S^{tr} o_1 o_2 \cdots o_n o'] - \text{Prob}[Q|\Pi \rightsquigarrow_S^{tr} o_1 o_2 \cdots o_n o']| \end{aligned}$$

D'où $P \not\simeq Q$ qui contredit notre hypothèse. ■

4.4 Étude de cas : Anonymat dans le protocole du "diner des cryptographes"

Pour démontrer l'utilité de notre approche, nous allons terminer ce chapitre par une étude de cas : l'analyse du protocole du *Dîner des Cryptographes* (Chaum, 1988). C'est

un exemple classique de protocole assurant la propriété d'anonymat inconditionnel. Son autheur le décrit comme suit :

> *Trois cryptographes sont assis dans leur restaurant favori pour diner. Leur hôte les informe qu'un arrangement a été pris avec le maître d'hôtel pour que le diner soit payé de manière anonyme. Il se pourrait que l'un des cryptographes paie le diner, ou il pourrait être payé par la NSA (U.S. National Security Agency). Les trois cryptographes respectent le droit de chacun d'eux de payer de manière anonyme, mais se demandent si la NSA paie ou non.*

Pour résoudre leur problème, Chaum propose le protocole suivant : chaque cryptographe lance un jeton se trouvant entre lui et le cryptographe à sa droite, de telle sorte qu'ils soient les seuls à voir le résultat. Ensuite, chacun dit à haut voix si les deux pièces qu'il voit sont tombées sur la même face ou non. Un cryptographe payeur (s'il y'en a bien sûr !) dit le contraire de ce qu'il voit. L'idée est que si les jetons ne sont pas biaisés et que les cryptographes jouent correctement le protocole, alors un nombre impair de "différent" indique que l'un d'eux paie sans que les deux autres apprenent quoique soit sur l'idendité du payeur. Autrement, c'est la NSA qui paie.

4.4.1 Une spécification incorrecte du protocole

Dans la spécification[6] ci-dessous, nous supposons que la NSA choisit le payeur selon une distribution probabiliste (connue de lui seul !) définie par la fonction λ_{NSA} et informe chaque cryptographe à travers un canal sûr (privé) s'il est le payeur ou non. Pour assurer l'équité entre les cryptographes, chaque lancement de jetons est fait par une "tierce partie de confiance" grâce à la fonction *flips* et le résultat est rendu

[6] où $\oplus$ dénote l'addition modulo 2.

disponible simultanement aux deux cryptographes concernés.

$$
\begin{aligned}
NSA ::= \quad & [\lambda_{NSA}(3) \hookrightarrow x](\prod_{0\leq i\leq 3} [x=i]Payer_i) \\
Payer_3 ::= \quad & \overline{c}_0(nopay).\overline{c}_1(nopay).\overline{c}_2(nopay).\mathbf{0} \\
Payer_i ::= \quad & \overline{c}_i(pay).\overline{c}_{i\oplus 1}(nopay).\overline{c}_{i\oplus 2}(nopay).\mathbf{0} \text{ if } 0 \leq i \leq 2. \\
Crypts ::= \quad & [flips(coin_0) \hookrightarrow y_0][flips(coin_1) \hookrightarrow y_1][flips(coin_2) \hookrightarrow y_2] \prod_{0\leq i\leq 2} Crypt_i \\
Crypt_i ::= \quad & c_i(z_i).([z_i = pay]P_i|[z_i = nopay]Q_i) \\
P_i ::= \quad & [y_i = y_{i\oplus 1}]\overline{pub}_i(desagree).\mathbf{0}|[y_i \neq y_{i\oplus 1}]\overline{pub}_i(agree).\mathbf{0} \\
Q_i ::= \quad & [y_i = y_{i\oplus 1}]\overline{pub}_i(agree).\mathbf{0}|[y_i \neq y_{i\oplus 1}]\overline{pub}_i(desagree).\mathbf{0}
\end{aligned}
$$

Le protocole est alors spécifié comme suit : $DCP^1 ::= (NSA|Crypts)\backslash\{c_0, c_1, c_2\}$

4.4.2 Spécification de l'anonymat

Nous proposons une version probabiliste d'une spécification de la propriété d'anonymat due à Schneider et Sidiropoulos, 1996, dans le modèle possibiliste CSP. L'idée est que, étant donné deux ensembles A et O d'événements anonymes et observables respectivement, un protocole P assure l'anonymat d'événements A envers un observateur quelconque ne voyant que les événements O si P ne lui permet pas de déterminer toute dépendance causale entre les distributions probabilistes des événements A et O.

Soit $\text{Perm}(A)$, l'ensemble des permutations des éléments de A et $\pi \in \text{Perm}(A)$ une permutation. Nous dénotons par $\pi(P)$ le processus obtenu en remplaçant toute occurrence d'un événement a dans P par l'événement $\pi(a)$.

Définition 4.4.1 : [Propriété d'anonymat] P assure l'anonymat des événements A si et seulement si

$$\forall_{\pi\in Perm(A)} P \simeq^{tr} \pi(P)$$

Dans la spécification ci-dessus, les événements anonymes sont $A_{DCP} = \{(c_i, m) \mid i = 0, 1, 2 \text{ and } m = pay,\ nopay\}$ et les événements observables sont les communications à travers les canaux publics.

Soit

$$Sched_\tau = \{S \in Sched \mid \sum_{(\tau,i,j)\in A, i,j\in I_1} \text{Prob}[S(A, I_1, I_2) = (\tau, i, j)] = 1\}$$

le sous-ensemble d'ordonnanceurs donnant priorité aux actions internes du protocole et $\simeq_\tau^{tr}$ l'équivalence de trace induite par $Sched_\tau$. Alors, nous avons les résultats suivants qui montrent que *Sched* détecte la faille dans DCP^1 alors que $Sched_\tau$ ne peut pas la détecter.

Théorème 4.4.1

- $\forall_{\pi\in Perm(A_{DCP^1})}$, *si* $\forall_{i=0,1,2}$ $\text{Prob}[flips(coin_i) = Head] = \frac{1}{2}$ *alors* $DCP^1 \simeq_\tau^{tr} \pi(DCP^1)$.
- *Quelle que soit la distribution probabiliste des jetons, si* π *n'est pas la permutation identité alors* $DCP^1 \not\simeq^{tr} \pi(DCP^1)$.

La faille DCP^1 est due au fait que le cryptographe qui paie a un certain avantage sur les deux autres puisqu'il est le premier à être informé par la NSA. De plus, cette spécification est faite de telle sorte que chaque cryptographe puisse automatiquement émettre son message dès qu'il est informé de son statut de payeur ou non indépendamment des autres. Ainsi, un ordonnanceur donnant la priorité aux actions observables va seulement générer des traces observables commençant par le message du cryptographe payeur (s'il en existe un bien sûr !). Puisque si π n'est pas la perputation identité, alors elle change l'idendité du payeur. Avec de tels ordonnanceurs, DCP^1 et $\pi(DCP^1)$ ne peuvent que générer des traces observables différentes. Ces deux processus ne sont donc pas équivalents. Une spécification corrigeant cette faille est donnée ci-dessous. Elle bloque simplement le lancement des pièces jusqu'à ce que tous les

cryptographes reçoivent leurs messages de la part de la NSA.

$$
\begin{array}{ll}
NSA ::= & [\lambda_{NSA}(3) \hookrightarrow x](\prod_{0\leq i\leq 3}[x=i]Payer_i) \\
Payer_3 ::= & \overline{c}_0(nopay).\overline{c}_1(nopay).\overline{c}_2(nopay).\mathbf{0} \\
Payer_i ::= & \overline{c}_i(pay).\overline{c}_{i\oplus 1}(nopay).\overline{c}_{i\oplus 2}(nopay).\mathbf{0} \text{ if } 0\leq i\leq 2. \\
Crypts2 ::= & c_0(z_0).c_1(z_1).c_2(z_2).Flip \\
Flip ::= & [flips(coin_0)\hookrightarrow y_0][flips(coin_1)\hookrightarrow y_1][flips(coin_2)\hookrightarrow y_2](\prod_{0\leq i\leq 2} Crypt_i) \\
Crypt_i ::= & ([z_i=pay]P_i|[z_i=nopay]Q_i) \\
P_i ::= & [y_i=y_{i\oplus 1}]\overline{pub}_i(desagree).\mathbf{0}|[y_i\neq y_{i\oplus 1}]\overline{pub}_i(agree).\mathbf{0} \\
Q_i ::= & [y_i=y_{i\oplus 1}]\overline{pub}_i(agree).\mathbf{0}|[y_i\neq y_{i\oplus 1}]\overline{pub}_i(desagree).\mathbf{0}
\end{array}
$$

Le protocole est alors spécifié par : $DCP^2 ::= (NSA|Crypts2)\backslash\{c_0,c_1,c_2\}$ et nous avons cet important résultat très classique du DCP qui stipule que si les pièces ne sont pas biaisées, alors DCP assure l'anonymat du cryptographe payeur.

Théorème 4.4.2 *Si* $\forall_{0\leq i\leq 2}\ \text{Prob}[flips(coin_i)=Head]=\frac{1}{2}$ *alors* $\forall_{\pi\in Perm(A_{DCP^2})}$ $DCP^2\simeq^{tr}\pi(DCP^2)$

4.4.3 Implémentation du protocole par des canaux chiffrés

Supposons que la NSA ne veut pas informer les cryptographes de sa décision avant qu'ils n'aillent diner. La raison pourrait être qu'un payeur averti à l'avance pourrait manquer le rendez-vous pour éviter de payer. Nous proposons une solution qui permet à la NSA d'implémenter les canaux privés par des canaux chiffrés, i.e. elle chiffre ses choix en utilisant des clés secrètes communes avant de les envoyer à travers des canaux publics, comme indiqué par la spécification donnée dans le Tableau 4.4, où $Kgen$, enc

TABLEAU 4.4 *Spécification du protocole* DCP *par des canaux chiffrés*

NSA :

$$
\begin{aligned}
NSA ::= \;& [Kgen(\mathbf{N}) \hookrightarrow k_1][Kgen(\mathbf{N}) \hookrightarrow k_2][Kgen(\mathbf{N}) \hookrightarrow k_3]P \\
P ::= \;& \overline{priv}_1(k_1).\overline{priv}_2(k_2).\overline{priv}_3(k_3).Payers \\
Payers ::= \;& [\lambda_{NSA}(3) \hookrightarrow x](\prod_{0\leq i\leq 3}[x=i]Payer_i) \\
Payer_0 ::= \;& [enc(nopay,k_1) \hookrightarrow e_1]\overline{c}_1(e_1).[enc(nopay,k_2) \hookrightarrow e_2]\overline{c}_2(e_2). \\
& [enc(nopay,k_3) \hookrightarrow e_3]\overline{c}_3(e_3).\mathbf{0} \\
Payer_1 ::= \;& [enc(pay,k_1) \hookrightarrow e_1]\overline{c}_1(e_1).[enc(nopay,k_2) \hookrightarrow e_2]\overline{c}_2(e_2). \\
& [enc(nopay,k_3) \hookrightarrow e_3]\overline{c}_3(e_3).\mathbf{0} \\
Payer_2 ::= \;& [enc(nopay,k_1) \hookrightarrow e_1]\overline{c}_1(e_1).[enc(pay,k_2) \hookrightarrow e_2]\overline{c}_2(e_2). \\
& [enc(nopay,k_3) \hookrightarrow e_3]\overline{c}_3(e_3).\mathbf{0} \\
Payer_3 ::= \;& [enc(nopay,k_1) \hookrightarrow e_1]\overline{c}_1(e_1).[enc(nopay,k_2) \hookrightarrow e_2]\overline{c}_2(e_2). \\
& [enc(pay,k_3) \hookrightarrow e_3]\overline{c}_3(e_3).\mathbf{0}
\end{aligned}
$$

Cryptographes :

$$
\begin{aligned}
Crypts ::= \;& priv_1(key_1).priv_2(key_2).priv_3(key_3).Flip \\
Flip ::= \;& [flips(coin_1) \hookrightarrow y_1][flips(coin_2) \hookrightarrow y_2][flips(coin_3) \hookrightarrow y_3](\prod_{i\leq 3} Crypt_i) \\
Crypt_i ::= \;& priv_i(key_i).c_i(z_i).[dec(z_i,key_i) \hookrightarrow m_i]([m_i = pay]P_i|[m_i = nopay]Q_i) \\
P_i ::= \;& [y_i = y_{i\oplus 1}]\overline{pub}_i(desagree).\mathbf{0}|[y_i \neq y_{i\oplus 1}]\overline{pub}_i(agree).\mathbf{0} \\
Q_i ::= \;& [y_i = y_{i\oplus 1}]\overline{pub}_i(agree).\mathbf{0}|[y_i \neq y_{i\oplus 1}]\overline{pub}_i(desagree).\mathbf{0}
\end{aligned}
$$

Protocole DCP :

$$DCP ::= (NSA|Crypts)\backslash\{priv_1, priv_2, priv_3\}$$

et *dec* désignent respectivement un algorithme de génération des clés, le chiffrement à clé symétrique et l'algorithme de déchiffrage correspondant.

Avec une telle implémentation, nous pouvons monter une attaque qui brise la propriété d'anonymat du protocole en utilisant les propriétiés algébriques du crypto-système. En effet, supposons que l'algorithme de chiffrage utilisé est le chiffrement de Vernam à clé jetable le "One-time Pad" (OTP en abrégé), i.e. $enc(m,k) = XOR(m,k)$. Puisque l'OTP est sensible à la longueur des messages et que l'ensemble des messages secrets est $\{pay, nopay\}$, alors nous supposons que *pay* et *nopay* sont implémentés respectivement par les valeurs booléennes $True$ et $False$. Nous avons alors le résultat

suivant :

Théorème 4.4.3 *Quelle que soit la distribution probabiliste des pièces, le* DCP *n'assure pas l'anonymat du payeur.*

Preuve: Soit $k = XOR(True, False)$ et l'attaquant (Π, S) défini par $\Pi = c_i(e).[enc(e,k) \hookrightarrow e']\overline{c_i}(e').\mathbf{0}$ et S un ordonnanceur quelconque. Essentiellement, l'attaquant essaie d'inverser les messages *pay* et *nopay* du cryptographe i, puisque

$$XOR(XOR(k_i, True), XOR(True, False)) = XOR(k_i, False)$$

et inversement,

$$XOR(XOR(k_i, False), XOR(True, False)) = XOR(k_i, True).$$

Nous avons les cas suivants :

- Si le cryptographe attaqué est le payeur, alors le résultat de l'attaque est que la NSA paie, puisque le message *pay* du payeur est inversé à *nopay* et ainsi, tous les trois reçoivent des messages *nopay*. Ainsi, le résultat lui permet de déduire que le véritable payeur est le cryptographe qu'il a attaqué.
- Si c'est la NSA qui paie, alors le résultat de l'attaque est que le protocole se termine normalement et un des cryptographes doit payer. En effet, le cryptographe attaqué reçoit le message *pay* au lieu de *nopay*. Par conséquent, l'attaquant sait que le véritable payeur devrait être la NSA, mais son attaque fait que le cryptographe attaqué va aussi payer. Ce cas est le plus dangereux, puisque si l'attaquant se trouve être le maître d'hôtel (la NSA pourrait lui indiquer d'avance si elle paye ou si c'est l'un des cryptographes qui doit payer sans lui révéler son identité !), alors l'attaque lui permet de se faire payer deux fois, i.e. par la NSA et le cryptographe attaqué. Ainsi, si les cryptographes ne recommuniquent pas avec la NSA (par exemple pour

la remercier d'avoir payé leur facture), alors l'arnaque ne sera jamais détectée.

- Enfin, si ni le cryptographe attaqué ni la NSA ne paient, alors la conséquence de cette attaque est qu'il y aura deux payeurs : le véritable payeur choisi par la NSA et le cryptographe attaqué. Ainsi, le protocole doit se terminer par un échec. Au moins l'attaquant sait que l'un des cryptographes doit payer et que ce n'est pas celui qui est attaqué. Et comme l'attaque peut être interne, i.e. un des cryptographes collabore avec l'intrus (qui ne l'attaque pas bien sûr !), l'attaque permet à un tel cryptographe de déterminer l'identité réelle du payeur : c'est l'autre cryptographe non attaqué.

■

CHAPITRE 5

SÉMANTIQUE CONTEXTUELLE DU MODÈLE PROSPA

5.1 Introduction

Notre modèle ProSPA, ainsi que tous les modèles formels probabilistes pour la vérification des protocoles cryptographiques proposés à ce jour, se heurte à l'obligation de quantifier sur tous les attaquants possibles. De plus, ces modèles se heurtent également au problème des faux attaquants, i.e. des attaquants qui connaissent par magie les données secrètes du système avant de l'attaquer. En effet, considérons à nouveau le protocole d'authentification par mot de passe

$$(\overline{c}(pw)|c(x).c'(y)[y = x].\overline{c'}(ok).\mathbf{0})\backslash\{c\}.$$

Puisque dans ces modèles un attaquant est un processus quelconque muni d'une stratégie d'attaque, i.e. un ordonnnanceur, alors l'attaquant $\Pi = \overline{c'}(pw).\mathbf{0}$, qui transmet le bon mot de passe serait admissible. Aussi, s'inspirant des techniques déjà utilisées pour notre modèle possibiliste CSPAD, nous proposons dans ce chapitre, une sémantique contextuelle de notre modèle probabiliste.

5.1.1 Notre contribution

Pour contourner ces difficultés, nous avons séparé le modèle des systèmes de celui de l'attaquant. En effet, comme dans le modèle CSPAD, les protocoles sont modélisés par des processus, alors que l'attaquant est implicitement représenté par l'ensemble de ses connaissances plus un système déductif probabiliste polynomial et un ensemble des

fonctions probabilistes polynomiales contraignates, dénommées *fonctions de sélection.* Ces fonctions de sélection permettent à l'attaquant de pouvoir fixer un domaine de recherche "valide" pour tout canal d'écoute conformément à sa base de connaissance. En effet, si on considère, par exemple, le protocole d'authentification par mot de passe ci-dessus, l'attaquant peut fixer un domaine de recherche $D \subset \mathcal{M}$ pour le canal d'input $c'(y)$, i.e. pour le mot de passe. Si le mot de passe est généré de manière totalement aléatoire, alors la probabilité qu'une fonction de sélection F retoure un domaine de recherche contenant le bon mot de passe, doit être de l'ordre de $\frac{1}{2^n}$, où n désigne la longueur de la bande passante du canal c'. Si $\mathcal{F}$ dénote l'ensemble des fonctions de sélection et $\mathcal{P}oly$ l'ensemble des polynômes positifs, cette contrainte est formellement établie comme suit :

$$\forall_{F \in \mathcal{F}, q \in \mathcal{P}oly, \phi \subset \mathcal{M}} \text{ t.q. } pw \notin \phi \ \text{Prob}[pw \in F(c', \phi)] \leq \frac{1}{q(n)}.$$

Cette façon de faire, en plus de nous permettre d'éviter de quantifier sur tous les attaquants possibles, nous simplifie énormement la tâche au moment de définir les classes d'actions équiprobables et l'ordonnancement.

La sémantique contextuelle ainsi obtenue engendre des systèmes de transitions de la forme : les états sont des *configurations* $\phi \rhd P$ où P est un processus et ϕ est la connaissance actuelle de l'attaquant et les transitions $\phi \rhd P \xrightarrow{\alpha[p]} \psi \rhd Q$ qui signifient que le processus P interagissant avec le contexte de connaissance ϕ peut faire l'action α avec la probabilité p et évoluer comme Q interagissant avec un contexte de connaissance ψ.

En vue de vérifier des propriétés de sécurité, nous avons défini les versions contextuelles de nos équivalences asymptotiques du chapitre précédent. Au contaire de la sémantique concrète, elles ne sont plus équivalentes : l'équivalence de trace étant plus fine. Nous avons également défini une bisimulation faible contextuelle et prouvé

qu'elle raffine les deux équivalences asymptotiques.

Enfin, pour démontrer l'utilité de notre approche, nous avons analysé un important protocole de sécurité : le "Crowds protocol" (Reiter et Rubin, 1998) développé par Reiter et Ruben. C'est un protocole qui vise à assurer l'anonymat dans les transactions sur le Web. Pour assurer l'anonymat, ce protocole masque les communications de chaque usager en les faisant acheminer par un usager choisi de manière aléatoire à l'intérieur d'un groupe d'usagers similaires. Ainsi, même si un usager indiscret ou un membre malicieux du groupe observe un message envoyé par un usager particulier, il ne peux jamais être certain si l'usager est bien celui qui a envoyé le message ou s'il achemine simplement le message d'un autre usager. Cette analyse nous a permis de donner une nouvelle caractérisation de l'*innocence probable* en termes de probabilités cumulatives et de déduire, grâce au système de transitions engendré par la sémantique du protocole, un important théorème, dû à Reiter et Ruben, sur une relation que doivent vérifier les paramètres du protocole pour que l'innocence probable soit assurée.

5.1.2 Organisation du chapitre

Le reste du chapitre se présente comme suit : la sémantique contextuelle est donnée dans la section 5.2. Les propriétés de sécurité sont exprimées en termes d'équivalences asymptotiques présentées dans la section 5.3 et la bisimulation faible dans la section 5.4. Notre étude de cas est présentée dans la section 5.5.

