INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Projet SECSI

Sécurité des systèmes d'information

Futurs

THÈME 2A

Table des matières

1. Composition de l'équipe

SECSI est un projet commun entre l'INRIA et le laboratoire LSV, lui-même laboratoire commun entre le CNRS (UMR 8643) et l'ENS de Cachan.

Responsable scientifique
Jean Goubault-Larrecq [professeur]

Personnel Inria
Florent Jacquemard [CR, chez SECSI depuis le 1er octobre 2002]

Personnel CNRS
Ralf Treinen [Maître de conférences, à l'ENS Cachan depuis le 1er septembre 2002]
Stéphane Demri [CR CNRS]

Personnel ENS Cachan
Hubert Comon-Lundh [responsable permanent ; DR CNRS jusqu'au 31 août 2002, professeur depuis le 1er septembre 2002]

Chercheur post-doctorant
Fabrice Parrennes [CDD sur ACI « jeunes chercheurs » « protocoles cryptographiques et detection d'intrusions », depuis le 1er septembre 2002]

Ingénieur de recherche
Julien Olivain [CDD sur projet RNTL DICO, à partir du 1er décembre 2002]

Doctorants
Vincent Bernat [élève normalien, ENS Cachan, depuis le 1er septembre 2002]
Alexandre Boisseau [bourse AMN, Ecole Doctorale Sciences Pratiques]
Véronique Cortier [bourse AMN, Ecole Doctorale Sciences Pratiques]
Muriel Roger [bourse MENRT, Ecole Doctorale Sciences Pratiques]
Kumar Neeraj Verma [bourse MENRT, Ecole Doctorale Sciences Pratiques]
Yu Zhang [bourse MENRT sur ACI « cryptologie », Ecole Doctorale Sciences Pratiques, depuis le 1er octobre 2002]

Stagiaire
Shalendra Chhabra [Institute of Technology-Benares Hindu University, Vanirasi, Inde, du 1er mai au 15 juillet 2002]

2. Présentation et objectifs généraux

Action depuis mars 2002, le projet SECSI a été créé en tant que projet en novembre 2002. SECSI est un projet commun entre l'INRIA et le laboratoire LSV, lui-même laboratoire commun entre le CNRS (UMR 8643) et l'ENS de Cachan. Le projet SECSI est un projet de recherche sur la sécurité des systèmes d'information. Il est organisé autour de trois axes, et de leurs relations mutuelles :

- Vérification de protocoles cryptographiques ;
- Détection d'intrusions ;
- Analyse statique de programmes, dans le but de détecter des trous de sécurité et des vulnérabilités au niveau protocolaire.

Le projet SECSI a pour objectifs :

- de développer de nouveaux modèles et de nouvelles logiques de description de propriétés cryptographiques : secret, authentication, anonymat, intimité, etc.
- de développer de nouvelles méthodes de vérification automatique de protocoles cryptographiques ;

- de développer les techniques de model-checking, particulièrement de model-checking en ligne, dans le cadre de la détection d'intrusions ;
- de développer de nouvelles techniques d'analyse statique de code dans le but d'analyser la sécurité des applications cryptographiques ;
- d'intégrer les techniques d'analyse statique de code et de surveillance dynamique (détection d'intrusions).

Du côté des applications, SECSI vise particulièrement :

- la création d'une plateforme de vérification de protocoles cryptographiques ;
- la création d'outils de détection d'intrusions performants ;
- la réalisation de prototypes d'analyseurs statiques de code, utilisables sur de gros codes.

3. Fondements scientifiques

3.1. Quelques aspects de la sécurité informatique

Mots clés : *sécurité informatique*, *vérification*, *protocoles cryptographiques*, *analyse statique de code*, *détection d'intrusion*, *model-checking*.