5.2 Sémantique contextuelle

Nous rappellons le lecteur que la syntaxe et la sémantique concrète du modèle sont décrites dans la Section 4.2 du chapitre précedent. Comme dans le cas possibiliste, l'attaquant peut être modélisé par l'ensemble des connaissances, dénoté par ϕ, qu'il

acquiert lors de l'exécution du protocole plus un système déductif illustré par le Tableau 5.1 qui lui permet d'inférer des messages à partir de cette base de connaissances. Mais comme dans la vie réelle, l'environnement dans lequel évolue le protocole est totalement sous le contrôle de l'attaquant, ce dernier peut donc affecter une distribution probabiliste aux canaux publics sous son contrôle. Nous le modélisons en plaçant les canaux publics sous le contrôle d'un ordonnanceur externe S décrit plus loin. Ainsi, l'attaquant est formé par la paire (ϕ, S).

TABLEAU 5.1 *Règles d'inférence des messages* ProSPA.

$$\frac{m \in \phi}{\phi \vdash_1 m} \qquad \frac{\phi \vdash_{p_1} m_1 \quad \phi \vdash_{p_2} m_2}{\phi \vdash_{p_1 \cdot p_2} (m_1, m_2)}$$

$$\frac{\phi \vdash_{p_i} m_i \; (1 \leq i \leq k) \quad \lambda(m_1, \cdots, m_k) \overset{q}{\hookrightarrow} m}{\phi \vdash_{q \prod_1^k p_i} m}$$

5.2.1 Système déductif

Le système déductif probabiliste du Tableau 5.1 permet à l'intrus d'inférer des messages à partir de sa base de connaissances ϕ. La notation $\phi \vdash_p m$ signifie que m est déductible de ϕ avec la probabilité p. Intuitivement, l'attaquant peut inférer, avec probabilité certaine, tout message qui appartient à sa base de connaissances. De même, s'il peut inférer deux messages m_1 et m_2 avec les probabilités respectives p_1 et p_2, alors il peut inférer le couple (m_1, m_2) avec la probabilité $p_1 p_2$. Enfin, si λ est une fonction probabiliste polynomiale d'arité k et $\lambda(m_1, \cdots, m_k) \overset{p}{\hookrightarrow} m$, alors m est déductible de ϕ si tous les m_i le sont.

Définition 5.2.1 : *Soit ϕ un ensemble. Un message m est déductible de ϕ s'il peut être engendré par les règles d'inférence du Tableau 5.1. Nous dénotons par $\mathcal{D}(\phi)$ l'ensemble des messages déductibles de ϕ.*

Toute dérivation $\phi \vdash_p m$ n'est pas forcément effectué en temps polynomial, puisque rien n'empêche l'arbre de dérivation d'être exponentiel (en $\mathbf{N}$). Or, notre modèle d'attaquant doit être probabiliste polynomial. D'où l'intérêt de ne considérer que les dérivations qui se font en temps polynomial, dénoté par $\mathcal{D}_{poly(\mathbf{N})}(\phi)$.

Proposition 5.2.1 *Soient m, un message et ϕ un ensemble de messages. Une dérivation $\phi \vdash_p m$ appartient à $\mathcal{D}_{poly(\mathbf{N})}(\phi)$ si et seulement s'il existe un arbre de dérivation de $\phi \vdash_p m$ dont le nombre de noeuds est polynomial en $\mathbf{N}$.*

Preuve: Soit $\phi \vdash_p m$ une dérivation. Il est clair que si $\phi \vdash_p m \in \mathcal{D}_{poly(\mathbf{N})}(\phi)$, alors il existe un arbre de dérivation de $\phi \vdash_p m$ dont le nombre de noeuds, dénoté par nodes($\phi \vdash_p m$), est polynomial en $\mathbf{N}$. En effet, si nodes($\phi \vdash_p m$) n'est pas polynomial, alors le temps de dérivation de m ne le serait pas non plus car chaque étape de dérivation se fait en temps polynomial et il y aurait un nombre exponentiel d'étapes (ie de noeuds).
Supposons maintenant que nodes($\phi \vdash_p m$) est polynomial en $\mathbf{N}$. Puisque chaque étape se fait en temps polynomial, alors $\forall i \in [1..\text{nodes}(m, \phi)(\mathbf{N})]$, il existe un polynôme q_i tel que le temps de la i^{eme} étape de dérivation soit majoré par $q_i(\mathbf{N})$. Soit n le plus grand degré apparaissant dans tous les polynômes q_i. Pour chaque polynôme q_i, posons $q_i = \sum_{j \leq n} a_{ij} x^j$ où a_{ij} est le coefficient du monôme x^j si ce dernier apparaît dans q_i et 0 autrement. Soient $b_j = max\{a_{ij} \mid i \in [1..\text{nodes}(\phi \vdash_p m)(\mathbf{N})]\}$ et q le polynôme défini par $q = \sum_{j \leq n} b_j x^j$. Il est clair que le polynôme q majore les polynômes q_i et le temps de dérivation $\phi \vdash_p m$ est majoré par nodes($\phi \vdash_p m$)($\mathbf{N}$) $\times$ $q(\mathbf{N})$, qui est un polynôme en $\mathbf{N}$ puisque c'est le produit de deux polynômes. ■

La sémantique contextuelle de ProSPA est donnée à la Table 5.2. Si un processus P peut faire une action α pour atteindre le processus P', alors la probabilité totale de cette action est calculée comme étant la probabilité que P fasse l'action α et atteint P' sur le nombre total des façons différentes dont le processus peut faire

TABLEAU 5.2 *Sémantique contextuelle de* ProSPA.

$$\texttt{Output}\ \frac{P \xrightarrow{\overline{c}(m)[p]} P'}{\phi \rhd P \xrightarrow{\overline{c}(m)[\frac{p}{\overline{\Upsilon}(P,\overline{c}(m))}]} \phi \cup \{m\} \rhd P'} \qquad \texttt{Input}\ \frac{P \xrightarrow{c(m)[p]} P' \quad \phi \vdash_q m \in \mathcal{D}_{poly(\mathbf{N})}(\phi)}{\phi \rhd P \xrightarrow{c(m)[\frac{q \cdot p}{\overline{\Upsilon}(P,c(m))}]} \phi \rhd P'}$$

$$\texttt{Tau}\ \frac{P \xrightarrow{\tau[p]} P'}{\phi \rhd P \xrightarrow{\tau[\frac{p}{\Upsilon(P,\tau)}]} \phi \rhd P'} \qquad \texttt{Comm}\ \frac{P \xrightarrow{\alpha \cdot \overline{\alpha}[p]} P'}{\phi \rhd P \xrightarrow{\alpha \cdot \overline{\alpha}[\frac{p}{\Upsilon(P,\alpha \cdot \overline{\alpha})}]} \phi \rhd P'}$$

l'action α, qui est compté par la fonction de normalisation Υ définie ci-dessous. La fonction de normalisation nous permet de nous assurer que les probabilités de distribution sont bien comptées quand on évalue un processus dans un contexte donné, en particulier lorsqu'on compose plusieurs processus. En effet, lorsqu'on évalue un processus, l'ordonnanceur que nous expliquons plus loin en détail, choisit une action parmi toutes les actions disponibles et cette dernière est exécutée. L'ordonnanceur est certes contrôlé par l'attaquant, mais ce dernier ne doit pas contrôler la réaction interne du processus. Par exemple, l'attaquant peut décider d'envoyer le message m sur le canal c, si le processus peut lire sur ce canal en plusieurs endroits, alors chacun des inputs sur ce canal a la même probabilité de réagir à la requête de l'attaquant. Ainsi, chaque action est choisie uniformément parmi toutes les actions exécutables de même type. D'où l'importance de savoir le nombre exact de toute action exécutable à ce stade d'évaluation. A noter que dans le cas d'un output $\overline{c}(m)$ du protocole, nous normalisons la probabilité sur tous les outputs du protocole sur le canal c grâce à la fonction $\overline{\Upsilon}$, avec

$$\overline{\Upsilon}(P,\alpha) = \begin{cases} \sum_{m' \in \mathrm{dom(c)}} \Upsilon(P, \overline{c}(m')) & \text{si } \alpha \in \{\overline{c}(m) \mid c \in \mathcal{C} \text{ et } m \in \mathcal{M}\}. \\ \Upsilon(P,\alpha) & \text{sinon.} \end{cases}$$

Enfin, tout message envoyé par le protocole et intercepté par l'attaquant augmente la connaissance de l'attaquant. Tout message reçu de ce dernier doit être déductible de sa base de connaissance ϕ. Toute communication directe, observable ou privée, entre

deux composantes du protocole ne modifie pas sa base de connaissances.

Définition 5.2.2 : *La fonction normalisation* $\Upsilon : \mathcal{P}roc \times \mathcal{A}ct \longrightarrow \mathbb{N}$ *est définie récursivement comme suit :*

$$\textit{Si } P \notin \mathcal{B}locked \textit{ alors } \Upsilon(P,\alpha) = \begin{cases} 1 & \textit{si } \alpha = \tau \\ 0 & \textit{sinon} \end{cases}$$

sinon :

$\Upsilon(\mathbf{0},\alpha) = 0$

$$\Upsilon(\beta.P,\alpha) = \begin{cases} 1 & \textit{si } \alpha = \beta \\ 0 & \textit{sinon} \end{cases}$$

$$\Upsilon(P\backslash\Gamma,\alpha) = \begin{cases} \Upsilon(P,\alpha) & \textit{si } \alpha \notin \Gamma \\ 0 & \textit{sinon} \end{cases}$$

$$\Upsilon(P\backslash C,\alpha) = \begin{cases} \Upsilon(P,\alpha) & \textit{si } \mathrm{support}(\alpha) \notin C \textit{ et } \alpha \neq \tau \\ \Upsilon(P,\alpha) + \sum_{\{\beta \in \mathcal{P}artial \mid \mathrm{support}(\beta) \in C\}} \Upsilon(P,\beta\cdot\overline{\beta}) & \textit{si } \alpha = \tau \\ 0 & \textit{sinon} \end{cases}$$

$$\Upsilon(P|Q,\alpha) = \begin{cases} \Upsilon(P,\alpha) + \Upsilon(Q,\alpha) & \textit{si } \alpha \in (\mathcal{P}artial \cup \{\tau\}) \\ \Upsilon(P,\alpha) + \Upsilon(Q,\alpha) + \Upsilon(P,\beta) \times \Upsilon(Q,\overline{\beta}) & \textit{si } \alpha = \beta\cdot\overline{\beta} \end{cases}$$

Notons que, ici également, malgré la normalisation des probabilités, les systèmes de transitions engendrés par la sémantique contextuelle (Tableau 5.2) des processus ProSPA ne sont pas purement probabilistes, puisque la somme des probabilités des transitions sortantes d'un sommet peut être supérieure à 1. En effet, considérons les processus $P = \overline{c_1}(a)|\overline{c_2}(b)$ dans l'environnement de connaissance ϕ dont le sytème de transitions est illustré par la Figure 5.1 où $\alpha = \overline{c_1}(a)$ et $\beta = \overline{c_2}(b)$. On voit clairement que la somme des probabilités des transitions sortantes du sommet initial est égale à 2. Pour obtenir des systèmes de transitions purement probabilistes, il

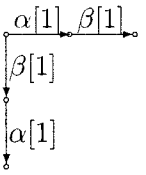

FIGURE 5.1 *Système de transitions probabiliste de la configuration* $\phi \rhd P = \phi \rhd \overline{c_1}(a).\mathbf{0}|\overline{c_2}(b).\mathbf{0}$

est donc essentiel d'ordonnancer les différentes actions disponibles à chaque étape d'évaluation du processus.

5.2.2 Actions stratégiquement équiprobables

Comme dans le cas concret, nous allons partitionner l'ensemble d'actions $\mathcal{A}ct$ en classes d'actions *stratégiquement équiprobables*. Nous rappellons que par stratégiquement équiprobables, nous voulons dire les actions qui doivent être choisies de manière uniforme étant donnée une stratégie S et que notre approche est guidée par les principes suivants :

1. Aucune stratégie ne doit distinguer les actions internes d'un protocole. En effet, les actions internes d'un protocole étant invisibles à l'intrus, si la stratégie consiste à laisser évoluer le protocole intérieurement (si c'est possible évidemment), alors si le protocole possède plusieurs actions internes, l'une de ces actions est choisie uniformement et exécutée.
2. Aucune stratégie ne doit permettre à l'intrus d'avoir un contrôle sur la réaction interne du protocole à ses requêtes. En effet, si l'intrus décide, par exemple, d'envoyer un message sur un canal public de lecture du protocole et que le protocole peut le lire à plusieurs positions, alors toutes ces positions doivent avoir la même chance de réagir à la requête de l'attaquant.

3. Quelle que soit la stratégie choisie, l'intrus doit avoir un contrôle total sur ses propres actions.

Définition 5.2.3 : *La relation de stratégies équiprobables $\sim$ sur l'ensemble $\mathcal{A}ct$ est définie comme suit :*

$$[\alpha]_\sim = \begin{cases} \{\tau\} & si\ \alpha = \tau \\ \{c(m)\} & si\ \alpha = c(m) \\ \{\overline{c}(m') \mid m' \in \mathrm{dom}(c)\} & si\ \alpha = \overline{c}(m) \\ \{\beta \cdot \overline{\beta} \mid \mathrm{support}(\beta) = \mathrm{support}(\alpha)\} & si\ \alpha = \alpha' \cdot \overline{\alpha'} \end{cases}$$

où $[\alpha]_\sim$ dénote la classe d'équivalence de l'action α.

Cette équivalence dit simplement que

1. Toutes les actions internes exécutables d'un protocole dans un état sont indistinguables.
2. Tous les inputs d'un message m sur le même canal sont équiprobables.
3. Tous les outputs sur un canal sont équiprobables. Donc en particulier si un protocole peut, dans un état donné, émettre deux messages différents, alors l'attaquant ne peut pas le forcer à émettre l'un des messages au détriment de l'autre.
4. Toutes les communications, observables sur un canal donné, sont équiprobables. En d'autres termes, un attaquant qui observe une communication sur un canal-public, ne peut pas à priori imposer quel message doit être communiqué sur ce canal.

5.2.3 L'ordonnanceur externe

Maintenant, nous définissons formellement l'ordonnanceur externe. N'oublions pas que ce dernier est contrôlé par l'attaquant et nous devons faire très attention pour que sa définition reflète la capacité réelle de nuisance de l'intrus. Rappelons que conformément à la sémantique contextuelle (cf. Table 5.2), tout message reçu de l'environnement (règle `Input`) doit être déductible de sa base de connaissance ϕ. Mais, comme le paramètre de sécurité $\mathbf{N}$ est public, alors toute base de connaissance ϕ d'un attaquant contient $\mathbf{N}$. De plus, comme toute fonction constante, i.e. une fonction de la forme

$$\forall x \ \lambda_m(x) = m,$$

est une fonction probabiliste polynomiale, alors tout message de longueur polynomiale est déductible en temps constant de ϕ. En effet, $\mathbf{N} \in \phi \Rightarrow \phi \vdash_1 \mathbf{N}$ et comme $\lambda_m(\mathbf{N}) = m$ (i.e. $\lambda_m(\mathbf{N}) \xhookrightarrow{1} m$), on a bien $\phi \vdash_1 m$. Ainsi, étant donné un canal d'écoute $c(x)$ d'un protocole, ces fonctions permettent à l'intrus de générer et d'envoyer, en temps constant, tout message appartenant au domaine de ce canal. En d'autres termes, si un protocole attend un message secret (par exemple un mot de passe), l'attaquant peut par magie choisir la bonne fonction constante, i.e. celle qui retourne le bon secret et ainsi réussir son attaque. Pour éliminer ces faux attaquants, il ne suffit pas, comme dans le cas possibiliste, d'imposer que le secret n'appartient pas initialement à ϕ. Aussi, nous allons munir l'attaquant d'un ensemble de fonctions probabilistes polynomiales contraignantes de la forme $F : 2^{\mathcal{M}} \times \mathcal{C} \longrightarrow 2^{\mathcal{M}}$, appelées *fonctions de sélection*. Étant donné l'ensemble des connaissances actuelles ϕ de l'attaquant et un canal public c, une fonction de sélection F choisit un domaine de recherche valide pour ce canal conformément à ϕ. Soit $\mathcal{C}onf$ l'ensemble des configurations de la forme $\phi \rhd P$ et $\mathcal{F}$ l'ensemble des fonctions de sélection. Nous définissons la fonction de localisation $\chi : \mathcal{C}onf \times \mathcal{F} \longrightarrow 2^{\mathcal{A}ct}$ conformément à la nouvelle sémantique. Elle prend une configuration $\phi \rhd P$ et une fonction de sélection F et retourne l'ensemble des

actions exécutables de $\phi \rhd P$ conformément aux contraintes imposées par F.

Définition 5.2.4 : *La fonction* $\chi : \mathcal{C}onf \times \mathcal{F} \rightarrow 2^{\mathcal{A}ct}$ *est définie récursivement par*
Si $P \notin \mathcal{B}locked$ *alors* $\chi(\phi \rhd P, F) = \{\tau\}$
sinon :

$$
\begin{aligned}
\chi(\phi \rhd \mathbf{0}, F) &= \emptyset \\
\chi(\phi \rhd \overline{c}(m).P, F) &= \{\overline{c}(m)\} \\
\chi(\phi \rhd c(x).P, F) &= F(\phi, c) \\
\chi(\phi \rhd P|Q, F) &= \chi(\phi \rhd P, F) \cup \chi(\phi \rhd Q, F) \\
&\quad \cup \{\alpha \cdot \overline{\alpha} \mid \alpha \in \chi(\phi \rhd P, F) \text{ et } \overline{\alpha} \in \chi(\phi \rhd Q, F)\} \\
\chi(\phi \rhd (P \backslash \Gamma), F) &= \chi(\phi \rhd P, F) \setminus \Gamma \\
\chi(\phi \rhd (P \backslash C), F) &= \begin{cases} (\chi(\phi \rhd P, F) \cup \{\tau\}) \setminus \{\alpha \in \mathcal{V}is \mid \text{support}(\alpha) \in C\} \\ \qquad si\ \exists_{\alpha \in \mathcal{P}artial}\ t.q.\ \text{support}(\alpha) \in C \text{ et } \alpha \cdot \overline{\alpha} \in \chi(\phi \rhd P, F) \\ \chi(\phi \rhd P, F) \setminus \{\alpha \in \mathcal{V}is \mid \text{support}(\alpha) \in C\} \\ \qquad sinon \end{cases}
\end{aligned}
$$

Nous sommes maintenant suffisament outillés pour définir formellement l'ordonnanceur externe :

Définition 5.2.5 : *Un* ordonnanceur externe S *est une paire de fonctions probabilistes polynomiales* (F, ζ) *où* $F \in \mathcal{F}$ *est une fonction de sélection et* $\zeta : 2^{\mathcal{A}ct} \rightarrow \mathcal{A}ct$, *une fonction probabiliste polynomiale stochastique vérifiant les conditions suivantes :*

1. $\forall_{A \subseteq \mathcal{A}ct}\ \tau \in A \Longrightarrow \text{Prob}[\zeta(A) = \tau] \in \{0, 1\}$.
2. $\forall_{\alpha,\beta \in A}\ \alpha \sim \beta \Longrightarrow \text{Prob}[\zeta(A) = \alpha] = \text{Prob}[\zeta(A) = \beta]$

Étant donné une configuration $\phi \rhd P$ (i.e. un état du système de transitions) et un ordonnanceur (F, ζ), le choix de la prochaine transition se déroule comme suit : la localisation $\chi(\phi \rhd P, F) = A$ retourne l'ensemble des actions exécutables de l'état $\phi \rhd$

P conformément aux contraintes imposées par F. Ensuite, la fonction ζ attribue une distribution probabiliste aux actions exécutables, i.e. elle choisit (avec une certaine probabilité) une action α dans A. Finalement, une des actions α disponibles est uniformément (cf. la normalisation) choisie et exécutée.

Lemme 5.2.1 *Soit $\phi \rhd P$ une configuration. Pour tout $(F, \zeta) \in Sched$, $A = \chi(\phi \rhd P, F) \neq \emptyset \Longrightarrow \exists \alpha \in A$ t.q.* $\text{Prob}[S(A) = \alpha] \neq 0$.

La preuve découle de la condition de stochasticité qui est une condition de progression en ce sens que, à chaque étape d'évaluation, quelque chose de positif doit se passer, i.e. une des actions disponibles va être exécutée. C'est une propriété importante en ce sens que l'ordonnancement n'introduit pas des deadlocks qui n'ont pas lieu d'être.

Lemme 5.2.2 *Soit $\phi \rhd P$ une configuration, $A = \chi(\phi \rhd P, F)$,* $\text{Out}(A)$ *et* $\text{Comm}(A)$ *respectivement les ensembles d'actions d'output partiels et de communications publiques dans A. Soit $\phi \rhd P \xrightarrow{\alpha[p]} \psi \rhd Q$ une transition sortante de $\phi \rhd P$ et q la probabilité que $\phi \rhd P$ exécute α et atteint $\psi \rhd Q$ selon l'ordonnanceur (F, ζ) dénotée $\phi \rhd P \xrightarrow{\alpha[q]}_S \psi \rhd Q$. Nous avons alors :*

1. $\alpha = \tau$ *alors* $q = p \times \text{Prob}[S(A) = \tau] = \begin{cases} p & si\ \text{Prob}[S(A) = \tau] = 1 \\ 0 & sinon \end{cases}$
2. $\alpha \in \text{Out}(A)$ *alors* $q = p \times \text{Prob}[S(A) = \alpha$ *et si* $q \neq 0$ *alors pour tout* $\beta \in \text{Out}(A)$ *t.q.* $\text{support}(\alpha) = \text{support}(\beta)$ *il existe* $\psi' \rhd Q' \in Conf$ *t.q.* $\phi \rhd P \xrightarrow{\beta[p']} \psi' \rhd Q'$, $\phi \rhd P \xrightarrow{\beta[q']}_S \psi' \rhd Q'$ *et* $\frac{p}{p'} = \frac{q}{q'}$.
3. $\alpha = c(m)$ *et* $\exists_{D:\alpha \in D}$ *alors* $q = p \times \text{Prob}[S(A) = \alpha] \times \text{Prob}[F(\phi, c) = D]$.

En d'autres termes, l'ordonnancement ne modifie pas les distributions probabilistes des actions internes d'un processus. L'ordonnancement n'influence que globalement la distribution probabiliste des actions d'output sur un même canal. Autrement dit, l'attaquant peut influer sur la distribution probabiliste des canaux d'output, mais pas sur leurs contenus. L'attaquant a une influence totale sur la distribution probabiliste

des actions d'input.

La preuve de 1 découle de la condition 1 de la Définition 5.2.5. En effet, d'après cette condition, on peut permettre à l'attaquant soit "d'attaquer" le protocole (i.e. d'intercepter un message émis par le protocole ou d'émettre un message sur l'un des ses canaux d'écoute) en choisissant une action visible (disponible bien sûr !) avec une distribution probabiliste conforme à son ordonnanceur : dans ce cas, aucune action interne n'est affectée. On peut aussi ne rien faire (à condition qu'au moins une action interne soit disponible, conformément à la condition de stochasticité) : dans ce cas, l'une des actions internes est exécutée sans que leur distribution probabiliste ne soit modifiée.

La preuve de 2 découle de la définition de la classe d'équivalence des outputs (Définition 5.2.3) et de la condition 1 de la Définition 5.2.5. En effet, la définition des classes équiprobables exige que, si plusieurs actions différentes d'output sur un même canal sont exécutables à un état donné, alors elles sont stratégiquement équiprobables, i.e. choisies avec la même probabilité. Ici encore, voyons comment notre stratégie diffère de celle de Mitchell *et al.*, 2006. Considérons à nouveau le processus $P = \overline{c}(m)|\overline{c}(m')$ qui envoie sur le canal public c soit le message m suivi de m' soit le message m' suivi de m. Soit $(\phi, (F, \zeta))$ un attaquant, avec notre stratégie, nous avons $A = \chi(\phi \rhd P, F) = \{\overline{c}(m), \overline{c}(m')\}$. Ces deux actions étant $(\text{support}(\overline{c}(m')) = \text{support}(\overline{c}(m)))$ équiprobables, d'après la condition de stochasticité, on a :

$$\text{Prob}[\zeta(A) = \overline{c}(m)] + \text{Prob}[\zeta(A) = \overline{c}(m')] = 1.$$

L'une, au moins, de ces probabilités est donc non nulle. Supposons que c'est la première et posons $\text{Prob}[\zeta(A) = \overline{c}(m)] = p \neq 0$. D'après la condition 2, on a :

$$\text{Prob}[\zeta(A) = \overline{c}(m')] = \text{Prob}[\zeta(A) = \overline{c}(m)] = p.$$

On a donc $2p = 1$ qui implique que $p = 1/2$ et ceci quelque soit l'ordonnanceur choisi.

Ce qui reflète bien la nature totalement non déterministe de P.

Nous obtenons alors le résultat important suivant :

Théorème 5.2.1 *La somme des probabilités des transitions sortantes d'un sommet selon un ordonnanceur quelconque est inférieure ou égale à 1.*

Preuve:

Soit $\phi \rhd P$ une configuration, $S = (F, \zeta)$ un ordonnanceur et $\text{Exec}_S(\phi \rhd P)$ l'ensemble des transitions sortantes de $\phi \rhd P$. Nous avons deux cas :

$[P \notin \mathcal{B}locked]$: dans ce cas nous avons

$$\text{Exec}_S(\phi \rhd P) = \{\phi \rhd P \xrightarrow{\tau[q]} \phi \rhd Q \mid \mathit{eval}(\text{P}) \xrightarrow{q} Q\}$$

puisque $\chi(\phi \rhd P, F) = \{\tau\}$ et $\Upsilon(P, \tau) = 1$.

La somme des probabilités des transitions sortantes de $\phi \rhd P$ selon S est

$$\begin{aligned} p &= \sum_{Q \in Im(\mathit{eval}(\text{P}))} (\text{Prob}[\mathit{eval}(\text{P}) = Q] \times \text{Prob}[\zeta(\{\tau\}) = \tau]) \\ &= \sum_{Q \in Im(\mathit{eval}(\text{P}))} \text{Prob}[\mathit{eval}(\text{P}) = Q] \qquad \text{(condition de stochasticité de } \zeta.) \\ &\leq 1 \qquad \text{(par définition de } \mathit{eval}.) \end{aligned}$$

$[P \in \mathcal{B}locked]$: dans ce cas, nous avons

$$\text{Exec}_S(\phi \rhd P) = \{t_{\alpha,i} = \phi \rhd P \xrightarrow{\alpha[p_{\alpha,i}]} \psi \rhd Q \mid \alpha \in \chi(\phi \rhd P, F) \text{ et } i \in [1..\Upsilon(P, \alpha)]\}.$$

La somme des probabilités des transitions sortantes de $\phi \rhd P$ selon S est

$$
\begin{aligned}
p &= \sum_{\alpha \in \chi(\phi \rhd P)} \text{Prob}[\zeta(\chi(\phi \rhd P, F)) = \alpha] \times q_\alpha \times \Big(\sum_{i \in [1..\Upsilon(P,\alpha)]} p_{\alpha,i} \Big) \\
&\quad \text{où } q_\alpha = \begin{cases} 1 & \text{si } \alpha \text{ n'est pas une action d'input} \\ \max_{\{D \in Im(F(\phi,c)) \mid m \in D\}}(\text{Prob}[F(\phi, c) = D]) & \text{si } \alpha = c(m) \end{cases} \\
&\leq \sum_{\alpha \in \chi(\phi \rhd P, F)} \text{Prob}[\zeta(\chi(\phi \rhd P, F)) = \alpha] \times q_\alpha \\
&\quad \Big(\sum_{i \in [1..\Upsilon(P,\alpha)]} p_{\alpha,i} \leq 1 : \text{ les } p_{\alpha,i} \text{ étant des probabilités normalisées de la} \\
&\quad \text{forme } \frac{p'_{\alpha,i}}{\Upsilon(P,\alpha)} \Big) \\
&\leq \sum_{\alpha \in \chi(\phi \rhd P, F)} \text{Prob}[\zeta(\chi(\phi \rhd P, F)) = \alpha] \\
&\quad \text{(puisque } q_\alpha \leq 1 : \text{ F étant une fonction probabiliste polynomiale)} \\
&\leq 1 \qquad \text{(par définition de } \zeta).
\end{aligned}
$$

■

5.2.4 Probabilité cumulative

Pour conclure cette section sur la sémantique contextuelle de ProSPA, nous adaptons les résultats sur le calcul des probabilités cumulatives dans la nouvelle sémantique. Soit $\sigma = \alpha_1 \alpha_2 \ldots \alpha_n$ une séquence d'actions. Nous disons que σ est un chemin de la configuration $\phi \rhd P$ à la configuration $\psi \rhd Q$ s'il existe des probabilités non nulles $p_1, p_2, \ldots, p_n$ telles que

$$\phi_0 \rhd P_0 \xrightarrow{\alpha_1[p_1]} \phi_1 \rhd P_1 \xrightarrow{\alpha_2[p_2]} \cdots \xrightarrow{\alpha_n[p_n]} \phi_n \rhd P_n,$$

$\phi = \phi_0$, $\psi = \phi_n$, $P = P_0$ et $Q = P_n$. Si aucune confusion n'est à craindre, nous dirons simplement que σ est un chemin de P à Q. De même, nous disons que $\phi \rhd P$

atteint $\psi \rhd Q$ par le chemin σ avec la probabilité p suivant l'ordonnancement $S = (F, \zeta)$, dénoté par $\phi \rhd P \stackrel{\sigma[p]}{\Longrightarrow}_S \psi \rhd Q$, si la probabilité que S choisit σ, définie par $\text{Prob}[S(\sigma)] = \prod_{1 \leq i \leq n} q_i$ où

$$q_i = \begin{cases} \text{Prob}[\zeta(\chi(\phi_{i-1} \rhd P_{i-1}, F)) = \alpha_i] p_i) & \text{si } \alpha \text{ n'est pas un input partiel} \\ \text{Prob}[\zeta(\chi(\phi_{i-1} \rhd P_{i-1}, F)) = \alpha_i] \times p_i \times & \\ \quad (\max_{D \in Im(F(\phi_{i-1}, c))} \text{Prob}[F(\phi_{i-1}, c) = D]) & \text{sinon.} \end{cases}$$

est égale à p. Soit α une action, nous désignons par *α-chemin* tout chemin de la forme $\tau^* \alpha$. La notation $\phi \rhd P \stackrel{\alpha[p]}{\Longrightarrow}_S \psi \rhd Q$ signifie qu'il existe un α-chemin σ tel que $\phi \rhd P \stackrel{\sigma[p]}{\Longrightarrow}_S \psi \rhd Q$. De même, $\phi \rhd P \stackrel{\hat{\alpha}[p]}{\Longrightarrow}_S \psi \rhd Q$ désigne $\phi \rhd P \stackrel{\alpha[p]}{\Longrightarrow}_S \psi \rhd Q$ si $\alpha \neq \tau$ et $\phi \rhd P \stackrel{\tau^*[p]}{\Longrightarrow}_S \psi \rhd Q$ sinon.

Soit *Conf* l'ensemble de toutes les configurations, $\mathscr{E} \subseteq$ *Conf* un ensemble de configurations et $\psi \rhd Q \in \mathscr{E}$. Soit P un processus, ϕ un environnement et σ un α-chemin de $\phi \rhd P$ à $\psi \rhd Q$, nous dirons que σ est *minimal* conformément à $\mathscr{E}$ s'il n'existe pas un autre α-chemin σ' de $\phi \rhd P$ à $\psi \rhd Q'$ tel que σ' est un préfixe de σ et $\psi \rhd Q' \in \mathscr{E}$.

Définition 5.2.6 : *Soit $\mathscr{E} \subseteq$ Conf un ensemble de configurations, P un processus, ϕ un environnement et α une action. Nous désignons par Paths$(\phi \rhd P, \stackrel{\alpha}{\Longrightarrow}, \mathscr{E})$ le multi-ensemble de tous les α-chemins minimaux de $\phi \rhd P$ vers un élément de $\mathscr{E}$.*

Notons que, si σ et σ' sont deux α-chemins tels que $\phi \rhd P \stackrel{\sigma[p]}{\Longrightarrow}_S \psi \rhd Q$ et $\phi \rhd P \stackrel{\sigma'[p']}{\Longrightarrow}_S \psi' \rhd Q'$, alors $\psi = \psi'$ puisque les deux chemins ne diffèrent que par le nombre d'actions τ qui précèdent α. Notons également que la condition de minimalité ne s'applique qu'aux τ-chemins.

Définition 5.2.7 : *Soit $\mathscr{E} \subseteq$ Conf un ensemble de configurations, P un processus, ϕ un environnement, S un ordonnanceur et α une action. La probabilité totale que la configuration $\phi \rhd P$ atteigne une configuration dans $\mathscr{E}$ par un α-chemin selon*

l'ordonnanceur S est calculée par la fonction de probabilité cumulative
$\mu : Conf \times \mathcal{A}ct \times 2^{\mathcal{C}onf} \times Sched \rightarrow [0,1]$ *définie par*

$$\mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathscr{E}) = \begin{cases} 1 & si\ \phi \rhd P \in \mathscr{E}\ et\ \alpha = \tau \\ \sum_{\sigma \in Paths(\phi \rhd P, \stackrel{\alpha}{\Longrightarrow}, \mathscr{E})} \text{Prob}[S(\sigma)] & sinon \end{cases}$$

Nous pouvons démontrer (par une induction sur la longueur des α-chemins) le théorème ci-dessous.

Théorème 5.2.2 *La fonction de probabilité cumulative est bien définie, i.e.*

$$\forall_{P,\alpha,\mathscr{E},S,\phi}\ \mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathscr{E}) \leq 1.$$

5.3 Équivalences asymptotiques

Dans cette section, nous allons redéfinir nos équivalences asymtotiques du chapitre précédent qui stipulent que deux processus sont équivalents, si et seulement si, soumis aux mêmes attaques, ils engendrent "approximativement" les mêmes observations. Par approximativement, nous voulons dire *asymptotiquement* proches par rapport au paramètre de sécurité. Nous allons montrer que, contrairement au cas concret, les nouvelles équivalences engendrées par cette sémantique contextuelle ne sont pas des congruences pour les opérateurs de restriction et de parallélisme. Nous allons également montrer qu'elles ne sont plus équivalentes : l'équivalence de trace est plus fine que l'équivalence observationnelle.

5.3.1 Équivalence observationnelle asymptotique

Nous commençons par définir la probabilité d'observer une action visible qui nous servira à établir l'équivalence observationnelle. Nous définissons la probabilité d'observer une action visible α comme somme de la probabilité d'observer directement α, i.e. la probabilité cumulative de faire un α-chemin pour atteindre n'importe quel état plus la probabilité de l'observer indirectement, i.e. en observant d'abord des actions visibles différentes de α avant de l'observer. Pour cela, nous avons besoin d'étendre la notion de probabilité cumulative à une notion de probabilité cumulative plus générale que nous appelons *probabilité cumulative "up to H" dans un environnement ϕ* où $H \subset \mathcal{V}is$.

Définition 5.3.1 : *Soit $\mathscr{E} \subseteq Conf$ un ensemble de configurations, P un processus, ϕ un environnement, S un ordonnanceur et H un ensemble d'actions visibles. La* probabilité cumulative "up to H" *est define récursivement comme suit :* $\forall \alpha \in \mathcal{A}ct \setminus H$

$$\begin{aligned}
\mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/H}, \mathscr{E}) & \\
&= 1 \qquad si\ \phi \rhd P \in \mathscr{E} \quad et \quad \alpha = \tau, \\
&= \mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S}, \mathscr{E}) + \\
&\quad \sum_{\beta \in H,\ \psi \rhd Q \in Conf} \mu(\phi \rhd P, \stackrel{\hat{\beta}}{\Longrightarrow}_{S}, \{\psi \rhd Q\}) \mu(\psi \rhd Q, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/H}, \mathscr{E})
\end{aligned}$$

Lemme 5.3.1 *La fonction de probabilité cumulative up to H est bien définie, i.e.*

$$\forall_{P,\alpha,\mathscr{E},S,H,\phi}\ \mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/H}, \mathscr{E}) \leq 1.$$

Preuve: Si $\phi \rhd P \in \mathscr{E}$ et $\alpha = \tau$, alors $\mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/H}, \mathscr{E}) = 1$ et le résultat en découle.

Supposons qu'on n'est pas dans ce cas. Soit $\pi = \tau^*\alpha_1 \cdots \tau^*\alpha_n$ un chemin, $\phi \rhd P$ une configuration et S un ordonnanceur. Nous disons que l'action α_i $(1 \leq i \leq n)$ est à une distance i de la configuration $\phi \rhd P$ sur le chemin π selon l'ordonnanceur S, dénoté par $d_\pi(\phi \rhd P, \alpha_i, S) = i$ s'il existe une probabilité non nulle p et une configuration $\psi \rhd Q$ telles que $\phi \rhd P \stackrel{\pi[p]}{\Longrightarrow}_S \psi \rhd Q$ et $\forall j < i\ \alpha_j \neq \alpha_i$. Autrement dit, $d_\pi(\phi \rhd P, \alpha, S)$ est la première position de l'action visible α sur le chemin π slon S. De même, soit $\alpha \notin H$ une action visible, posons $\Pi_H(\alpha) = \{\pi = \tau^*\alpha_1\tau^*\alpha_2 \cdots \tau^*\alpha_n\tau^*\alpha : n \in \mathbb{N}$ et $\forall i \leq n,\ \alpha_i \in H\}$, nous disons que l'action α est à une distance maximale k de la configuration $\phi \rhd P$ selon S conformément à l'ensemble H, dénoté par $d_H(\phi \rhd P, \alpha, S) = k$ si $\sup_{\pi \in \Pi_H(\alpha)}(d_\pi(\phi \rhd P, \alpha, S) = k$. Nous allons prouver le lemme par une induction sur $d_H(\phi \rhd P, \alpha, H)$. Soit $\alpha \in \mathcal{A}ct \setminus H$, nous avons :

- *[Base :]* Si $d_H(\phi \rhd P, \alpha, S) = 0$, alors

$$\mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/H}, \mathscr{E}) = \mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathcal{C}onf) \leq 1.$$

  La base d'induction est donc établie.
- *[Étape d'induction :]* Supposons maintenant, que pour tout ϕ, P et H tels que $d_H(\phi \rhd P, \alpha, S) < n$, alors $\mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/H}, \mathscr{E}) \leq 1$.
  Puisque

$$\begin{aligned}\mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/H}, \mathscr{E}) &= \mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathscr{E}) \\ &+ \sum_{\beta \in H,\ \psi \rhd Q \in \mathcal{C}onf} \mu(\phi \rhd P, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \{\psi \rhd Q\})\mu(\psi \rhd Q, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/H}, \mathscr{E})\end{aligned}$$

  et que l'hyphothèse d'induction nous garantit que $\mu(\psi \rhd Q, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/H}, \mathscr{E}) \leq 1$, il nous reste à montrer que

$$\mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathscr{E}) + \sum_{\beta \in H,\ \psi \rhd Q \in \mathcal{C}onf} \mu(\phi \rhd P, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \{\psi \rhd Q\}) \leq 1$$

Or nous avons

$$\mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathscr{E}) \quad + \sum_{\beta \in H,\ \psi \rhd Q \in Conf} \mu(\phi \rhd P, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \{\psi \rhd Q\})$$
$$\leq \sum_{\beta \in H \cup \{\alpha\}} \mu(\phi \rhd P, \stackrel{\hat{\beta}}{\Longrightarrow}_S, Conf)$$

Nous terminons la preuve en remarquant que $\sum_{\beta \in H \cup \{\alpha\}} \mu(\phi \rhd P, \stackrel{\hat{\beta}}{\Longrightarrow}_S, Conf)$ est inférieur à la somme des probabilités des transitions sortantes de $\phi \rhd P$ selon l'ordonnanceur S, qui est inférieure à 1 d'après le Théorème 5.2.1.

■

Définition 5.3.2 : *Soit $\alpha \in \mathcal{V}is$ une action observable et P un processus. La probabilité que P génère α sous l'environnement ϕ selon l'ordonnanceur S est*

$$\text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha] = \mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/(\mathcal{V}is \setminus \{\alpha\})}, Conf)$$

Avant de définir notre relation d'équivalence observationnelle, remarquons que le Lemme 5.3.1 nous assure que $\text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha]$ est bien une probabilité.

Lemme 5.3.2 *La probabilité d'observer une action visible est bien définie, i.e.*

$$\forall_{P,\alpha,S,\phi}\ \text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha] \leq 1.$$

Nous pouvons maintenant définir notre relation d'équivalence observationnelle asymptotique qui stipule que deux processus sont équivalents s'ils engendrent les mêmes observations avec approximativement les mêmes probabilités lorsqu'ils évoluent dans le même environnement et selon le même ordonnanceur.