Glossaire

vérification traduction de l'anglais « model-checking ».

model-checking regroupe un large éventail de techniques automatiques permettant de s'assurer qu'un modèle formel d'un système informatique obéit à une spécification, typiquement écrite comme une formule d'une logique adaptée.

protocole une suite de messages définissant une interaction entre deux ou plusieurs machines, logiciels, ou personnes.

protocole cryptographique un protocole fondé sur des moyens, de chiffrement entre autres, et visant à assurer des propriétés de confidentialité, d'authentification, ou autres propriétés de sécurité.

analyse statique de code corpus de méthodes permettant de déterminer les propriétés vérifiées par des programmes donnés, sans les exécuter, mais en examinant leurs codes source, ou parfois le code compilé ; l'analyse statique se confond largement avec l'interprétation abstraite de programmes.

détection d'intrusion corpus de méthodes ayant pour but de détecter des attaques, intrusions, ou anomalies dans les systèmes informatiques, par des moyens de surveillance en temps réel de l'activité des réseaux et des systèmes.

Le domaine de SECSI est la sécurisation des systèmes d'information. Plutôt que de discuter des enjeux économiques et des problèmes techniques de la sécurité informatique d'emblée, nous avons choisi de commencer ce texte par la citation suivante, prise parmi d'autres :

> "Sécuriser l'information doit être un souci général. Sécuriser les systèmes d'information est une obligation nationale majeure."
>
> Directive interministérielle nº 4201/SG du 13 avril 1995
>
> relative à la sécurité des systèmes d'information.

Bien sûr, hors du contexte, une telle citation n'a pas beaucoup de sens. Néanmoins, cette directive, parmi beaucoup d'autres, est représentative des préoccupations croissantes en matière de sécurité. En particulier, le problème de confidentialité se pose de manière aiguë dans le commerce électronique, la téléphonie mobile, les cartes à puce, la défense du caractère privé de certaines données.

Traditionnellement, le premier problème de sécurité est la confidentialité :

- Certaines données doivent rester confinées à un cercle d'initiés.
- Vu sous un autre angle, il s'agit de contrôler l'accès à certaines ressources.

Ce problème n'est pas nouveau et se pose depuis très longtemps dans les systèmes d'exploitation. Ce qui change essentiellement est qu'on s'intéresse à ces problèmes dans un environnement distribué dont les canaux de communication ne sont pas fiables.

Mais la confidentialité est loin d'être la seule propriété de sécurité intéressante. Tout aussi primordiale pour contrôler l'accès à certaines ressources est l'authentification (deux agents communiquant acquièrent la certitude qu'ils se parlent bien l'un à l'autre et non à un étranger tentant de se faire passer pour l'un d'eux). Des problèmes tels que l'anonymat, le délit d'initié [60][59] etc., sont aussi importants pour différentes transactions.

En réponse aux attaques, de plus en plus fréquentes, contre les propriétés ci-dessus, on compte un certain nombre de mesures sécuritaires, parmi lesquelles :

Le chiffrement. L'idée est qu'avec une très haute probabilité, seul le détenteur de la clef peut obtenir en clair le texte chiffré.

Aspects protocolaires. Le chiffrement est insuffisant pour garantir la sécurité, même si l'on transforme la « très haute probabilité » ci-dessus en une certitude (hypothèse de *chiffrement parfait*). Les exemples d'attaques soit par réplication soit par usurpation d'identité abondent. C'est ce que nous appellerons les *aspects logiques des protocoles cryptographiques* dans la suite.

Le pistage des pirates. L'hypothèse de chiffrement parfait n'est qu'une approximation. En outre, on ne peut pas faire l'hypothèse que des agents, même honnêtes, suivent scrupuleusement les règles d'un protocole. Enfin, les protocoles eux-mêmes ne sont pas fiables et leurs réalisations encore moins. Il faut donc de manière complémentaire prévoir des mesures pour faire face aux attaques et en limiter les dégâts : couper les connections, envoyer un message à l'administrateur système, documenter les circonstances d'une attaque effectuée pour instruire une action légale ultérieure par exemple.

Ceci s'effectue à l'aide de techniques couramment regroupées sous le terme de *détection d'intrusions*, et qui comprennent diverses techniques allant du simple pattern-matching d'événements réseaux à des techniques plus élaborées à base d'automates ou de réseaux bayésiens par exemple.