Définition 5.3.3 : *[Équivalence observationnelle asymptotique] Deux processus P et Q sont asymptotiquement équivalents dans l'environnement ϕ, dénoté par $P \simeq_\phi Q$, si et seulement si, $\forall q \in Poly$, $\forall \alpha \in \mathcal{V}is$, $\forall S \in Sched$, $\exists i_0$ t.q. $\forall \mathbf{N} \geq i_0$*

$$|\text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha] - \text{Prob}[\phi \rhd Q \rightsquigarrow_S \alpha]| \leq \frac{1}{q(\mathbf{N})}.$$

Théorème 5.3.1 *L'équivalence observationnelle asymptotique $\simeq_\phi$ est une relation d'équivalence.*

Preuve: La réflexivité et la symétrie sont évidentes. Nous allons donc prouver la transitivité. Soit P, Q et R des processus et ϕ un environnement tels que $P \simeq_\phi Q$ et $Q \simeq_\phi R$. Soit $q \in Poly$, $\alpha \in \mathcal{V}is$ et $S \in Sched$, alors on a :
$P \simeq_\phi Q \Rightarrow \exists i_0$ tel que $\forall \mathbf{N} \geq i_0$

$$|\text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha] - \text{Prob}[\phi \rhd Q \rightsquigarrow_S \alpha]| \leq \frac{1}{2q(\mathbf{N})},$$

et $Q \simeq_\phi R \Rightarrow \exists j_0$ tel que $\forall \mathbf{N} \geq j_0$

$$|\text{Prob}[\phi \rhd Q \rightsquigarrow_S \alpha] - \text{Prob}[\phi \rhd R \rightsquigarrow_S \alpha]| \leq \frac{1}{2q(\mathbf{N})}.$$

Nous avons donc, $\forall \mathbf{N} \geq k_0 = \max(i_0, j_0)$

$|\text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha] - \text{Prob}[\phi \rhd R \rightsquigarrow_S \alpha]|$

$$\begin{aligned}
&= |\text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha] \;-\; \text{Prob}[\phi \rhd Q \rightsquigarrow_S \alpha] \\
&\qquad\qquad + \;\text{Prob}[\phi \rhd Q \rightsquigarrow_S \alpha] - \text{Prob}[\phi \rhd R \rightsquigarrow_S \alpha]| \\
&\leq |\text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha] \;-\; \text{Prob}[\phi \rhd Q \rightsquigarrow_S \alpha]| \\
&\qquad\qquad + \;|\text{Prob}[\phi \rhd Q \rightsquigarrow_S \alpha] - \text{Prob}[\phi \rhd R \rightsquigarrow_S \alpha]| \\
&\qquad \leq \frac{1}{2q(\mathbf{N})} \;+\; \frac{1}{2q(\mathbf{N})} = \frac{1}{q(\mathbf{N})}.
\end{aligned}$$

D'où $P \simeq_\phi R$. ■

Lemme 5.3.3 *Soit P, Q et R des processus, ϕ un environnement et L un ensemble actions visibles. Nous avons :*

1. *Si $P \simeq_{\phi\cup\{a\}} Q$ alors $\overline{c}(a).P \simeq_\phi \overline{c}(a).Q$*
2. *Si $P \simeq_\phi Q$ alors $c(a).P \simeq_\phi c(a).Q$*
3. *Si $P \simeq_\phi Q$ alors $[\lambda(a_1,\cdots a_n)\overset{p}{\hookrightarrow} m]P \simeq_\phi [\lambda(a_1,\cdots a_n)\overset{p}{\hookrightarrow} m].Q$*

Preuve: 1. Nous avons $\text{Prob}[\phi \rhd \overline{c}(a).P \rightsquigarrow_S \alpha] = \mu(\phi \rhd P, \overset{\hat{\alpha}}{\Longrightarrow}_{S/(\mathcal{V}is\setminus\{\alpha\})}, \mathcal{C}onf)$. Alors, si $\alpha = \overline{c}(a)$, de la Définition 5.3.1, nous avons :

$$\begin{aligned}\text{Prob}[\phi \rhd \overline{c}(a).P \rightsquigarrow_S \overline{c}(a)] &= \sum_{\psi\rhd Q\in\mathcal{C}onf} \mu(\phi \rhd \overline{c}(a).P, \overset{\hat{\overline{c}(a)}}{\Longrightarrow}_S, \{\psi \rhd Q\}),\\ &= \mu(\phi \rhd \overline{c}(a).P, \overset{\hat{\overline{c}(a)}}{\Longrightarrow}_S, \{\phi\cup\{a\} \rhd P\}) = 1.\end{aligned}$$

De même,

$$\text{Prob}[\phi \rhd \overline{c}(a).Q \rightsquigarrow_S \overline{c}(a)] = \mu(\phi \rhd \overline{c}(a).Q, \overset{\hat{\overline{c}(a)}}{\Longrightarrow}_S, \{\phi\cup\{a\} \rhd Q\}) = 1.$$

D'où $|\text{Prob}[\phi \rhd \overline{c}(a).P \rightsquigarrow_S \overline{c}(a)] - \text{Prob}[\phi \rhd \overline{c}(a).Q \rightsquigarrow_S \overline{c}(a)]| = 0$.

Si $\alpha \neq \overline{c}(a)$ alors de la Définition 5.3.1 nous avons :

$\text{Prob}[\phi \rhd \overline{c}(a).P \rightsquigarrow_S \alpha]$

$$\begin{aligned}&= \sum_{\beta\in\mathcal{V}is\setminus\{\alpha\},\ \psi\rhd Q\in\mathcal{C}onf} \mu(\phi \rhd \overline{c}(a).P, \overset{\hat{\beta}}{\Longrightarrow}_S, \mathcal{C}onf)\mu(\psi \rhd Q, \overset{\hat{\beta}}{\Longrightarrow}_{S/(\mathcal{V}is\setminus\{\alpha\})}, \mathcal{C}onf)\\ &= \mu(\phi \rhd \overline{c}(a).P, \overset{\hat{\overline{c}(a)}}{\Longrightarrow}_S, \{\phi\cup\{a\} \rhd P\})\mu(\phi\cup\{a\} \rhd P, \overset{\hat{\alpha}}{\Longrightarrow}_{S/(\mathcal{V}is\setminus\{\alpha\})}, \mathcal{C}onf),\end{aligned}$$

C'est-à-dire

$$\text{Prob}[\phi \rhd \overline{c}(a).P \rightsquigarrow_S \alpha] = 1 \times \text{Prob}[\phi\cup\{a\} \rhd P \rightsquigarrow_S \alpha].$$

De même, nous avons

$$\text{Prob}[\phi \rhd \overline{c}(a).Q \rightsquigarrow_S \alpha] = 1 \times \text{Prob}[\phi \cup \{a\} \rhd Q \rightsquigarrow_S \alpha].$$

Puisque $P \simeq_{\phi \cup \{a\}} Q$, nous avons $\overline{c}(a).P \simeq_\phi \overline{c}(a).Q$.

2. La preuve de $c(a).P \simeq_\phi c(a).Q$ est identique à la preuve précédente à l'exception du fait que l'environnement n'évolue pas.

3. Nous avons, $\text{Prob}[\phi \rhd [\lambda(a_1, \cdots a_n) \overset{p}{\hookrightarrow}].P \rightsquigarrow_S \alpha] = p \times \text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha]$, et de même, $\text{Prob}[\phi \rhd [\lambda(a_1, \cdots a_n) \overset{p}{\hookrightarrow}].Q \rightsquigarrow_S \alpha] = p \times \text{Prob}[\phi \rhd Q \rightsquigarrow_S \alpha]$. Finalement le résultat découle du fait qu'on a $P \simeq_\phi Q$. ■

Lemme 5.3.4 *L'équivalence observationnelle asymptotique n'est pas une congruence pour les opérateurs de restriction et le parallélisme, i.e.*

1. $P \simeq_\phi Q \not\Rightarrow P\backslash\Gamma \simeq_\phi Q\backslash\Gamma$
2. $P \simeq_\phi Q \not\Rightarrow P|R \simeq_\phi Q|R$

Preuve: Nous allons donner un exemple de deux processus obsrervationnellement équivalents, mais qui ne le sont pas par restriction ou par composition parallèle. Soit $\lambda(n) \overset{\frac{1}{n+1}}{\hookrightarrow} i$ $(0 \leq i \leq n$ et t.q. $n+1$ est une puissance de 2), $\alpha = \overline{c}(m)$ et $\beta = \overline{c}(m')$. Considérons alors les processus suivants :

$$P = \alpha.\beta.\mathbf{0}, \quad Q = [\lambda(1) \hookrightarrow x]([x = 0]\alpha.\beta.\mathbf{0}|[x = 1]\beta.\alpha.\mathbf{0}) \text{ et } R = c(y).\mathbf{0}$$

Il est clair que $P \simeq_\phi Q$ quel que soit l'environnement ϕ, mais $P\backslash\Gamma \not\simeq_\phi Q\backslash\Gamma$ si $\Gamma = \{\alpha\}$. En effet, $P\backslash\Gamma = \mathbf{0}$, alors que $Q\backslash\Gamma \neq \mathbf{0}$ et permet d'observer β avec une probabilité non nulle. C'est évidemment dû au fait que l'ordre d'apparition des actions n'a pas d'importance pour l'équivalence observationnelle et que, contrairement au cas concret, l'attaquant ne peut pas insérer ses propres observations pour pouvoir distinguer P de Q. De même, $P|R \not\simeq_\phi Q|R$. En effet, considérons l'ordonnanceur $S = (\zeta, F)$

privilégiant les deux actions de communication $\alpha \cdot \overline{\alpha}$ et $\beta \cdot \overline{\beta}$, i.e. formellement $\alpha \cdot \overline{\alpha} \in A \Rightarrow \zeta(A) \xrightarrow{1} \alpha \cdot \overline{\alpha}$ et $\beta \cdot \overline{\beta} \in A \Rightarrow \zeta(A) \xrightarrow{1} \beta \cdot \overline{\beta}$[1], et F une fonction de sélection quelconque. Alors, $\text{Prob}[\phi \rhd P | R \rightsquigarrow_S \alpha] = 0$ et $\text{Prob}[\phi \rhd Q | R \rightsquigarrow_S \alpha] = \frac{1}{2}$ ■

Bien que pratique, l'équivalence observationnelle contextuelle s'avère non suffisament forte devant un attaquant qui essaie, par exemple, de distinguer deux systèmes en se basant sur l'observation d'une séquence d'actions visibles au lieu d'une action à la fois. Pour illustrer ce fait brièvement, considérons par exemple les processus $P = \overline{c}(a).\overline{c}(a).\mathbf{0}$ et $Q = \overline{c}(a).\mathbf{0}$. Ces deux processus sont insensibles à l'ordonnancement et $\forall S \in \mathit{Sched}\, \forall \alpha \in \mathcal{V}is$ et ϕ on a

$$\begin{cases} \text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha] = \text{Prob}[\phi \rhd Q \rightsquigarrow_S \alpha] = 1 & \text{si } \alpha = \overline{c}(a) \\ \text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha] = \text{Prob}[\phi \rhd Q \rightsquigarrow_S \alpha] = 0 & \text{sinon} \end{cases}$$

Donc, on a bien $P \simeq_\phi Q$. Ainsi, aucun attaquant ne peut distinguer P et Q bien qu'ils engendrent des traces observables différentes. Ici encore, contrairement à la sémantique concrète, où l'attaquant $\Pi = c(x).c(y).\overline{c'}(b).\mathbf{0}$ peut distinguer ces deux processus grâce à sa propre observation $\overline{c'}(b)$ insérée après avoir observé la séquence aa, la sémantique contextuelle ne lui permet pas ce genre de manoeuvre. Pour remédier à ce problème, nous allons étendre l'équivalence observationnelle pour inclure toutes les traces observables.

5.3.2 Équivalence de trace asymptotique

Nous commençons d'abord par définir la probabilité d'observer une suite d'actions visibles qui nous servira à établir notre nouvelle équivalence. Nous définissons la probabilité d'observer $\alpha_1 \alpha_2 \cdots \alpha_n$ récursivement comme la probabilité d'observer directe-

[1] À noter que ces deux actions ne peuvent pas être disponibles tous les deux à la fois dans un même état.

ment α_1, i.e. la probabilité cumulative de faire un α_1-chemin pour atteindre n'importe quel état fois la probabilité que cet état engendre $\alpha_2 \cdots \alpha_n$.

Définition 5.3.4 : *Soit $\alpha_1\alpha_2 \cdots \alpha_n$ une suite d'actions observables et P un processus. La probabilité que P génère $\alpha_1\alpha_2 \cdots \alpha_n$ sous l'environnement ϕ selon l'ordonnanceur S est*

$$\begin{aligned} \text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha_1\alpha_2 \cdots \alpha_n] \\ = \sum_{\psi \rhd Q \in Conf} \mu(\phi \rhd P, \stackrel{\hat{\alpha_1}}{\Longrightarrow}_S, \{\psi \rhd Q\}) \cdot \text{Prob}[\psi \rhd Q \rightsquigarrow_S \alpha_2 \cdots \alpha_n]. \end{aligned}$$

Avant de définir notre équivalence, nous avons le lemme suivant (qui découle d'une simple induction sur la longueur des traces observables et le Théorème 5.2.2) nous assurant que la probabilité d'observer une séquence d'actions visibles est bien définie.

Lemme 5.3.5 *La probabilité d'observer une suite d'actions visibles est bien définie, i.e., $\forall_{P,\alpha_1,\alpha_2,\cdots,\alpha_n,S,\phi}$ $\text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha_1\alpha_2 \cdots \alpha_n] \leq 1$.*

Nous pouvons maintenant définir notre relation d'équivalence de trace asymptotique qui stipule que deux processus sont équivalents, si et seulement s'ils engendrent les mêmes traces observables avec approximativement les mêmes probabilités lorsqu'ils sont soumis aux mêmes attaques, i.e. lorsqu'ils évoluent dans le même environnement et selon le même ordonnanceur.

Définition 5.3.5 : *[Équivalence de trace asymptotique] Deux processus P et Q sont asymptotiquement trace-équivalents dans l'environnement ϕ, dénoté par $P \simeq_\phi^{tr} Q$, si et seulement si $\forall q \in \mathcal{P}oly$, $\forall \alpha_1, \alpha_2, \cdots, \alpha_n \in \mathcal{V}is$, $\forall S \in Sched$, $\exists i_0$ tel que $\forall \mathbf{N} \geq i_0$*

$$|\text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha_1\alpha_2 \cdots \alpha_n] - \text{Prob}[\phi \rhd Q \rightsquigarrow_S \alpha_1\alpha_2 \cdots \alpha_n]| \leq \frac{1}{q(\mathbf{N})}.$$

Théorème 5.3.2 *L'équivalence de trace asymptotique $\simeq_\phi^{tr}$ est une relation d'équivalence.*

Preuve: La réflexivité et la symétrie sont évidentes. Nous allons donc prouver la transitivité. Soit donc P, Q et R, des processus et ϕ un environnement tels que $P \simeq_\phi Q$ et $Q \simeq_\phi R$. Soit $q \in Poly$, s une séquence d'actions observables et $S \in Sched$, alors on a :
$P \simeq_\phi^{tr} Q \Rightarrow \exists i_0$ tel que $\forall \mathbf{N} \geq i_0$

$$|\text{Prob}[\phi \rhd P \rightsquigarrow_S s] - \text{Prob}[\phi \rhd Q \rightsquigarrow_S s]| \leq \frac{1}{2q(\mathbf{N})},$$

et $Q \simeq_\phi^{tr} R \Rightarrow \exists j_0$ tel que $\forall \mathbf{N} \geq j_0$

$$|\text{Prob}[\phi \rhd Q \rightsquigarrow_S s] - \text{Prob}[\phi \rhd R \rightsquigarrow_S s]| \leq \frac{1}{2q(\mathbf{N})}.$$

Nous avons donc, $\forall \mathbf{N} \geq k_0 = \max(i_0, j_0)$

$|\text{Prob}[\phi \rhd P \rightsquigarrow_S s] - \text{Prob}[\phi \rhd R \rightsquigarrow_S s]|$

$$\begin{aligned}
= |\text{Prob}[\phi \rhd P \rightsquigarrow_S s] &- \text{Prob}[\phi \rhd Q \rightsquigarrow_S s] \\
&+ \text{Prob}[\phi \rhd Q \rightsquigarrow_S s] - \text{Prob}[\phi \rhd R \rightsquigarrow_S s]| \\
\leq |\text{Prob}[\phi \rhd P \rightsquigarrow_S s] &- \text{Prob}[\phi \rhd Q \rightsquigarrow_S s]| \\
&+ |\text{Prob}[\phi \rhd Q \rightsquigarrow_S s] - \text{Prob}[\phi \rhd R \rightsquigarrow_S s]| \\
\leq \frac{1}{2q(\mathbf{N})} &+ \frac{1}{2q(\mathbf{N})} = \frac{1}{q(\mathbf{N})}.
\end{aligned}$$

D'où $P \simeq_\phi^{tr} R$. ■

Lemme 5.3.6 *Soit P, Q et R, des processus, ϕ un environnement et L un ensemble d'actions visibles. Nous avons alors :*

1. *Si* $P \simeq^{tr}_{\phi\cup\{a\}} Q$ *alors* $\overline{c}(a).P \simeq^{tr}_{\phi} \overline{c}(a).Q$
2. *Si* $P \simeq^{tr}_{\phi} Q$ *alors* $c(a).P \simeq^{tr}_{\phi} c(a).Q$
3. *Si* $P \simeq^{tr}_{\phi} Q$ *alors* $[\lambda(a_1,\cdots a_n)\overset{p}{\hookrightarrow} m]P \simeq^{tr}_{\phi} [\lambda(a_1,\cdots a_n)\overset{p}{\hookrightarrow} m].Q$

Preuve: 1. Nous allons le prouver par induction sur la longueur de la séquence s.
[*Base :*] $s = \alpha$. Nous avons $\text{Prob}[\phi \rhd \overline{c}(a).P \rightsquigarrow_S \alpha] = \mu(\phi \rhd P, \overset{\hat{\alpha}}{\Longrightarrow}_{S/(\mathcal{V}is\setminus\{\alpha\})}, \mathcal{C}onf)$,
et alors :
si $\alpha = \overline{c}(a)$, de la Définition 5.3.1, nous avons :

$$\begin{aligned}\text{Prob}[\phi \rhd \overline{c}(a).P \rightsquigarrow_S \overline{c}(a)] &= \sum_{\psi\rhd Q\in\mathcal{C}onf} \mu(\phi \rhd \overline{c}(a).P, \overset{\overline{c}(a)}{\Longrightarrow}_S, \{\psi \rhd Q\}),\\ &= \mu(\phi \rhd \overline{c}(a).P, \overset{\overline{c}(a)}{\Longrightarrow}_S, \{\phi\cup\{a\} \rhd P\}) = 1.\end{aligned}$$

De même : $\text{Prob}[\phi \rhd \overline{c}(a).Q \rightsquigarrow_S \overline{c}(a)] = \mu(\phi \rhd \overline{c}(a).Q, \overset{\overline{c}(a)}{\Longrightarrow}_S, \{\phi\cup\{a\} \rhd Q\}) = 1$.
D'où $|\text{Prob}[\phi \rhd \overline{c}(a).P \rightsquigarrow_S \overline{c}(a)] - \text{Prob}[\phi \rhd \overline{c}(a).Q \rightsquigarrow_S \overline{c}(a)]| = 0$.
Si $\alpha \neq \overline{c}(a)$ alors, $\text{Prob}[\phi \rhd \overline{c}(a).P \rightsquigarrow_S \alpha] = \text{Prob}[\phi \rhd \overline{c}(a).Q \rightsquigarrow_S \alpha] = 0$.

[*Étape d'induction :*] $s = \alpha_1\alpha_2\cdots\alpha_n$. Si $\alpha_1 = \overline{c}(a)$ alors, par définition, nous avons :
$\text{Prob}[\phi \rhd \overline{c}(a).P \rightsquigarrow_S s]$

$$\begin{aligned}&= \sum_{\beta\in\mathcal{V}is\setminus\{\alpha\},\ \psi\rhd Q\in\mathcal{C}onf} \mu(\phi \rhd \overline{c}(a).P, \overset{\hat{\beta}}{\Longrightarrow}_S, \mathcal{C}onf)\mu(\psi \rhd Q, \overset{\hat{\beta}}{\Longrightarrow}_{S/(\mathcal{V}is\setminus\{\alpha\})}, \mathcal{C}onf)\\ &= \mu(\phi \rhd \overline{c}(a).P, \overset{\overline{c}(a)}{\Longrightarrow}_S, \{\phi\cup\{a\} \rhd P\})\mu(\phi\cup\{a\} \rhd P, \overset{\hat{\alpha}}{\Longrightarrow}_{S/(\mathcal{V}is\setminus\{\alpha\})}, \mathcal{C}onf),\end{aligned}$$

C'est-à-dire $\text{Prob}[\phi \rhd \overline{c}(a).P \rightsquigarrow_S \alpha] = 1 \times \text{Prob}[\phi\cup\{a\} \rhd P \rightsquigarrow_S \alpha]$. De même, nous avons $\text{Prob}[\phi \rhd \overline{c}(a).Q \rightsquigarrow_S \alpha] = 1 \times \text{Prob}[\phi\cup\{a\} \rhd Q \rightsquigarrow_S \alpha]$. Enfin, puisque $P \simeq^{tr}_{\phi\cup\{a\}} Q$, nous avons $\overline{c}(a).P \simeq^{tr}_{\phi} \overline{c}(a).Q$.

2. La preuve de $c(a).P \simeq^{tr}_{\phi} c(a).Q$ est identique à la preuve précédente à l'exception du fait que l'environnement n'évolue pas.

3. Nous avons $\text{Prob}[\phi \rhd [\lambda(a_1, \cdots a_n)\overset{p}{\hookrightarrow}].P \rightsquigarrow_S s] = p \times \text{Prob}[\phi \rhd P \rightsquigarrow_S s]$, et de même, $\text{Prob}[\phi \rhd [\lambda(a_1, \cdots a_n)\overset{p}{\hookrightarrow}].Q \rightsquigarrow_S s] = p \times \text{Prob}[\phi \rhd Q \rightsquigarrow_S s]$. Finallement, le résultat découle du fait qu'on a $P \simeq_\phi^{tr} Q$. ■

Lemme 5.3.7 *L'équivalence de trace contextualisée n'est pas une congruence pour les opérateurs de restriction et le parallélisme, i.e.*

1. $P \simeq_\phi^{tr} Q \not\Rightarrow P\backslash\Gamma \simeq_\phi^{tr} Q\backslash\Gamma$
2. $P \simeq_\phi^{tr} Q \not\Rightarrow P|R \simeq_\phi^{tr} Q|R$

Preuve: Voici un contre-exemple. Considérons les processus suivants :

$$P = (\overline{c}(m).\mathbf{0}|c(x).\overline{c'}(a).\mathbf{0})\backslash\{c\},\ Q = \overline{c'}(a).\mathbf{0} \text{ and } R = \overline{c'}(b).\mathbf{0}.$$

Évidemment, nous avons $P \simeq_\phi^{tr} Q$ pour tout environement ϕ, mais $P|R \not\simeq_\phi Q|R$. En effet, considérons l'ordonnanceur $S = (F, \zeta)$ ne donnant pas priorité aux actions internes, i.e. $\zeta(A)\overset{0}{\hookrightarrow}\tau$ si $\{\tau\} \subsetneq A$. Alors, on a :

$$\text{Prob}[\phi \rhd P|R \rightsquigarrow_S \overline{c'}(a)\overline{c'}(b)] = 0 \text{ et } \text{Prob}[\phi \rhd P|R \rightsquigarrow_S \overline{c'}(b)\overline{c'}(a)] = 1,$$

tandis que

$$\text{Prob}[\phi \rhd Q|R \rightsquigarrow_S \overline{c'}(a)\overline{c'}(b)] = \text{Prob}[\phi \rhd Q|R \rightsquigarrow_S \overline{c'}(b)\overline{c'}(a)] = \frac{1}{2}.$$

■

Nous avons le résultat suivant qui établit que l'équivalence de trace asymptotique raffine l'équivalence observationnelle asymptotique.