Le protectionnisme. Dans l'esprit de limiter les dégâts, les portes d'entrée vers des sous-réseaux mettent en place des pare-feu (firewall) qui contrôlent l'accès au sous-réseau en empêchant les échanges directs. Diverses autres protections peuvent être mises en œuvre à différents niveaux, dépendant de la politique locale de sécurité : droits d'accès de fichiers Unix, profils de protection Java 2, interdiction partielle ou totale d'exécuter du code Java ou Javascript dans les browsers Web, etc.

Ces quatre mesures sont complémentaires et aucune d'entre elles considérée isolément ne peut suffire à garantir la sécurité.

Naïvement, on pourrait croire que chiffrer un message de manière à ce que seul l'interlocuteur puisse le déchiffrer, garantit la confidentialité de ce message. Mais il n'en est rien et l'exemple fameux du protocole de Needham-Schroeder à clés publiques, employé pendant plus de 15 ans (et soi-disant prouvé) avant que Gavin Lowe en trouve une attaque en 1996 [56], le démontre. En fait, presque tous les protocoles employés [49] se sont avérés attaquables.

Pour donner une idée plus précise du problème, décrivons informellement ce fameux protocole de Needham-Schroeder. Dans ce protocole, un agent honnête (Alice) entre en communication avec un agent malhonnête (Charles). Celui-ci, se faisant passer pour Alice, utilise les données envoyées par Alice dans une communication avec un troisième agent, Bob. La réponse de Bob, chiffrée de manière à ce que seule Alice puisse ouvrir le message, est renvoyée telle quelle à Alice qui, elle, ouvre le message croyant qu'il s'agit de la réponse de Charles à son premier envoi (et donc que les données contenues dans le message sont connues de Charles).

Elle en renvoie une partie, chiffrée de manière que seul Charles puisse ouvrir le message. Mais alors, Charles prend connaissance des données confidentielles de Bob, qui ne lui étaient pas destinées.

On voit que, dans cet exemple, le chiffrement n'est pas en cause : il s'agit bien d'un problème logique.

De même, il serait illusoire de croire qu'un protocole qui n'a pas d'attaque dans un modèle formel est complètement sûr, car la réalisation de ce protocole comporte toujours des constructions non décrites dans le modèle formel. On ne peut donc exclure l'existence d'attaques. Enfin, les entreprises utilisant des protocoles sécuritaires continuent souvent à utiliser ceux-ci, alors même qu'une attaque a été répertoriée. Il y a deux raisons à cela. La première est économique : il est souvent trop coûteux de re-développer un produit alors que, déjà, de nombreuses applications l'utilisent. L'autre raison est pratique : le protocole peut être devenu un standard, soit qu'il fasse l'objet d'une norme, soit qu'il se soit imposé comme un standard de fait (comme SSL [52]). Il est alors impossible de le changer car il faudrait imposer ce changement à tous les utilisateurs potentiels simultanément, ce qui n'est pas réaliste. On peut dire par exemple que, même si une attaque était découverte sur SSL, pendant de nombreuses années SSL resterait la base des connexions sécurisées sur le Web.

L'idée est alors de détecter les attaques et d'en limiter les dégâts, voire de se retourner contre les pirates. C'est pourquoi une trace des connexions et plus généralement du trafic sur le réseau est le plus souvent conservée dans d'énormes « fichiers de log », qui sont en principe analysés pour détecter des tentatives d'intrusion. La difficulté ici vient de la taille de ces fichiers et de l'absence d'une définition de ce qu'est une intrusion.

Pour terminer, on ne peut pas faire confiance aux trois mesures précédentes, et des mesures doivent donc être prises pour limiter l'impact d'une attaque. Il y a bien sûr des précautions de bon sens (par exemple qu'une unique clef ne permette pas d'accéder à tout ce qui est confidentiel dans une entreprise). D'autres mesures peuvent être prises dans la gestion du réseau en mettant en place des portes successives, de plus en plus difficiles à ouvrir selon le degré d'importance de la confidentialité des données.