Théorème 5.3.3 *Si deux processus P et Q sont trace-équivalents, alors ils sont observationnellement équivalents, i.e. $P \simeq_\phi^{tr} Q \Rightarrow P \simeq_\phi Q$.*

Preuve: La preuve se fait par induction sur la distance $d_H(\phi \rhd P, \alpha, S)$ définie dans la preuve du Lemme 5.3.1. Soit α une action visible, $S = (\zeta, F)$ un ordonnanceur, q un polynôme et $H = \mathcal{V}is \setminus \{\alpha\}$.

$$\begin{aligned}
\text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha] &= \mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/H}, \mathcal{C}onf) \\
&= \mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S}, \mathcal{C}onf) \\
&\quad + \sum_{\beta \in H,\ \psi \rhd R \in \mathcal{C}onf} \mu(\phi \rhd P, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \{\psi \rhd R\}) \\
&\qquad \times \text{Prob}[\psi \rhd R \rightsquigarrow_S \alpha].
\end{aligned}$$

Soit

$$Tr_k^H(\phi \rhd P, \alpha, S) = \{s = \alpha_1\alpha_2 \cdots \alpha_k\alpha \mid \forall_{i \leq k}\ \alpha_i \in H \text{ et } \text{Prob}[\phi \rhd P \rightsquigarrow_S^{tr} s] \neq 0\},$$

l'ensemble des traces observables de $\phi \rhd P$ terminant par l'action α précédée exactement de k actions dans H. Notons que $Tr_k^H(\phi \rhd P, \alpha)$ est un ensemble fini puisque, d'une part, ce sont des traces de longueurs finies (égales à $k+1$) et, d'autre part, le système de transitions sous un ordonnanceur S est à branchement fini. En effet à chaque état $\psi \rhd R$, l'ensemble des actions choisies par S est $Im(\zeta(\chi(\phi \rhd P, F)))$ qui est fini, puisque ζ est une fonction probabiliste (cf : Définition 4.2.1). Le nombre de transitions sortantes de $\psi \rhd R$ selon l'ordonnanceur S est inférieur ou égal à $\sum_{\beta \in Im(\zeta(\chi(\phi \rhd P, F)))} \Upsilon(R, \beta)$. Nous avons

$$\text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha] = \sum_{k \geq 0} \sum_{s \in Tr_k^H(\phi \rhd P, \alpha, S)} \text{Prob}[\phi \rhd P \rightsquigarrow_S^{tr} s].$$

De même,

$$\text{Prob}[\phi \rhd Q \rightsquigarrow_S \alpha] = \sum_{k \geq 0} \sum_{s \in Tr_k^H(\phi \rhd Q, \alpha)} \text{Prob}[\phi \rhd Q \rightsquigarrow_S^{tr} s].$$

[Base :] si $d_H(\phi \rhd P, \alpha, S) = d_H(\phi \rhd Q, \alpha, S) = 0$, alors on a :

$$\text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha] = \mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathcal{C}onf)$$

et

$$\text{Prob}[\phi \rhd Q \rightsquigarrow_S \alpha] = \mu(\phi \rhd Q, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathcal{C}onf).$$

Puisque $P \simeq^{tr}_{\phi} Q$, on a

$$|\mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathcal{C}onf) - \mu(\phi \rhd Q, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathcal{C}onf)| \leq \frac{1}{q(\mathbf{N})}$$

D'où

$$|\text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha] - \text{Prob}[\phi \rhd Q \rightsquigarrow_S \alpha]| \leq \frac{1}{q(\mathbf{N})}$$

[Étape d'induction :] supposons que $\max(d_H(P|\Pi, o, S), d_H(Q|\Pi, o, S) = n$, et $\forall_{q \in Poly}$ nous avons

$$\sum_{k<n} \sum_{s \in Tr^H_k(\phi \rhd P, \alpha, S) \cup Tr^H_k(\phi \rhd Q, \alpha, S)} |\text{Prob}[P|\Pi \rightsquigarrow^{tr}_S s] - \text{Prob}[Q|\Pi \rightsquigarrow^{tr}_S s]| \leq \frac{1}{2q(\mathbf{N})}.$$

Maintenant, posons

$$U_k = Tr^H_k(\phi \rhd P, \alpha, S) \cup Tr^H_k(\phi \rhd Q, \alpha, S),$$

alors on a :

$$\begin{aligned}
|\text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha] \; &- \; \text{Prob}[\phi \rhd Q \rightsquigarrow_S \alpha]| \\
&= \; |\sum_{k \leq n} \sum_{s \in U_k} (\text{Prob}[\phi \rhd P \rightsquigarrow^{tr}_S s] - \text{Prob}[\phi \rhd Q \rightsquigarrow^{tr}_S s])| \\
&\leq \; |\sum_{k<n} \sum_{s \in U_k} \text{Prob}[\phi \rhd P \rightsquigarrow^{tr}_S s] - \text{Prob}[\phi \rhd Q \rightsquigarrow^{tr}_S s]| \\
&\quad + \sum_{s \in U_n} |\text{Prob}[\phi \rhd P \rightsquigarrow^{tr}_S s] - \text{Prob}[\phi \rhd Q \rightsquigarrow^{tr}_S s]|
\end{aligned}$$

Par hypothèse d'induction, nous avons

$$|\sum_{k<n}\sum_{s\in U_k} \text{Prob}[\phi \rhd P \rightsquigarrow_S^{tr} s] - \text{Prob}[\phi \rhd Q \rightsquigarrow_S^{tr} s]| \leq \frac{1}{2q(\mathbf{N})}.$$

Posons $r_n = |Tr_n^H(\phi \rhd P, \alpha)| + |Tr_n^H(\phi \rhd Q, \alpha)|$.

$$P \simeq_\phi^{tr} Q \Rightarrow \forall_{s\in U_n} |\text{Prob}[\phi \rhd P \rightsquigarrow_S^{tr} s] - \text{Prob}[\phi \rhd Q \rightsquigarrow_S^{tr} s]| \leq \frac{1}{r_n \cdot 2q(\mathbf{N})}.$$

Soit

$$\sum_{s\in U_n} |\text{Prob}[\phi \rhd P \rightsquigarrow_S^{tr} s] - \text{Prob}[\phi \rhd Q \rightsquigarrow_S^{tr} s]| \leq \frac{r_n}{r_n \cdot 2q(\mathbf{N})} = \frac{1}{2q(\mathbf{N})}.$$

On a donc,

$$\begin{aligned} |\text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha] - \text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha]| &\leq \frac{1}{2q(\mathbf{N})} + \frac{1}{2q(\mathbf{N})} \\ &\leq \frac{1}{q(\mathbf{N})}. \end{aligned}$$

■

5.4 Bisimulation probabiliste

Nous adaptons la bisimulation probabiliste (faible) Mitchell *et al.*, 2006, qui elle-même est une adaptation de celle de Milner, 1989, en présence d'un ordonnanceur probabiliste au lieu d'un ordonnanceur non-déterministe. C'est une façon classique de définir la bisimulation faible dans un modèle probabiliste (Mitchell *et al.*, 2006; Aldini *et al.*, 2002; Bengt *et al.*, 2002) découlant des travaux de van Glabbeek *et al.*, 1995. L'idée de base consiste à remplacer les transitions de la forme $\phi \rhd P \stackrel{\hat{a}}{\Longrightarrow} \phi' \rhd P'$ de la Définition 2.4.3 par la probabilité cumulative d'atteindre, à partir de l'état $\phi \rhd P$,

un ensemble d'états (ie processus) équivalents via des séquences de transitions de la forme τ^*â.

Soit $\Re$ une relation d'équivalence sur l'ensemble des configurations $Conf$. L'ensemble quotient de $Conf$ par $\Re$ est dénoté par $Conf/_{\Re}$ et pour toute configuration $\phi \rhd P \in Conf$, la classe d'équivalence de $\phi \rhd P$ est désignée par $[\phi \rhd P]_{\Re}$.

Définition 5.4.1 : *[Bisimulation contextuelle] Une relation d'équivalence $\Re$ sur l'ensemble Conf est une* bisimulation faible, *si pour toute configuration $(\phi \rhd P, \psi \rhd Q) \in \Re$ on a :*

$$\forall \mathscr{U} \in Conf/\Re. \forall S \in Sched. \forall \alpha \in Act : \mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathscr{U}) = \mu(\psi \rhd Q, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathscr{U})$$

Deux configurations $\phi \rhd P$ et $\psi \rhd Q$ sont probabilistiquement bisimilaires, dénoté par $\phi \rhd P \approx \psi \rhd Q$, s'il existe une relation de bisimulation $\Re$ telle que $(\phi \rhd P, \psi \rhd Q) \in \Re$. Nous sommes outillés pour définir la bisimulation sous un environnement donné.

Définition 5.4.2 : *[Bisimulation sous un environnement] Deux processus P et Q sont probabilistiquement bisimilaires sous l'environnement ϕ, dénoté par $P \approx_\phi Q$, s'il existe une relation de bisimulation $\Re$ telle que $(\phi \rhd P, \phi \rhd Q) \in \Re$.*

Nous avons le résultat suivant établissant que l'équivalence observationnelle asymptotique est moins fine que la bisimulation probabiliste.

Théorème 5.4.1 *Si deux processus P et Q sont bisimilaires, alors ils sont observationnellement équivalents, i.e. $P \approx_\phi Q \Rightarrow P \simeq_\phi Q$.*

Preuve: Nous allons faire la preuve par induction sur la distance $d_H(\phi \rhd P, \alpha, S)$

définie dans la preuve du Lemme 5.3.1. Plus spécifiquement, nous allons montrer que pour toute action visible α et tout ordonnanceur S,

$$\text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha] = \text{Prob}[\phi \rhd Q \rightsquigarrow_S \alpha].$$

Supposons que $P \approx_\phi Q$. Remarquons tout d'abord que $P \approx_\phi Q$ implique qu'il existe une bisimulation $\Re$ t.q. $(\phi \rhd P, \phi \rhd Q) \in \Re$. De plus, grâce à la condition de minimalité des α-chemins, on a : $\forall \mathscr{U} \in \mathcal{C}onf/\Re.\forall S \in \mathcal{S}ched.\forall \alpha \in \mathcal{A}ct$:

$$\mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathscr{U}) = \sum_{\psi \rhd R \in \mathscr{U}} \mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \{\psi \rhd R\}).$$

Soit α une action visible, S un ordonnanceur et $H = \mathcal{V}is \setminus \{\alpha\}$
[Base :] si $d_H(\phi \rhd P, \alpha, S) = d_H(\phi \rhd Q, \alpha, S) = 0$, alors on a :

$$\begin{aligned}
\text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha] &= \mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathcal{C}onf), \\
&= \sum_{\psi \rhd R \in \mathcal{C}onf} \mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \{\psi \rhd R\}), \\
&= \sum_{\mathscr{U} \in \mathcal{C}onf/\Re} \mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathscr{U}).
\end{aligned}$$

De même, on a :

$$\text{Prob}[\phi \rhd Q \rightsquigarrow_S \alpha] = \sum_{\mathscr{U} \in \mathcal{C}onf/\Re} \mu(\phi \rhd Q, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathscr{U}).$$

Puisque

$$\forall \mathscr{U} \in \mathcal{C}onf/\Re : \mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathscr{U}) = \mu(\phi \rhd Q, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathscr{U}),$$

on a

$$|\text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha] - \text{Prob}[\phi \rhd Q \rightsquigarrow_S \alpha]| = 0$$

[Étape d'induction :] supposons que pour tout ψ_1, ψ_2, R_1 et R_2 t.q.

$d_H(\psi_1 \rhd R_1, \alpha) < n$ et $d_H(\psi_2 \rhd R_2, \alpha) < n$, on a $(\psi_1 \rhd R_1, \psi_2 \rhd R_2) \in \Re \Rightarrow$

$$\text{Prob}[\psi_1 \rhd R_1 \rightsquigarrow_S \alpha] = \text{Prob}[\psi_2 \rhd R_2 \rightsquigarrow_S \alpha].$$

Maintenant, si $\max(d_H(\phi \rhd P, \alpha, S) = d_H(\phi \rhd Q, \alpha, S)) = n$ $(n > 0)$, on a

$$\begin{aligned}
\text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha] &= \mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S/H}, \mathcal{C}onf) \\
&= \mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_{S}, \mathcal{C}onf) \\
&\quad + \sum_{\beta \in H,\ \psi_1 \rhd R_1 \in \mathcal{C}onf} \mu(\phi \rhd P, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \{\psi_1 \rhd R_1\}) \\
&\qquad \times \text{Prob}[\psi_1 \rhd R_1 \rightsquigarrow_S \alpha] \\
&= \sum_{\mathscr{U} \in \mathcal{C}onf/\Re} \mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathscr{U}) \\
&\quad + \sum_{\mathscr{U} \in \mathcal{C}onf/\Re} \sum_{\beta \in H,\ \psi_1 \rhd R_1 \in \mathscr{U}} \mu(\phi \rhd P, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \{\psi_1 \rhd R_1\}) \\
&\qquad \times \text{Prob}[\psi_1 \rhd R_1 \rightsquigarrow_S \alpha].
\end{aligned}$$

De même,

$$\begin{aligned}
\text{Prob}[\phi \rhd Q \rightsquigarrow_S \alpha] &= \sum_{\mathscr{U} \in \mathcal{C}onf/\Re} \mu(\phi \rhd Q, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathscr{U}) \\
&\quad + \sum_{\mathscr{U} \in \mathcal{C}onf/\Re} \sum_{\beta \in H,\ \psi_2 \rhd R_2 \in \mathscr{U}} \mu(\phi \rhd Q, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \{\psi_2 \rhd R_2\}) \\
&\qquad \times \text{Prob}[\psi_2 \rhd R_2 \rightsquigarrow_S \alpha].
\end{aligned}$$

Soit $\psi_{\mathscr{U}} \rhd R_{\mathscr{U}}$, un représentant de la classe d'équivalence $\mathscr{U}$. Par hypothèse d'induction, nous avons :

$$\forall \psi \rhd R \in \mathscr{U}.\text{Prob}[\psi \rhd R \rightsquigarrow_S \alpha] = \text{Prob}[\psi_{\mathscr{U}} \rhd R_{\mathscr{U}} \rightsquigarrow_S \alpha].$$

Ainsi, on a :

$$\begin{aligned}\text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha] &= \sum_{\mathscr{U} \in Conf/\Re} \mu(\phi \rhd P, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathscr{U}) \\ &\quad + \sum_{\mathscr{U} \in Conf/\Re} \sum_{\beta \in H} \mu(\phi \rhd P, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \mathscr{U}) \cdot \text{Prob}[\psi_{\mathscr{U}} \rhd R_{\mathscr{U}} \rightsquigarrow_S \alpha],\end{aligned}$$

et

$$\begin{aligned}\text{Prob}[\phi \rhd Q \rightsquigarrow_S \alpha] &= \sum_{\mathscr{U} \in Conf/\Re} \mu(\phi \rhd Q, \stackrel{\hat{\alpha}}{\Longrightarrow}_S, \mathscr{U}) \\ &\quad + \sum_{\mathscr{U} \in Conf/\Re} \sum_{\beta \in H} \mu(\phi \rhd Q, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \mathscr{U}) \cdot \text{Prob}[\psi_{\mathscr{U}} \rhd R_{\mathscr{U}} \rightsquigarrow_S \alpha].\end{aligned}$$

Puisque

$$P \approx_\phi Q \Rightarrow \forall \beta \in \mathcal{A}ct.\, \forall \mathscr{U} \in Conf/\Re, \mu(\phi \rhd P, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \mathscr{U}) = \mu(\phi \rhd Q, \stackrel{\hat{\beta}}{\Longrightarrow}_S, \mathscr{U})$$

on a $\text{Prob}[\phi \rhd P \rightsquigarrow_S \alpha] = \text{Prob}[\phi \rhd Q \rightsquigarrow_S \alpha]$. ■

Enfin, pour conclure cette section, nous obtenons le résultat suivant établissant que la bisimulation raffine l'équivalence de trace asymptotique.

Théorème 5.4.2 *Si deux processus P et Q sont bisimilaires, alors ils sont trace-équivalents, i.e.,*

$$\forall_{P,Q \in \mathcal{P}roc}\, P \approx_\phi Q \Rightarrow P \simeq_\phi^{tr} Q.$$

Preuve: Similaire à celle du Théorème 5.4.1. Le lecteur le trouvera en Annexe II. ■

Nous avons finalement ce classement :

Corollaire 5.4.2 *La bisimulation raffine l'équivalence de trace qui elle même raffine l'équivalence observationnelle, i.e., nous avons les inclusions strictes suivantes :*

$$\approx_\phi \subsetneq \simeq_\phi^{tr} \subsetneq \simeq_\phi .$$

Pour démontrer l'utilité de notre approche, nous allons terminer ce chapitre par une étude de cas : l'analyse de l'anonymat dans le "Crowds protocol" de Reiter et Rubin, 1998. Nous allons donner une nouvelle caractérisation de l'*innocence probable* en termes des notions définies dans cette section. Nous allons également déduire, grâce au système de transitions engendré par la sémantique du protocole dans le modèle ProSPA, un important théorème dû à Reiter et Ruben sur une relation que doivent vérifier les paramètres du protocole pour que l'innocence probable soit assurée.

5.5 Étude de cas : anonymat dans le "Crowds protocol".

Le "Crowds protocol" (Reiter et Rubin, 1998) développé par Reiter et Ruben vise à assurer l'anonymat dans les transactions sur le Web. Pour assurer l'anonymat, ce protocole masque les communications de chaque usager en les faisant acheminer par un usager choisi de manière aléatoire à l'intérieur d'un groupe d'usagers semblables. Ainsi, même si un usager indiscret ou un membre malicieux du groupe observe un message envoyé par un usager particulier, il ne peux jamais être certain si l'usager est bien celui qui a envoyé le message ou s'il achemine simplement le message d'un autre usager.

5.5.1 Spécification du protocole

Soit N le nombre d'usagers participant au protocol, $U_H = \{U_i | 1 \leq i \leq n, \quad n < N\}$ l'ensemble de usagers honnêtes (non corrompus) et $U_E = \{U_j | n < j \leq N\}$ l'ensemble des usagers sous le contrôle de l'ennemi. Quand un usager U_i dans U_H (l'initiateur doit être un usager honnête) veut établir une connection avec un serveur web, tout en restant anonyme, son fuerteur commence le processus de création d'un chemin aléatoire jusqu'au serveur web comme suit :

- l'*initiateur* U_i choisit aléatoirement un usager U_j (*relayeur*) parmi les N membres du Crowd (y compris lui-même) et lui redirige sa requête de connexion (chiffré avec la clé symétrique K_{ij} partagée par les usagers i et j).
- Soit, avec la probabilité fixe $(1-p)$, ce dernier achemine la requête au serveur. Soit il choisit aléatoirement un nouveau relayeur U_k (avec la probabilité p) parmi les N membres (y compris lui même) et lui transmet la requête re-chiffrée avec la clé K_{jk}. Cet étape se répète jusqu'à ce que la requête soit acheminée au serveur.

Dans la spécification du "Crowds protocol" ci-dessous, nous utilisons la sommation $\sum_{i \leq n} p_i P_i$ comme abréviation du processus probabiliste défini par le produit $[\lambda(m_1, \cdots, m_k) \hookrightarrow x](\prod_{i \leq n}[x = i]P_i)$ où $\lambda(m_1, \cdots, m_k) \xrightarrow{p_i} i$ et $\sum p_i = 1$. Ce processus modélise le choix probabiliste alternatif où chaque processus P_i est choisi avec la probabilité p_i. Par exemple, prenons le cas de l'initiateur du protocole. D'après la description ci-dessus, il choisit aléatoirement un usager U_j parmi les N membres du Crowd (y compris lui-même) et lui redirige sa requête de connexion. Ce qu'on peut modéliser par la somme $\sum_{j \leq N} \frac{1}{N} \overline{c}_{ij}(req).\mathbf{0}$, où la fonction λ est la fonction *random* qui retourne i ($i \leq N$) avec la probabilité uniforme, ie $p_i = \frac{1}{N}$. Ensuite, il doit se comporter comme n'importe quel relayeur, i.e., il attend sur ces canaux d'écoute un message de la part des autres usagers ; ce qui est modélisé par le produit $\prod_{k \leq N} c_{ki}(y_k).U'_{ik}(y_k)$. Enfin, s'il reçoit le message m d'un autre usager sur le canal d'ecoute c_{ki}, il doit se

comporter comme le processus $U'_{ik}(m)$ qui, soit transmet le message au serveur avec la probabilité $1-p$, soit le fait suivre à un autre usager et revient à sa position initiale. Nous pouvons donc spécifier les usagers par des processus probabilistes suivants :

Initiateur : U_i $(i \leq n)$

$$U_i ::= (\sum_{j\leq N} \frac{1}{N}\overline{c}_{ij}(req).\mathbf{0})|\prod_{k\leq N} c_{ki}(y_k).U'_{ik}(y_k)$$

$$U'_{ik}(z) ::= ((\sum_{l\leq N} \frac{1}{N}\overline{c}_{il}(z).\mathbf{0})|c_{ki}(z').U'_{ik}(z')) +^p \overline{c}_{is}(z).\mathbf{0}$$

Relayeur honnête : U_j $(i \neq j \leq n)$

$$U_j ::= \prod_{k\leq N} c_{kj}(y_k).U'_{jk}(y_k)$$

$$U'_{jk}(z) ::= ((\sum_{l\leq N} \frac{1}{N}\overline{c}_{jl}(z).\mathbf{0})|c_{kj}(z').U'_{jk}(z')) +^p \overline{c}_{is}(z).\mathbf{0}$$

Relayeur corrompu : U_j $(n < j \leq N)$

$$U_j ::= \prod_{k\leq N} c_{kj}(y_k).\overline{pub}(ID_k).U'_{jk}(y_k)$$

$$U'_{jk}(z) ::= ((\sum_{l\leq N} \frac{1}{N}\overline{c}_{jl}(z).\mathbf{0})|c_{kj}(z').U'_{jk}(z')) +^p \overline{c}_{is}(z).\mathbf{0}$$

Serveur : $S ::= \prod_{i\leq N} c_{is}(x).\mathbf{0}$

où c_{ij} $(c_{ij} \neq c_{ji})$ est le canal de transmission des messages de l'usager U_i à l'usager U_j et $P +^p Q$ est l'abréviation de $pP + (1-p)Q$. Notons que, d'après la description du protocole, aucun usager ne peut voir une communication qui ne l'implique pas. Pour que notre spécification respecte cette description, toutes les communications doivent se faire à travers des canaux privés, c'est-à-dire des canaux restreints. C'est pourquoi, dans le cas d'un usager corrompu, nous avons ajouté l'action "décorative" $\overline{pub}(ID_k)$ qui modélise le fait que l'usager corrompu informe l'attaquant (à travers le canal public *pub*) de l'identité de celui qui lui a envoyé le message. Notons également que, dans cette spécification, nous avons omis le chiffrement dont le seul but, pour ce protocole, est d'assurer le caractère privé des communications entre les membres du Crowd. Le protocole se modélise par la composition parallèle de tous ses membres où seul le canal *pub* est public. Formellement, soit $C = \{c_{ij} : 1 \leq i,j \leq N\} \cup \{c_{is} : 1 \leq i \leq N\}$,

la spécification du protocole dans notre modèle est le processus P défini comme suit :

$$P ::= (S| \prod_{i \leq N} U_i) \backslash C$$

5.5.2 Spécification de la propriété d'anonymat

Reiter et Rubin, 1998, ont défini trois propriétés d'anonymat pour ce protocole. Celle qui nous intéresse ici, est la propriété d'*innocence probable* qui stipule que, du point de vue de l'attaquant, l'usager de qui il reçoit le message ne parait pas être initiateur du protocole plus probable qu'un simple relayeur. En d'autres termes, la probabilité que l'usager, qui lui a envoyé le message, soit effectivement l'initiateur du protocole est inférieure ou égale à $\frac{1}{2}$. Notons que seul le premier usager honnête à avoir envoyé le message à un usager corrompu peut être considéré comme étant l'initiateur probable du protocole du point de vue de l'attaquant. En effet, si un usager apparait après un usager corrompu dans la chaîne de transmission du message, alors l'attaquant est sûr qu'à cette position, il n'apparait qu'à titre de relayeur. Nous pouvons donc définir l'innocence probable comme suit :

Définition 5.5.1 : *Le "Crowds protocol" assure l'innocence probable à un initiateur si, sachant qu'il existe au moins un usager corrompu dans la chaîne de transmission du message, la probabilité totale que l'initiateur puisse être le premier à choisir comme successeur immédiat (relayeur) un usager malveillant est inférieure à* $\frac{1}{2}$.

Soit E, l'évènement *"il existe au moins un usager corrompu dans la chaîne de transmission du message"* et E_i l'évènement *"l'usager* U_i *est le premier à choisir comme successeur imédiat un usager malveillant"*. Soit U_{i_0} l'initiateur le protocole, l'innocence probable est assurée si la probabilité que l'évènement E_{i_0} soit vrai, sachant que E est vrai, est inférieure à $\frac{1}{2}$, ie $\text{Prob}[E_{i_0}|E] \leq \frac{1}{2}$. Or $\text{Prob}[E_{i_0}|E] = \frac{\text{Prob}[E_{i_0} \vee E]}{\text{Prob}[E]}$

et, puisque $E_{i_0} \Rightarrow E$, nous avons $\text{Prob}[E_{i_0}|E] = \frac{\text{Prob}[E_{i_0}]}{\text{Prob}[E]}$. Puisque nous nous intéressons uniquement au premier usager à avoir transmis le message à un usager corrompu, nous avons $E = \bigcup_{1 \leq i \leq n} E_i$. Or les évènements E_i sont mutuellement exclusifs, donc $\text{Prob}[E] = \sum_{1 \leq i \leq n} \text{Prob}[E_i]$. Enfin, soit $\text{Prob}[\phi \rhd P \rightsquigarrow_S E_i]$ la probabilité que sous l'ordonnanceur S, le protocole engendre l'évènement E_i sous l'environnement de connaissance initiale ϕ. Nous avons le résultat suivant :

Proposition 5.5.1 $\forall_{i \in [1..n], S \in Sched} \text{Prob}[\phi \rhd P \rightsquigarrow_S E_i] = \mu(\phi \rhd P, \overset{\overline{pub}(\hat{ID_i})}{\Longrightarrow}_S, Conf)$.

La preuve de cette proposition découle tout simplement du fait que les seules actions visibles sont les actions de la forme $\overline{pub}(\hat{ID_i})$ et du fait que les évènements E_i sont mutuellement exclusifs et que $\text{Prob}[\phi \rhd P \rightsquigarrow_S E_i] = \text{Prob}[\phi \rhd P \rightsquigarrow_S \overline{pub}(ID_i)]$. Avec ce résultat, nous pouvons donc spécifier la propriété d'innocence probable grâce aux probabilités cumulatives. Ainsi, nous avons le théorème suivant :

Théorème 5.5.1 *Le protocole assure l'innocence probable de l'usager U_{i_0} ssi*

$$\forall S \in Sched. \frac{\mu(\phi \rhd P, \overset{\overline{pub}(\hat{ID_{i_0}})}{\Longrightarrow}_S, Conf)}{\sum_{1 \leq i \leq n} \mu(\phi \rhd P, \overset{\overline{pub}(\hat{ID_i})}{\Longrightarrow}_S, Conf)} \leq \frac{1}{2}$$

À première vue, on pourrait penser que prouver que l'innocence probable est assurée par le protocole est une tâche ardue, puisqu'il faut calculer les différentes probabilités pour tous les ordonnanceurs et il pourrait y en avoir une infinité. Heureusement, notre spécification du "Crowds protocol" est insensible à l'ordonnancement, car il n'y a qu'un seul canal public et les actions sur ce canal ne sont que des actions d'ouput. De plus, il est facile de constater qu'il n'y a qu'une seule action pour chaque étape d'exécution du protocole. Enfin, signalons que la connaissance initiale de l'intrus ϕ a peu d'importance car c'est un attaquant passif.

Avant de calculer formellement la valeur de cette probabilité cumulative, nous allons exploiter quelques propriétés du protocole pour simplifier le calcul. En effet, tous les usagers honnêtes jouent des rôles similaires. Nous pouvons donc les rassembler pour former un seul relayeur honnête en redistribuant correctement les probabilités. Nous pouvons faire de même pour les usagers corrompus. Enfin, nous pouvons modifier la spécification des usagers corrompus sans affecter la propriété. En effet, nous pouvons permettre aux usagers corrompus de transmettre leur message au serveur puisque le relayer à un autre usager ne leur apporte rien d'intéressant sur la nature de l'identité de l'initiateur. Nous pouvons donc spécifier les différents agents comme suit :

Initiateur : I

$$I_0 ::= \frac{n}{N}\overline{c}_{ih}(req).\mathbf{0} + \frac{N-n}{N}\overline{c}_{ie}(req).\mathbf{0}$$

Relayeur honnête : H

$$H ::= c_{ih}(x).Q_h(x)$$

où $\quad Q_h(x) ::= R_h(x)|c_{hh}(z).Q_h(z)$ et

$$R_h(x) ::= (1-p)\overline{c}_{hs}(x).\mathbf{0} + \frac{p \times n}{N}\overline{c}_{hh}(x).\mathbf{0} + \frac{p(N-n)}{N}(\frac{1}{n}\overline{c}_{ie}(x).\mathbf{0} + \frac{n-1}{n}\overline{c}_{he}(x).\mathbf{0})$$

Relayeur corrompu : E

$$E ::= c_{ie}(x).\overline{pub}(ID_i).\overline{c}_{es}(x)|c_{he}(y).\overline{pub}(ID_h).\overline{c}_{es}(y)$$

Serveur : $S ::= c_{is}(x).\mathbf{0}|c_{hs}(y).\mathbf{0}|c_{es}(z).\mathbf{0}$

La nouvelle spécification du protocole est ainsi obtenue par la compostion parallèle

$$P = (I_0|H|E|S)\backslash C'$$

avec $C' = \{c_{xy} : x, y \in \{i, h, e, s\}\}$ dont le système de transitions est donné par la Figure 5.2. Les transitions étiquetées par τ représentent les actions internes du système tandis que les deux transitions étiquetées par α et β représentent respectivement les deux actions publiques $\overline{pub}(ID_i)$ et $\overline{pub}(ID_h)$ par lesquelles l'usager corrompu informe l'attaquant de l'identité de celui qui lui a transmis le message.

Nous pouvons maintenant calculer, à partir du système de transitions engendré par

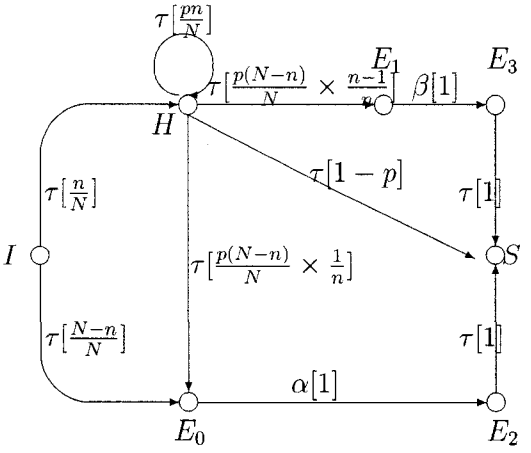

FIGURE 5.2 *Système "Crowds"*

la sémantique du protocole, les probabilités

$$\mu(\emptyset \rhd P, \overset{\overline{pub}(\hat{ID}_i)}{\Longrightarrow}_S, \mathcal{C}onf) \text{ et } \mu(\emptyset \rhd P, \overset{\overline{pub}(\hat{ID}_h)}{\Longrightarrow}_S, \mathcal{C}onf).$$

En effet, d'après la Figure 5.2, nous avons $\forall S \in \mathit{Sched}$,

$$\begin{aligned}
\mu(\emptyset \rhd P, \overset{\overline{pub}(\hat{ID}_i)}{\Longrightarrow}_S, \mathcal{C}onf) &= \tfrac{N-n}{N} \times 1 + \tfrac{n}{N} \times (\textstyle\sum_{i\geq 0}(\tfrac{p}{N})^i) \times \tfrac{p(N-n)}{N} \times \tfrac{1}{n} \times 1 \\
&= \tfrac{N-n}{N}(1 + \tfrac{p}{N-pn}) \\
\mu(\emptyset \rhd P, \overset{\overline{pub}(\hat{ID}_h)}{\Longrightarrow}_S, \mathcal{C}onf) &= \tfrac{N-n}{N} \times 1 + \tfrac{n}{N} \times (\textstyle\sum_{i\geq 0}(\tfrac{p}{N})^i) \times \tfrac{p(N-n)}{N} \times \tfrac{1}{n} \times 1 + \\
&\quad \tfrac{n}{N} \times (\textstyle\sum_{i\geq 0}(\tfrac{p}{N})^i) \times \tfrac{p(N-n)}{N} \times \tfrac{n-1}{n} \times 1 \\
&= \tfrac{N-n}{N} \times \tfrac{p(n-1)}{(N-pn)}
\end{aligned}$$

D'après le théorème 5.5.1, nous obtenons le résultat suivant :

Lemme 5.5.1 *Le protocole Crowds assure l'innocence probable ssi*

$$\frac{N + p(1-n)}{N} \leq \frac{1}{2}$$

Nous pouvons ainsi dériver cet important résultat dû à Reiter et Rubin, 1998 :

Théorème 5.5.2 *Soit $p > \frac{1}{2}$ la probabilité de relayer. Si le nombre total d'usagers corrompus, dénoté par c, est tel que $N \geq \frac{p(c+1)}{p-\frac{1}{2}}$, alors le protocole assure l'innocence probable de l'initiateur.*

Preuve: Puisque n est le nombre d'usagers honnêtes, nous avons $c = N - n$ et donc $\frac{p(c+1)}{p-\frac{1}{2}} = \frac{p(N-n+1)}{p-\frac{1}{2}}$. Puisque $p > \frac{1}{2}$ nous avons

$$\begin{aligned}
\tfrac{p(c+1)}{p-\frac{1}{2}} \leq N \quad &\Rightarrow \quad p(c+1) \leq (p-\tfrac{1}{2})N \\
&\Rightarrow \quad p(N-n+1) \leq (p-\tfrac{1}{2})N \\
&\Rightarrow \quad p(1-n) \leq \tfrac{-1}{2}N \\
&\Rightarrow \quad N + p(1-n) \leq \tfrac{1}{2}N \\
&\Rightarrow \quad \tfrac{N+p(1-n)}{N} \leq \tfrac{1}{2}.
\end{aligned}$$

Le Lemme 5.5.1 permet de conclure. ■

CONCLUSION

Interférence admissible

L'interférence admissible est une notion très générale qui apparaît être une primitive très naturelle à partir de laquelle il est possible d'exprimer la plupart des requis des systèmes à flux d'information admissible. Toutefois les modèles proposés jusqu'à présent, pour l'analyse des protocoles cryptographiques dans le cadre de la théorie du flux d'information admissible, se heurtent à un problème majeur : l'obligation de quantifier sur tous les attaquants possibles. Par conséquent, l'interférence admissible, telle que définie dans ces modèles, n'est pas très pratique comme moyen de vérification.

Modèle CSPAD. Pour résoudre ce problème, nous avons proposé un modèle de spécification des protocoles cryptographiques très simple dénommé *Cryptographic Security Process Algebra with Downgrading* ou CSPAD en abrégé. C'est une algèbre de processus à la CCS (Milner, 1989) avec passage de paramètres par valeur qui étend l'algèbre de processus SPA (Focardi et Gorrieri, 2001) syntaxiquement, pour prendre explicitement en compte la spécification des primitives cryptographiques, et sémantiquement, pour prendre en compte la spécification des niveaux de sécurité avec mécanisme de déclassification. Nous y avons défini de nouvelles caractérisations de l'interférence admissible grâce à une représentation symbolique (donc finie) des attaquants. Le problème d'analyse se réduit alors à un problème d'équivalence checking de systèmes infinis. Ces équivalences sont définies sur un système de transitions enrichi dont les transitions sont contraintes par la connaissance acquise par l'environnement, c'est-à-dire des attaquants éventuels, à la réception de chaque message du protocole. On a défini sur ce système de transitions enrichi une équivalence de trace et une bisimulation faible à partir desquelles nous avons défini nos nouvelles caractérisations de l'interférence admissible qui évitent de quantifier sur tous les attaquants. Le problème

d'analyse est ainsi un problème semi-décidable. Les principales sources d'indécidabilité proviennent du caractère non borné des paramètres du système à vérifier (nombre de sessions, nombre des participants, tailles des messages, etc.). Cette dernière difficulté peut être surmontée en adoptant l'idée de la réduction symbolique que nous avons proposé dans (Bastien *et al.*, 2006) ou celle proposée dans (Lafrance et Mullins, 2003a;Lafrance, 2004). Bien que dans cette thèse nous nous sommes limités aux requis de secret et de correspondance qui sont des propriétés de sûreté, notre approche peut être facilement étendue pour prendre en compte des propriétés de vivacité, telle que l'équité, grâce aux caractérisations de l'interférence admissible basées sur la bisimulation proposées dans cette thèse.

Vérification. Notre méthode a été implémenté dans un prototype d'analyseur symbolique des protocoles cryptographiques ASPiC (Bastien, 2004;Bastien *et al.*, 2006) qui nous a permis d'analyser la plupart des protocoles de la librairie Clark et Jacob, 1997, et surtout, de détecter une faille de sécurité dans le protocole *SET* (Brlek *et al.*, 2006a) d'e-commerce de Visa et Mastercard. Le fait que l'article (Brlek *et al.*, 2006a) tienne le premier rang au palmarès des articles de la revue scientifique Information Processing Letters les plus téléchargés dès sa parution, en début de l'année 2006, et continue de figurer dans le top 25 de ce palmarès presque deux ans après sa parution, témoigne de l'intérêt qu'une telle approche suscite au sein de la communauté scientifique internationale.

Modèle probabiliste

Toutefois, notre modèle CSPAD, ainsi que la plupart des méthodes formelles existantes de la sécurité de l'information développées quand nous avons débuté ce projet, est basée, d'une part sur l'approche dite "possibiliste" dans laquelle le non-déterminisme sert à modéliser les mécanismes aléatoires de génération de tous les

comportements possibles des systèmes analysés et, d'autre part sur l'hypothèse de cryptographie parfaite qui stipule que les primitives cryptographiques sont des boites noires. Ces approches sont trop grossières pour décrire les fuites d'information probabilistes et ainsi, prévenir les attaques basées sur les distributions probabilistes des évènements observables du système.

Modèle ProSPA. Aussi, nous avons proposé un nouveau modèle probabiliste polynomial dénommé ProSPA (*Probabilistic Security Process Algebra*). C'est un modèle qui étend aussi bien syntaxiquement que sémantiquement le modèle CSPAD pour prendre en compte la spécification des systèmes de sécurité probabilistes utilisant des primitives cryptographiques (probabilistes) polynomiales et l'analyse de tels systèmes dans un environnement hostile. Nous avons montré que la sémantique de notre modèle reflète adéquatement la capacité de l'intrus à contrôler le réseau de communication sans pour autant contrôler la réaction interne du système avec lequel il interagit. En vue de vérifier des propriétés de sécurité, nous avons défini deux équivalences asymptotiques. La première est une reformulation de *l'équivalence observationnelle asymptotique* proposée dans (Mitchell *et al.*, 2006) dans le cadre de ProSPA. Constatant qu'elle n'est pas suffisamment adaptée pour nos besoins de vérification, nous l'avons reformulé pour prendre en compte toute trace observable au lieu d'une observation à la fois.

Cependant, comme la plupart des problèmes d'équivalence checking pour des modèles de calcul à base d'algèbres de processus, nos équivalences asymptotiques souffrent du problème de quantification sur tous les intrus possibles. S'inspirant des techniques utilisées pour le modèle possibiliste CSPAD, nous avons défini une sémantique contextuelle pour notre modèle probabiliste. Cette sémantique, en plus de nous permettre d'éviter de quantifier sur tous les attaquants possibles, nous a énormément simplifié la tâche au moment de définir les classes d'actions stratégiquement équiprobables et surtout d'avoir une définition simple d'ordonnanceurs et uniforme. Par uniforme, nous voulons dire applicable aussi bien aux processus bloqués qu'aux processus non

bloqués. Nous avons reformulé nos équivalences asymptotiques, conformément à cette nouvelle sémantique et avons prouvé que, contrairement à la sémantique concrète, elles ne sont pas équivalentes ; l'équivalence observationnelle étant moins fine, justifiant le passage à l'équivalence de trace. Nous avons également défini, dans cette nouvelle sémantique, une bisimulation faible qui raffine les équivalences asymptotiques.

Vérification. Enfin, pour démonter l'utilité de ce modèle, nous avons vérifié deux importants protocoles probabilistes : le protocole "le Dîner des Cryptographes" (Chaum, 1988) et le "Crowds protocol" (Reiter et Rubin, 1998). Dans le premier cas, nous avons proposé une version probabiliste de la spécification de la propriété d'*anonymat*, due à Schneider et Sidiropoulos, 1996, et une spécification du protocole comportant une faille de sécurité et avons montré qu'une sémantique qui privilégie les actions internes, telle que celle proposée dans l'article (Mitchell *et al.*, 2006), ne peut pas détecter cette faille alors que la notre la détecte. Nous avons ensuite montré que, dans une implémentation idéale du protocole, l'anonymat du payeur ne dépend que du "fairness" des jetons. Nous avons également montré que, même si les jetons ne sont pas biaisés, une implémentation réelle du protocole peut ne pas être sécuritaire, due à certaines propriétés mathématiques des crypto-systèmes utilisés ; ce qui montre l'importance d'avoir des modèles qui vont au-delà des hypothèses traditionnelles à la Dolev-Yao. Dans le deuxième cas, l'analyse nous a permis de donner une nouvelle caractérisation de l'*innocence probable* en termes de probabilités cumulatives et de déduire, grâce au système de transitions engendré par la sémantique du protocole, un important théorème dû à Reiter et Ruben sur une relation que doivent vérifier les paramètres du protocole, pour que l'innocence probable soit assurée. Bien que n'ayant pas nécessité l'utilisation des équivalences contextuelles, cette analyse donne un aperçu de toutes les belles choses que l'on peut faire dans ce modèle. Le lecteur intéressé trouvera dans l'article (Brlek *et al.*, 2007a) une étude du protocole DCP basée sur la bisimulation contextuelle.