De par nos compétences, notre projet porte sur les protocoles et le pistage des pirates (et les liens entre les deux), c'est-à-dire les deux mesures centrales dans la description du paragraphe précédent. Nous nous limitons à ces aspects dans les paragraphes qui suivent, à l'exclusion des aspects de chiffrement et des aspects réseaux/systèmes, sujets sur lesquels il existe des équipes plus compétentes en France.

D'une façon générale, les objectifs du projet SECSI sont d'appliquer les méthodes formelles à deux problèmes : la vérification de protocoles sécuritaires et la détection d'intrusion. Ces deux questions sont en fait complémentaires et il est primordial de les faire collaborer dans une optique de globalisation de la sécurité. En particulier, une des originalités de notre projet est de dériver de l'échec de la vérification d'un protocole des scénarii d'attaque, utilisés pour détecter des intrusions. Comme nous l'avons vu plus haut, cet objectif est particulièrement pertinent.

Comme dans nos travaux de recherche en vérification, l'objectif est aussi d'allier des recherches théoriques et des applications, au travers de collaborations industrielles et de développement de logiciels.

4. Domaines d'application

4.1. Panorama

Mots clés : *cartes à microprocesseur*, *architectures distribuées sécurisées*, *SSL*, *TLS*, *sécurité*.

Les domaines d'application de SECSI couvrent une large portion de la sécurité informatique.

Les protocoles cryptographiques sont une composante essentielle des applications sur cartes à microprocesseur, mais ce n'est pas l'unique domaine d'application visé au sein de SECSI. En particulier, ils sont de plus en plus utilisés dans des applications aussi diverses que des serveurs d'entreprise, des architectures de communication dans le ferroviaire, ou des interfaces homme-machine distribuées et sécurisées, dans le domaine militaire notamment. Ces domaines d'application sont donnés à titre d'illustration, SECSI n'étant à l'heure actuelle engagé dans aucun domaine particulier. Dans tous ces domaines, l'analyse de protocoles

cryptographiques ne suffit pas à garantir la sécurité du système ou du réseau, et les techniques de détection d'intrusion sont un indispensable complément.

Une application concrète, mais non liée à un secteur d'activité spécifique, et largement diffusée est le protocole SSL, et son successeur TLS, qui sont utilisés dans toutes les transactions sécurisées via les navigateurs Web actuels, ainsi que dans les applications `ssh`, `slogin`, `scp` de connexion à distance et de copie de fichiers sécurisées.

5. Logiciels

5.1. Les logiciels et réalisations informatiques chez SECSI

Le projet SECSI démarre avec une base de logiciels relativement importante : des outils d'analyse syntaxique, de traduction, de vérification de protocoles cryptographiques qui s'insèrent dans l'architecture du projet RNTL EVA (dont CPV, CPV2, Securify), un outil en cours de développement d'analyse de code C (CSur), un outil de détection d'intrusions (logWeaver). Tous ces logiciels sont antérieurs à SECSI. La page Web SPORE est elle nouvelle, et est une bibliothèque publique et ouverte de protocoles cryptographiques à vocation de centralisation des informations au plan international.

5.2. L'architecture EVA

Participants : Jean Goubault-Larrecq [correspondant], Florent Jacquemard, Véronique Cortier.

Mots clés : *EVA*, *protocole cryptographique*, *architecture*.

Le projet EVA (Explication et Vérification Automatique de protocoles cryptographiques) est un projet RNTL portant sur la création d'outils de vérification automatique de protocoles cryptographiques, et l'extraction d'explications de ces vérifications, sous forme de preuves lisibles et re-vérifiables.

En tant que système logiciel, EVA est aussi une architecture logicielle, illustrée dans la figure ci-dessous.