Perpectives

Flux d'information probabiliste. Bien que, n'ayant pas encore totalement exploré les conséquences des nos définitions des équivalences probabilistes, nous pensons qu'elles permettront d'établir de nouvelles caractérisations importantes des propriétés de flux d'information probabiliste qui exigent que les distributions probabilistes des événements observables d'un système ne soient pas altérées par n'importe quel intrus. On serait tenté de croire que les techniques présentées dans la première partie de cette thèse (Chapitres 2 et 3), pour les modèles possibilistes, sont transposables de manière naturelle au modèle computationnel ProSPA. Malheureusement, la réponse est négative. En effet, les caractérisations des propriétés de flux d'information pour les modèles possibilistes sont essentiellement basées sur le fait qu'en masquant et en supprimant les actions de haut niveau d'un processus, nous obtenons deux processus observationnellement équivalents. Or ces deux opérations n'affectent pas l'ensemble des actions exécutables à un état de la même manière. La première remplace certaines actions visibles par une action interne alors que la seconde les supprime. Puisque la stratégie d'attaque, i.e. les ordonnanceurs, dépend de l'ensemble des actions exécutables, l'attaquant pourrait utiliser cette différence pour distinguer les deux processus. Il serait donc nécessaire de développer des nouvelles caractérisations de la non-interférence et de l'interférence admissible probabilistes adaptées aux particularités des modèles computationnels. A côté de cela, il serait également important d'étendre le champs des systèmes de sécurité à vérifier pour inclure une émergente et non moins importante catégorie des systèmes de sécurité : *les protocoles stéganographiques.*

Protocoles stéganographiques. L'art de la dissimulation est à la stéganographie ce que celui du secret est à la cryptographie. L'analyse des systèmes stéganographiques est encore peu développée. Si la cryptographie est devenue un champ de recherche de pointe, la stéganographie est émergente. Encore peu médiatisées, les applications de la stéganographie n'en sont pas moins spectaculaires et redoutables dans le contexte

de la mondialisation des économies, des tensions géo-politiques, de l'expansion de l'Internet, de l'espionnage industriel, du piratage ou encore de la protection de droit d'auteur par "filigrane" (watermarking). Il devient donc important de développer des méthodes d'analyse et de détection fiables. Toutes les méthodes d'analyse développées à ce jour sont basées sur la théorie de l'information. La sécurité des systèmes de sécurité peut être classée selon la capacité de calcul de l'adversaire. La sécurité des systèmes contre des attaquants de capacité de calcul "raisonnablement" restreinte est dite *sécurité computationnelle*, alors que celle ne faisant aucune hypothèse sur leur capacité de calcul est dite *sécurité au sens de la théorie d'information.* Or la sécurité de la plupart des systèmes de sécurité pratiques, en particulier les protocoles stéganographiques, est une sécurité computationnelle, puisqu'ils doivent être robustes contre tout attaquant de capacité de calcul en temps polynomial. Nous pensons que la théorie de flux d'information que nous avons abordée dans la première partie de cette thèse, couplée à un modèle computationnel comme le modèle ProSPA, serait parfaitement adaptée pour exprimer les propriétés de sécurité des protocoles stéganographiques pour lesquels l'observation d'un stégo-message au niveau public ne doit en aucun cas révéler l'existence du message dissimulé.

Toutefois, l'analyse des protocoles stéganographiques s'avère plus complexe que celle des crypto-protocoles car un stégo-système peut être vu comme un crypto-système avec l'hypothèse supplémentaire que l'observateur connaît, à priori, la distribution probabiliste des variables aléatoires des stéganogrammes. De plus, l'adversaire n'a ni le même rôle ni la même puissance de nuisance que celui des crypto-protocoles. Il s'agira donc de développer une nouvelle caractérisation de la non-interférence probabiliste capable de détecter toute fuite d'information dans ces systèmes en tenant compte de leur spécificité. Une autre spécificité des protocoles stéganographiques est que, dans le cas de la détection des stégo-messages, le protocole analysé ne remplit pas le "principe de Kerckhoffs" (généralement admis pour les crypto-protocoles) qui stipule que le protocole est du domaine public : seules les clés doivent être se-

crètes. En effet, un observateur qui intercepte une communication et qui voudrait vérifier si cette dernière contient ou non un stégo-message, ne connaît pas, à priori, le stégo-système éventuellement utilisé pour dissimuler le message. On est dans ce cas devant un problème de test d'hypothèse. Il est donc important de trouver une nouvelle propriété computationnelle de flux d'information correspondant aux bornes Bayesiennes (Chatzikokolakis *et al.*, 2007) de la théorie d'information qui minimise la probabilité d'erreur. Enfin, signalons que la quantification du flux d'information à été abordée dans des modèles formels (Clark *et al.*, 2005), mais à notre connaissance aucun résultat de ce type n'existe dans un modèle computationnel. Il s'agira donc d'étendre ces résultats aux modèles computationnels prenant en compte les spécificités des stégo-systèmes.

RÉFÉRENCES

ABADI, M. (1999). Secrecy by typing in security protocols. *Journal of the ACM*, 46, 749–786.

ABADI, M. ET CORTIER, V. (2006). Deciding knowledge in security protocols under equational theories. *Theoretical Computer Science*. To appear.

ABADI, M. ET ROGAWAY, P. (2000). Reconciling two views of cryptography (the computational soundness of formal encryption). *TCS '00 : Proceedings of the International Conference IFIP on Theoretical Computer Science, Exploring New Frontiers of Theoretical Informatics*. Springer-Verlag, London, UK, 3–22.

ADÃO, P., BANA, G., HERZOG, J. ET SCEDROV, A. (2005). Soundness of formal encryption in the presence of key-cycles. S. D. C. di Vimercati, P. F. Syverson et D. Gollmann, éditeurs, *ESORICS*. Springer, vol. 3679 de *Lecture Notes in Computer Science*, 374–396.

ALDINI, A., BRAVETTI, M. ET GORRIERI, R. (2002). A process algebra approach for the analysis of probabilistic non-interference. Rapport technique.

ATALLAH, M. (1999). *Algorithms and Theory of Computation Handbook*. CRC Press LLC.

BACKES, M. ET PFITZMANN, B. (2003). Intransitive non-interference for cryptographic purpose. *IEEE Symposium on Security and Privacy*. IEEE Computer Society, 140–.

BASTIEN, G. (2004). *ASPiC : un outil d'analyse symbolique automatisée de protocoles cryptographiques basé sur le modèle de flux d'information*. Mémoire de maîtrise, École Polytechnique de Montréal.

BASTIEN, G., HAMADOU, S. ET MULLINS, J. (2006). Aspic : A tool for symbolic analysis of cryptographic protocols. *Annals of Telecommunications (Submited for publication)*, 1–28.

BELLARE, M., DESAI, A., JOKIPII, E. ET ROGAWAY, P. (1997). A concrete security treatment of symmetric encryption. *FOCS*. 394–403.

BELLARE, M., KILIAN, J. ET ROGAWAY, P. (1994). The security of cipher block chaining. Y. Desmedt, éditeur, *CRYPTO*. Springer, vol. 839 de *Lecture Notes in Computer Science*, 341–358.

BELLARE, M. ET ROGAWAY, P. (1994). Entity authentication and key distribution. *CRYPTO '93 : Proceedings of the 13th annual international cryptology conference on Advances in cryptology*. Springer-Verlag New York, Inc., New York, NY, USA, 232–249.

BENGT, J., LARSON, K. ET YI, W. (2002). Probabilistic extension of process algebra. *Handbook of process algebra*. 565.

BIRD, R., GOPAL, I., HERZBERG, A., JANSON, P., KUTTEN, S., MOLVA, R. ET YUNG, M. (1991). Systematic design of two-party authentication protocols. *Lecture Notes in Computer Science*, 576, 44–??

BLANCHET, B. (2006). A computationally sound mechanized prover for security protocols. *S&P*. IEEE Computer Society, 140–154.

BOREALE, M. ET BUSCEMI, M. (2003). Symbolic analysis of crypto-protocols based on modular exponentiation. *Proceedings of 28th International Symposium Mathematical Foundations of Computer Science (MFCS'03)*. Springer-Verlag, Bratislavia, Slovac Republic, vol. 2747 de *LNCS*, 269–278.

BOREALE, M., NICOLA, R. D. ET PUGLIESE, R. (1999). Proof techniques for cryptographic processes. *Logic in Computer Science.* 157–166.

BOSSI, A., PIAZZA, C. ET ROSSI, S. (2004). Modelling downgrading in information flow security. *CSFW '04 : Proceedings of the 17th IEEE workshop on Computer Security Foundations.* IEEE Computer Society, Washington, DC, USA, 187.

BREMAUD, P. (1999). Markov Chains : Gibbs Fields, Monte Carlo Simulation, and Queues. Springer-Verlag, New-York, N.Y.

BRESSE, P., D'AUGÈRES, G. B. ET THUILLIER, S. (1997). *Paiement Numérique sur Internet.* International THOMSON Publishing.

BRLEK, S., HAMADOU, S. ET MULLINS, J. (2003). Aset, un protocole anonyme et sécuritaire pour les transactions électroniques. *Actes de la 2ème conférence francophone sur Sécurité et Architecture Réseaux (SAR'03).* Nancy, France, 1–15.

BRLEK, S., HAMADOU, S. ET MULLINS, J. (2005). Anonymous and secure electronic transaction protocol. *Annals of Telecommunications*, 60, 530–557.

BRLEK, S., HAMADOU, S. ET MULLINS, J. (2006a). A flaw in the electronic commerce protocol SET. *Inf. Process. Lett.*, 97, 104–108.

BRLEK, S., HAMADOU, S. ET MULLINS, J. (2006b). Some remarks on the certificates registration of the electronic commerce protocol SET. *AICT/ICIW.* IEEE Computer Society, 119.

BRLEK, S., HAMADOU, S. ET MULLINS, J. (2007a). A probabilistic process algebra for cryptographic protocols. *27th Foundations of Software Technology and Theoretical Computer Science (FSTTCS'07).* (submitted for publication), 2-14 December, New Delhi, India.

BRLEK, S., HAMADOU, S. ET MULLINS, J. (2007b). A probabilistic scheduler for the analysis of cryptographic protocols. *Electronic Notes in Theoretical Computer Science*, 194, 61–83.

BURROWS, M., ABADI, M. ET NEEDHAM, R. (1990). A logic of authentication. *ACM Transactions on Computer Systems*, 8, 18–36.

CANETTI, R., CHEUNG, L., KAYNAR, D. K., LISKOV, M., LYNCH, N. A., PEREIRA, O. ET SEGALA, R. (2006). Time-bounded task-pioas : A framework for analyzing security protocols. S. Dolev, éditeur, *DISC*. Springer, vol. 4167 de *Lecture Notes in Computer Science*, 238–253.

CHATZIKOKOLAKIS, K. ET PALAMIDESSI, C. (2005). A Framework for Analysing Probabilistic Protocols and its Applications to the Partial Secrets Exchange. *Proc. of the Sym. on Trust. Glob. Comp. (STGC'05)*. Spr.-Ver., LNCS.

CHATZIKOKOLAKIS, K. ET PALAMIDESSI, C. (2007). Making random choices invisible to the scheduler. *Proc. of CONCUR'07. To appear.*

CHATZIKOKOLAKIS, K., PALAMIDESSI, C. ET PANANGADEN, P. (2007). Probability of error in information-hiding protocols. *Proc. of the 20th IEEE CSF.*

CHAUM, D. (1988). The dining cryptographers problem : Unconditional sender and recipient untraceability. *J. Cryptology*, 1, 65–75.

CHEVALIER, Y., KÜSTERS, R., RUSINOWITCH, M. ET TURUANI, M. (2003). Deciding the Security of Protocols with Diffie-Hellman Exponentiation and Products in Exponents. *Proceedings of the Foundations of Software Technology and Theoretical Computer Science, FSTTCS'03*. Springer, Lecture Notes in Computer Science. Long version available as Christian-Albrecht Universität IFI-Report 0305, Kiel (Germany).

CHEVALIER, Y., KÜSTERS, R., RUSINOWITCH, M. ET TURUANI, M. (2005). Deciding the security of protocols with commuting public key encryption. *Workshop on Automated Reasoning for Security Protocol Analysis, ARSPA'2004*. Cork, Ireland, vol. 125 de *Electronic Notes in Theoretical Computer Science - ENTCS*, 55–66.

CLARK, D., HUNT, S. ET MALACARIA, P. (2005). Quantified interference for a while language. *Electr. Notes Theor. Comput. Sci.*, 112, 149–166.

CLARK, J. ET JACOB, J. (1997). A survey of authentication protocol litterature. URL = http ://www.cs.york.ac.uk/ jac/papers/.

COMON-LUNDH, H. ET CORTIER, V. (2003). New decidability results for fragments of first-order logic and application to cryptographic protocols. *Proceedings 14th International Conference on Rewriting Techniques and Applications (RTA'03)*. Springer-Verlag, Valencia, Spain, vol. 2706 de *Lecture Notes in Computer Science*, 148–164.

COMON-LUNDH, H. ET TREINEN, R. (2003). Easy intruder deductions. *In Verification : Theory and Practice, Essays Dedicated to Zohar Manna on the Occasion of His 64th Birthday*. Springer-Verlag, vol. 2772 de *Lecture Notes in Computer Science*, 225–242.

CORTIER, V., DELAUNE, S. ET LAFOURCADE, P. (2006). A Survey of Algebraic Properties Used in Cryptographic Protocols. *Journal of Computer Security*, 14, 1–43.

CORTIER, V. ET WARINSCHI, B. (2005). Computationally sound, automated proofs for security protocols. (Sagiv, 2005), 157–171, 157–171.

DELAUNE, S. ET JACQUEMARD, F. (2004). A decision procedure for the verification of security protocols with explicit destructors.

DESHARNAIS, J., GUPTA, V., JAGADEESAN, R. ET PANANGADEN, P. (2003). Approximating labelled Markov Proccesses. vol. 184 de *Information and Computation*, 160–200.

DIFFIE, W., VAN OORSCHOT, P. C. ET WIENER, M. J. (1992a). Authentication and authenticated key exchanges. *Designs, Codes and Cryptography*, 2, 107–125.

DIFFIE, W., VAN OORSCHOT, P. C. ET WIENER, M. J. (1992b). Authentication and authenticated key exchanges. *Designs, Codes and Cryptography*, 2, 107–125.

DOLEV, D. ET YAO, A. (1983). On the security of public key protocols. *IEEE Transactions on Information Theory*, 29, 198–208.

EVEN, S., GOLDREICH, O. ET SHAMIR, A. (1985). On the security of pig-pong protocols when implemented using RSA. *Proc. Advences in Cryptology (CRYPTO'85.* Springer-Verlag, Santa Barbara, Calfornia USA, vol. 218 de *LNCS*, 58–72.

FABREGA, F. J. T., HERZOG, J. ET GUTTMAN, J. (1998a). Strand space pictures. *Proc. of the Workshop on Formal Methods and Security Protocols.* Version available at `http ://www.cs.bell-labs.com/who/nch/fmsp/program.html`.

FABREGA, F. J. T., HERZOG, J. ET GUTTMAN, J. (1998b). Strand spaces : Why is a security protocol correct ? *Proc. of the 1998 IEEE Symposium on Security and Privacy.* IEEE Computer Society, 160–171.

FEINBURG, E., SCHWARTZ, K. ET YI, W. (2001). Handbook of Markov Decision processes, Methods and Applications. *Kluwer.*

FOCARDI, R. (2001). Classification of security properties (part II : Network security). slides. Second Internation School on Foundations Of Security Analysis and Design (FOSAD'01).

FOCARDI, R. ET GORRIERI, R. (2001). Classification of security properties (part i : Information flow). *Foundations of Security Analysis and Design.* Springer-Vale, vol. 2171 de *LNCS*, 331–396.

GARCIA, F. D., VAN ROSSUM, P. ET SOKOLOVA, A. (2007). Probabilistic anonymity and admissible schedulers. *http ://arxiv.org/abs/0706.1019.*

GARFINKEL, S. ET SPAFFORD, G. (2001). *Web Security & Commerce.* Cambridge, MA : O'Reilly and Assoc.

GOGUEN, J. A. ET MESEGUER, J. (1982). Security policies and security models. *IEEE Symposium on Security and Privacy.* 11–20.

GOGUEN, J. A. ET MESEGUER, J. (1984). Unwinding and inference control. *IEEE Symposium on Security and Privacy.* 75–87.

GOLDREICH, O., MICALI, S. ET WIGDERSON, A. (1991). Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *J. ACM*, 38, 691–729.

GOLDWASSER, S. ET MICALI, S. (1984). Probabilistic encryption. *J. Comput. Syst. Sci.*, 28, 270–299.

GOLDWASSER, S., MICALI, S. ET RIVEST, R. L. (1988). A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17, 281–308.

GORDON, A. D. ET JEFFREY, A. (2002). Typing one-to-one and one-to-many correspondences in security protocols. M. Okada, B. C. Pierce, A. Scedrov, H. Tokuda et A. Yonezawa, éditeurs, *ISSS.* Springer, vol. 2609 de *Lecture Notes in Computer Science*, 263–282.

GORDON, A. D. ET JEFFREY, A. (2004). Authenticity by typing for security protocols. *J. Comput. Secur.*, 11, 451–519.

HADJ-ALOUANE, N. B., LAFRANCE, S., LIN, F., MULLINS, J. ET YEDDES, M. M. (2005). On the verification of intransitive noninterference in multilevel security. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 35, 948–958.

HAIGH, J. T. ET YOUNG, W. D. (1987). Extending the noninterference version of mls for sat. *IEEE Trans. Software Eng.*, 13, 141–150.

HAMADOU, S. (2002). *Cryptographie et protocoles transactionnels.* Mémoire de maîtrise, UQAM - Université du Québec à Montréal.

HERZOG, J. (2005). A computational interpretation of dolev-yao adversaries. *Theor. Comput. Sci.*, 340, 57–81.

HOARE, C. A. R. (1978). Communicating sequential processes. *Commun. ACM*, 21, 666–677.

HUGHES, D. ET SHMATIKOV, V. (2004). Information hiding, anonymity and privacy : a modular approach. *J. Computer Security*, 12, 3–36. Version available at `http ://boole.stanford.edu/ dominic/papers.html`.

JANVIER, R., LAKHNECH, Y. ET MAZARÉ, L. (2005). Completing the picture : Soundness of formal encryption in the presence of active adversaries. (Sagiv, 2005), 172–185, 172–185.

KAPUR, D., NARENDRAN, P. ET WANG, L. (2003). An E-unification algorithm for analyzing protocols use modular exponentiation. *Proc. 14th International Conference on Rewriting Techniques and Applications (RTA'03).* Springer-Verlag, Valencia, Spain, vol. 2706 de *LNCS*, 165–179.

KOZEN, D. (2003). Semantics of probabilistic programs. Journal of Computer and Systems Sciences, vol. 22, 328–350.

KREMER, S. ET RYAN, M. (2005). Analysing the vulnerability of protocols to produce known-pair and choosen text attacks. *Proc. 2nd International Workshop on Security Issues in Coordination Models, Languages and Systems (SecCo'04)*. Elsvier Science Publishers, London, UK, ENTCS.

KWIATKOWSKA, M., NORMAN, G. ET PARKER, D. (2001). PRiSM : Probabilistic Symbolic Model-cheker. *Technical Report 760/2001*. University of Dortmund, Italy.

LAFRANCE, S. (2004). Symbolic approach to the analysis of security protocols. *J. UCS*, 10, 1156–1198.

LAFRANCE, S. ET MULLINS, J. (2002). Bisimulation-based non-deterministic admissible interference with applications to the analysis of cryptographic protocols. J. Harland, éditeur, *Computing : The Australasian Theory Symposium*. Elsevier, vol. 61 de *Electronic Notes in Theoretical Computer Science*, 1–24.

LAFRANCE, S. ET MULLINS, J. (2003a). Symbolic approach to the analysis of security protocols. I. Cervesato, éditeur, *Proceedings of Fondations of Computer Security*.

LAFRANCE, S. ET MULLINS, J. (2003b). Using admissible interference to detect denial of service vulnerabilities. J. M. Morris, B. Aziz et F. Oehl, éditeurs, *Sixth International Workshop in Formal Methods*. Electronic Workshops in Computing (eWiC) by British Computer Society (BCS).

LAUD, P. ET VENE, V. (2005). A type system for computationally secure information flow. M. Liskiewicz et R. Reischuk, éditeurs, *FCT*. Springer, vol. 3623 de

Lecture Notes in Comp. Sc., 365–377.

LINCOLN, P. ET RUSHBY, J. M. (1993). A formally verified algorithm for interactive consistency under a hybrid fault model. *FTCS*. 402–411.

LOWE (1997). A hierarchy of authentication specifications. *PCSFW : Proceedings of The 10th Computer Security Foundations Workshop*. IEEE Computer Society Press.

LOWE, G. (1996). Breaking and fixing the Needham-Schroeder public-key protocol using FDR. *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. Springer-Verlag, vol. 1055 de *LNCS*, 147–166.

LOWE, G. (2002). Quantifying information flow.

LOWE, G. ET ROSCOE, B. (1997). Using csp to detect errors in the tmn protocol. *IEEE Transactions on Software Engineering*, 23, 659–669.

MANTEL, H. (2001). Information flow control and applications - bridging a gap. J. N. Oliveira et P. Zave, éditeurs, *FME*. Springer, vol. 2021 de *Lecture Notes in Computer Science*, 153–172.

MEADOWS, C. (1996). The NRL protocol analyzer : An overview. *Journal of Logic Programming*, 26, 113–131.

MEADOWS, C. (2001). A cost-based framework for analysis of denial of service networks. *Journal of Computer Security*, 9, 143–164.

MEADOWS, C. ET SYVERSON, P. (1998). A formal specification of requirements for payment transactions in the set protocol. *DRAFT for Preproceedings of Financial Cryptography 98*. Anguilla, BWI.

MILLEN, J. ET SHMATIKOV, V. (2003). Symbolic protocol analysis with pro-

ducts and Diffie-Hellman exponentiation. *Proc. 16th Computer Security Foundation Workshop (CSFW'03)*. IEEE Comp. Soc. Press, Pacific Grove, Calfornia USA, 47–62.

MILLEN, J. K. (1984). The interrogator : A tool for cryptographic protocol security. *IEEE Symposium on Security and Privacy*. 134–141.

MILNER, R. (1989). *Communication and Concurrency*. Prentice-Hall.

MITCHELL, J., RAMANATHAN, A., SCEDROV, A. ET TEAGUE, V. (2006). A probabilistic polynomial-time process calculus for the analysis of cryptographic protocols. *Theoretical Computer Science*, 353, 118–164.

MULLINS, J. (2000). Non-deterministic admissible interference. *Journal of Universal Computer Science*.

NEEDHAM, R. M. ET SCHROEDER, M. D. (1978). Using encryption for authentication in large networks of computers. *Commun. ACM*, 21, 993–999.

NORMAN, G. ET SHMATIKOV, V. (2002a). Analysis of probabilistic contract signing. *FASec*. 81–96.

NORMAN, G. ET SHMATIKOV, V. (2002b). Analysis of probabilistic contract signing. A. E. Abdallah, P. Ryan et S. Schneider, éditeurs, *FASec*. Springer, vol. 2629 de *Lecture Notes in Computer Science*, 81–96.

PINSKY, S. (1992). An algebraic approach to non-interference. *CSFW*. 34–47.

PUTERMAN, M. (1994). Markov Decision Processes-Discrete Stochastic Dynamic programming. John Wiley & Sons, Inc, New-York, N.Y.

REITER, M. K. ET RUBIN, A. D. (1998). Crowds : Anonymity for web transactions.

ACM Transactions on Information and Systems Security, 1, 66–92.

ROSCOE, A. W. ET GOLDSMITH, M. H. (1999). What is intransitive noninterference? *CSFW*. 228–238.

RUSINOWITCH, M. ET TURUANI, M. (2001). Protocol insecurity with finite number of sessions is np-complete. *Proceedings of the 14th Computer Security Foundations Workshop*. IEEE Computer Society Press, Cape Breton, Nova Scotia, Canada, 174–190.

RYAN ET SCHNEIDER (1999). Process algebra and non-interference. *PCSFW : Proceedings of The 12th Computer Security Foundations Workshop*. IEEE Computer Society Press.

RYAN, P. Y. A. ET SCHNEIDER, S. A. (1998). An attack on a recursive authentication protocol. a cautionary tale. *Inf. Process. Lett.*, 65, 7–10.

SAGIV, S., éditeur (2005). *Programming Languages and Systems, 14th European Symposium on Programming,ESOP 2005, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2005, Edinburgh, UK, April 4-8, 2005, Proceedings*, vol. 3444 de *Lecture Notes in Computer Science*. Springer.

SCHNEIDER, S. ET SIDIROPOULOS, A. (1996). Csp and anonymity. *Proc. Comp. Security - ESORICS 96*. Springer-Vale, vol. 1146 de *LNCS*, 198–218.

SHMATIKOV, V. (2004a). Decidable analysis of cryptographic protocols with products and modular exponentiation. *Proc. 13th European Symposium On Programing (ESOP'04)*. Springer-Verlag, Barcelona, Spain, vol. 2986 de *LNCS*, 355–369.

SHMATIKOV, V. (2004b). Probabilistic model checking of an anonymity system. *Journal of Computer Security*, 12, 355–377.

STALLINGS, W. (2002). *Cryptography and Network Security : Principles and Practice.* Pearson Education.

SYVERSON, P. ET MEADOWS, C. (1993). A logical language for specifying cryptographic protocol requirements. *SP '93 : Proceedings of the 1993 IEEE Symposium on Security and Privacy.* IEEE Computer Society, Washington, DC, USA, 165.

SYVERSON, P. ET MEADOWS, C. (1995). Formal requirements for key distribution protocols. *Lecture Notes in Computer Science*, 950, 320–331.

SYVERSON, P. F. ET MEADOWS, C. (1996). A formal language for cryptographic protocol requirements. *Designs, Codes and Cryptography*, 7, 27–59.

SYVERSON, P. F. ET STUBBLEBINE, S. G. (1999). Group principals and the formalization of anonymity. *FM '99 : Proceedings of the Wold Congress on Formal Methods in the Development of Computing Systems-Volume I.* Springer-Verlag, London, UK, 814–833.

VAN GLABBEEK, R., SMOLKA, S. ET STEFFEN, B. (1995). Reactive, Generative and Stratified Models of Probabilistic Proccesses. vol. 121 de *Information and Computation*, 59–80.

ANNEXE I

Spécification du protocole ASET dans le modèle CSPAD

TABLEAU I.1 *Spécifications des processus client et sa banque en* CSPAD

$$
\begin{array}{|rl|}
\hline
C ::= & c_1(x_1).C^1 \\
C^1 ::= & [(n_c, id_c, x_1) \vdash_{pair} x_2][(x_2, KS_c) \vdash_{enc} x_3][((x_2, x_3), K_b) \vdash_{enc} x_4]\overline{c_2}(x_4).C^2 \\
C^2 ::= & c_3(x_5).[(x_5, {K_c}^{-1}) \vdash_{dec} x_6] \prod_{1 \le i \le 3} [x_6 \vdash_{p_i} x_{6+i}] \\
& [(n_c, x_7, id_c, x_1, x_8) \vdash_{pair} x_{10}][x_{10} \vdash_{hash} x_{11}] \\
& [(x_9, {KS_b}^{-1}) \vdash_{dec} x_{12}][x_{11} = x_{12}]C^3 \\
C^3 ::= & [((n_c, K_{cm}), K_m) \vdash_{enc} x_{13}][(listP, x_{13}) \vdash_{pair} x_{14}]\overline{c_4}(x_{14}).C^4 \\
C^4 ::= & c_5(x_{15})[(x_{15}, K_{cm}) \vdash_{dec} x_{16}] \prod_{1 \le i \le 4} [x_{16} \vdash_{p_i} x_{16+i}] \\
& [(n_c, x_{17}, id_m, x_{18}, listP, x_{19}) \vdash_{pair} x_{21}][x_{21} \vdash_{hash} x_{22}] \\
& [(x_{20}, {KS_m}^{-1}) \vdash_{dec} x_{23}][x_{22} = x_{23}]C^5 \\
C^5 ::= & [(n_c, x_{17}, x_7, id_m, id_c, x_{18}, x_8, x_{19}) \vdash_{pair} x_{24}][(x_{24}, KS_C) \vdash_{pair} x_{25}] \\
& [(n_c, x_{17}, cert_b, x_{18}, x_8, x_{19}, x_{25}) \vdash_{pair} x_{26}][(x_{26}, K_{cm}) \vdash_{enc} x_{27}] \\
& [(x_{18}, x_{27}) \vdash_{pair} x_{28}]\overline{c_6}(x_{28}).C^6 \\
C^6 ::= & c_{11}(x_{29})[(x_{29}, K_{cm}) \vdash_{dec} x_{30}] \prod_{1 \le i \le 4} [x_{28} \vdash_{p_i} x_{30+i}] \\
& [(n_c, x_{17}, x_{18}, x_8, x_{31}, x_{32}, x_{33}) \vdash_{pair} x_{35}][x_{35} \vdash_{hash} x_{36}] \\
& [(x_{34}, {KS_m}^{-1}) \vdash_{dec} x_{37}][x_{36} = x_{37}] \\
& [(n_c, x_{17}, x_7, id_m, id_c, x_{18}, x_8, x_{19}, x_{31}) \vdash_{pair} x_{38}][x_{38} \vdash_{hash} x_{39}] \\
& [(x_{33}, {KS_b}^{-1}) \vdash_{dec} x_{40}][x_{39} = x_{40}]\mathbf{0} \\
\hline
B ::= & c_2(z_1).[(x_1, {K_b}^{-1}) \vdash_{dec} z_2][z_2 \vdash_{p_1} z_3][z_2 \vdash_{p_2} z_4] \\
& \prod_{1 \le i \le 3} [z_3 \vdash_{p_i} z_{4+i}][z_6 = id_c][z_3 \vdash_{hash} z_8] \\
& [(z_4, {KS_c}^{-1}) \vdash_{dec} z_9][z_8 = z_9][z_7 = cert_m] \\
B^1 ::= & [(z_5, n_b, z_6, z_7, ref_b).z_{11} := enc_B(KS_b, z_{10}) \vdash_{pair} z_{10}] \\
& [((n_b, ref_b, z_{11}), K_c) \vdash_{enc} z_{12}]\overline{c_3}(z_{12}).B^2 \\
B^2 ::= & c_9(z_{13}).[(z_{13}, {K_b}^{-1}) \vdash_{dec} z_{14}][z_{14} \vdash_{p_1} z_{15}][z_{14} \vdash_{p_2} z_{16}] \\
& \prod_{1 \le i \le 7} [z_{15} \vdash_{p_i} z_{16+i}][z_{21} = ref_b \vee z_{17} = z_5][z_{15} \vdash_{hash} z_{17}] \\
& [(z_{16}, {KS_m}^{-1}) \vdash_{dec} z_{18}][z_{17} = z_{18}][(z_5, z_{20}, n_b, z_{21}, z_6, z_{22}, ref_b, z_{24}) \vdash_{pair} z_{26}] \\
& [z_{26} \vdash_{hash} z_{27}][(z_{25}, {KS_c}^{-1}) \vdash_{dec} z_{28}][z_{27} = z_{28}] \\
B^3 ::= & [(z_5, z_{20}, n_b, z_{21}, z_6, z_{22}, ref_b, z_{24}, status) \vdash_{pair} z_{29}][(z_{29}, KS_b) \vdash_{enc} z_{30}] \\
& [(z_5, z_{20}, z_{21}, id_b, z_{22}, ref_b, z_{24}, status, z_{30}) \vdash_{pair} z_{31}] \\
& [(z_{31}, KS_b) \vdash_{enc} z_{32}][((z_{22}, status, z_{30}, z_{32}), K_m) \vdash_{enc} z_{33}]\overline{c_{10}}(x_{33}).\mathbf{0} \\
\hline
\end{array}
$$

TABLEAU I.2 *Spécifications des processus marchand et sa banque en* CSPAD

$$
\begin{array}{rl}
M ::= & \overline{c_1}(cert_m).M^1 \\
M^1 ::= & c_4(y_1).[y_1 \vdash_{p_1} y_2][y_1 \vdash_{p_2} y_3][(y_3, K_m^{-1}) \vdash_{p_1} y_4] \\
& [y_4 \vdash_{p_1} y_5][y_4 \vdash_{p_2} y_6]M^2 \\
M^2 ::= & [(y_5, n_m, id_m, ref_m, y_2, \$) \vdash_{pair} y_7][(y_7, KS_m) \vdash_{enc} y_8] \\
& [((n_m, ref_m, \$, y_8), y_6) \vdash_{enc} y_9]\overline{c_5}(y_9).M^3 \\
M^3 ::= & c_6(y_{10}).[y_{10} \vdash_{p_1} y_{11}][y_{10} \vdash_{p_2} y_{12}] \\
& [y_{11} = ref_m][(y_{12}, y_6) \vdash_{dec} y_{13}] \prod_{1 \le i \le 6} [y_{13} \vdash_{p_i} y_{13+i}] \\
& [y_{14} = y_5 \vee y_{15} = n_m \vee y_{17} = ref_m][(n'_m, id_m, y_{18}) \vdash_{pair} y_{20}] \\
& [(y_{20}, KS_m) \vdash_{enc} y_{21}][((y_{20}, y_{21}), K_a) \vdash_{enc} y_{22}]\overline{c_7}(y_{22}).M^4 \\
M^4 ::= & c_8(y_{23}).[y_{20} \vdash_{hash} y_{24}][(y_{23}, KS_a^{-1}) \vdash_{dec} y_{25}][y_{24} = y_{25}]M^5 \\
M^5 ::= & [(y_5, n_m, id_m, ref_m, y_{18}, \$, y_{19}) \vdash_{pair} y_{26}][(y_{26}, KS_m) \vdash_{pair} y_{27}] \\
& [((y_{26}, y_{27}), K_b) \vdash_{enc} y_{28}]\overline{c_9}(y_{28}).M^6 \\
M^6 ::= & c_{10}(y_{29}).[(y_{29}, {K_m}^{-1}) \vdash_{dec} y_{30}] \prod_{1 \le i \le 4} [y_{30} \vdash_{p_i} y_{30+i}] \\
& [y_{31} = ref_m][(y_5, n_m, id_m, id_b, ref_m, y_{18}, \$, y_{32}, y_{33}) \vdash_{pair} x_{35}] \\
& [y_{35} \vdash_{hash} y_{36}][(y_{34}, KS_b^{-1}) \vdash_{dec} y_{37}][y_{36} = y_{37}]M^7 \\
M^7 ::= & [(y_5, n_m, ref_m, y_{18}, receipt, y_{33}) \vdash_{pair} y_{38}] \\
& [(y_{38}, KS_m) \vdash_{sing} y_{39}][((receipt, y_{32}, y_{33}, y_{39}), y_6) \vdash_{enc} y_{40}]\overline{c_{11}}(y_{40}).\mathbf{0}
\end{array}
$$

$$
\begin{array}{rl}
A ::= & c_7(t_1).[(t_1, K_a^{-1}) \vdash_{dec} t_2][t_2 \vdash_{p_1} t_3][t_2 \vdash_{p_2} t_4] \\
& \prod_{1 \le i \le 3} [t_3 \vdash_{p_i} t_{4+i}][t_6 = id_m][t_3 \vdash_{hash} t_8][(t_4, KS_m^{-1}) \vdash_{dec} t_9][t_8 = t_9]A^1 \\
A^1 ::= & [t_7 = cert_b][(t_8, KS_a) \vdash_{enc} t_{10}]\overline{c_8}(t_{10}).\mathbf{0}
\end{array}
$$

ANNEXE II

Preuve du théorème 5.4.2

Nous allons prouver le théorème par une induction sur la longueur des séquences (traces) observables. Plus particulièrement, nous allons prouver que, pour toute séquence $s = \alpha_1\alpha_2\cdots\alpha_n$ d'actions visibles et tout ordonnenceur S,

$$\text{Prob}[\phi \rhd P \rightsquigarrow_S s] = \text{Prob}[\phi \rhd Q \rightsquigarrow_S s].$$

$P \approx_\phi Q \Rightarrow \exists\Re$, une bisimulation t.q. $(\phi \rhd P, \phi \rhd Q) \in \Re$.
[Base :] $n = 1$, c'est-à-dire $s = \alpha_1$.

$$\begin{aligned}
|\text{Prob}[\phi \rhd P \rightsquigarrow_S s] &- \text{Prob}[\phi \rhd Q \rightsquigarrow_S s]| \\
&= |\sum_{\psi \rhd R \in Conf} \mu(\phi \rhd P, \stackrel{\hat{\alpha_1}}{\Longrightarrow}_S, \{\psi \rhd R\}) \\
&\quad - \sum_{\psi' \rhd R' \in Conf} \mu(\phi \rhd Q, \stackrel{\hat{\alpha_1}}{\Longrightarrow}_S, \{\psi' \rhd R'\})| \\
&= |\mu(\phi \rhd P, \stackrel{\hat{\alpha_1}}{\Longrightarrow}_S, Conf) - \mu(\phi \rhd Q, \stackrel{\hat{\alpha_1}}{\Longrightarrow}_S, Conf)| \\
&= |\sum_{\mathscr{U} \in Conf/\Re} (\mu(\phi \rhd P, \stackrel{\hat{\alpha_1}}{\Longrightarrow}_S, \mathscr{U}) - \mu(\phi \rhd Q, \stackrel{\hat{\alpha_1}}{\Longrightarrow}_S, \mathscr{U}))| \\
&= 0.
\end{aligned}$$

[Étape d'induction :] supposons que, pour toute séquence $s' = \alpha'_1\alpha'_2\cdots\alpha'_k$ t.q. $k < n$ et pour toutes configurations $(\psi \rhd R, \psi' \rhd R') \in \Re$, on a

$$|\text{Prob}[\psi \rhd R \rightsquigarrow_S s'] - \text{Prob}[\psi' \rhd R' \rightsquigarrow_S s']| = 0.$$

Alors, on a :

$$
\begin{aligned}
&|\mathrm{Prob}[\phi \rhd P \rightsquigarrow_S s] - \mathrm{Prob}[\phi \rhd Q \rightsquigarrow_S s]| \\
&\quad = |\textstyle\sum_{\psi \rhd R \in Conf} \mu(\phi \rhd P, \stackrel{\hat{\alpha_1}}{\Longrightarrow}_S, \{\psi \rhd R\}) \cdot \mathrm{Prob}[\psi \rhd R \rightsquigarrow_S \alpha_2 \cdots \alpha_n] \\
&\qquad - \textstyle\sum_{\psi' \rhd R' \in Conf} \mu(\phi \rhd Q, \stackrel{\hat{\alpha_1}}{\Longrightarrow}_S, \{\psi' \rhd R'\}) \cdot \mathrm{Prob}[\psi' \rhd R' \rightsquigarrow_S \alpha_2 \cdots \alpha_n]| \\
&\quad = |\textstyle\sum_{\mathscr{U} \in Conf/\Re} (\sum_{\psi \rhd R \in \mathscr{U}} \mu(\phi \rhd P, \stackrel{\hat{\alpha_1}}{\Longrightarrow}_S, \{\psi \rhd R\}) \cdot \mathrm{Prob}[\psi \rhd R \rightsquigarrow_S \alpha_2 \cdots \alpha_n] \\
&\qquad - \textstyle\sum_{\psi' \rhd R' \in \mathscr{U}} \mu(\phi \rhd Q, \stackrel{\hat{\alpha_1}}{\Longrightarrow}_S, \{\psi' \rhd R'\}) \cdot \mathrm{Prob}[\psi' \rhd R' \rightsquigarrow_S \alpha_2 \cdots \alpha_n])
\end{aligned}
$$

Soit $\psi_{\mathscr{U}} \rhd R_{\mathscr{U}}$, un représentant de la classe d'équivalence $\mathscr{U}$. Par hypothèse d'induction nous avons :

$$\forall \psi \rhd R \in \mathscr{U}.\mathrm{Prob}[\psi \rhd R \rightsquigarrow_S \alpha_2 \cdots \alpha_n] = \mathrm{Prob}[\psi_{\mathscr{U}} \rhd R_{\mathscr{U}} \rightsquigarrow_S \alpha_2 \cdots \alpha_n].$$

Ainsi, on a :

$$
\begin{aligned}
&|\mathrm{Prob}[\phi \rhd P \rightsquigarrow_S s] - \mathrm{Prob}[\phi \rhd Q \rightsquigarrow_S s]| \\
&\quad = |\textstyle\sum_{\mathscr{U} \in Conf/\Re} (\sum_{\psi \rhd R \in \mathscr{U}} \mu(\phi \rhd P, \stackrel{\hat{\alpha_1}}{\Longrightarrow}_S, \{\psi \rhd R\}) \\
&\qquad - \textstyle\sum_{\psi' \rhd R' \in \mathscr{U}} \mu(\phi \rhd Q, \stackrel{\hat{\alpha_1}}{\Longrightarrow}_S, \{\psi' \rhd R'\})) \cdot \mathrm{Prob}[\psi_{\mathscr{U}} \rhd R_{\mathscr{U}} \rightsquigarrow_S \alpha_2 \cdots \alpha_n] \\
&\quad = |\textstyle\sum_{\mathscr{U} \in Conf/\Re} (\mu(\phi \rhd P, \stackrel{\hat{\alpha_1}}{\Longrightarrow}_S, \mathscr{U}) - \mu(\phi \rhd Q, \stackrel{\hat{\alpha_1}}{\Longrightarrow}_S, \mathscr{U})) \\
&\qquad\qquad \cdot \mathrm{Prob}[\psi_{\mathscr{U}} \rhd R_{\mathscr{U}} \rightsquigarrow_S \alpha_2 \cdots \alpha_n] \\
&\quad = 0 \text{ puisque} P \approx_\phi Q.
\end{aligned}
$$