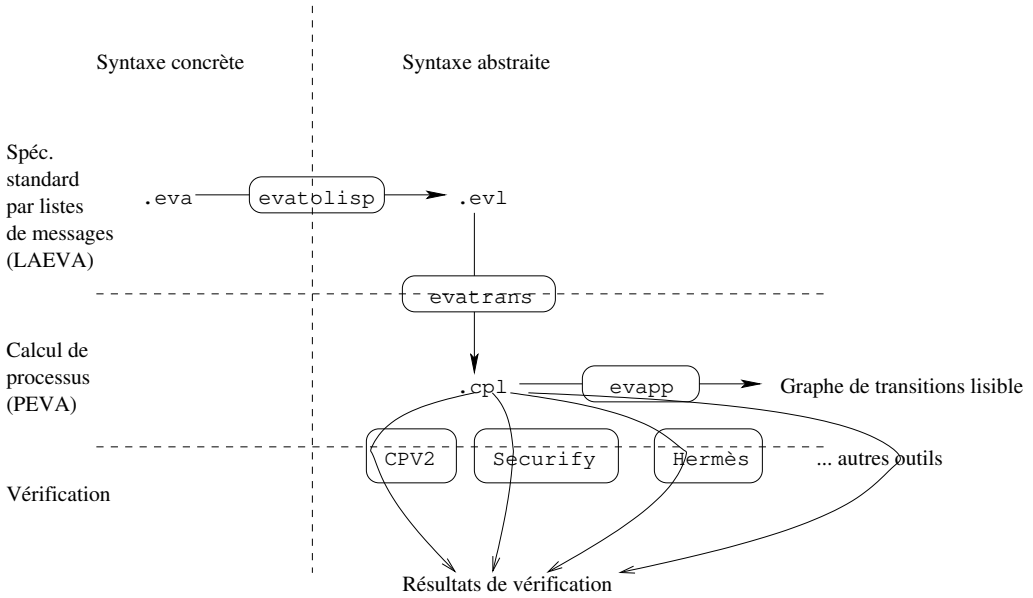

Figure 1.

L'architecture se veut le plus ouverte possible. EVA consiste en plusieurs outils, communiquant via des fichiers dans des formats texte documentés, dans une notation ressemblant à Lisp.

L'outil `evatolisp` est l'*analyseur syntaxique*, et produit une description brute des protocoles EVA dans un format textuel (`.evl`) en notation Lisp. L'outil `evatrans` est le *traducteur*, qui transforme les descriptions brutes de protocoles EVA en des descriptions à base d'algèbre de processus (`.cpl`, un autre format textuel en notation Lisp), après avoir vérifié les types des messages et l'implémentabilité des protocoles. L'outil `evapp` est le *pretty-printer*, permettant d'obtenir une version lisible des algèbres de processus en question. Les outils de vérification prennent des descriptions de protocoles cryptographiques, avec les propriétés à vérifier, au format `.cpl` en entrée. A l'heure actuelle, les outils de vérification existants sont Securify (LSV, SECSI) et Hermès (Verimag). L'outil CPV2 devrait être finalisé en 2003.

5.3. L'analyseur syntaxique et le traducteur EVA

Participants : Jean Goubault-Larrecq [correspondant, jusqu'au 1er décembre 2002], Florent Jacquemard [correspondant, à partir du 1er décembre 2002].

Mots clés : *EVA*, *protocole cryptographique*, *notation standard*.

Dans le cadre du projet EVA (Explication et Vérification Automatique de protocoles cryptographiques), Jean Goubault-Larrecq a réalisé un analyseur syntaxique de spécifications de protocoles cryptographiques, en notation standard. Cet analyseur vérifie aussi l'implémentabilité des protocoles, le bon typage des messages ; finalement, il les traduit en une collection de processus, ou programmes, décrivant les actions de chaque participant du protocole.

En tant que tel, ce logiciel a été réalisé par des participants du projet SECSI, mais a commencé en 2001, avant la création du projet.

Dans le cadre de l'architecture logicielle EVA (figure ci-dessous), ce logiciel est constitué des trois programmes `evatolisp`, `evatrans`, `evapp`.

Par exemple, le protocole de Needham-Schroeder à clés publiques s'écrira en notation standard :

```
alg : asym_algo     // un algorithme de chiffrement à clés publiques
everybody knows alg // connu de tous.
A, B, S : principal // les identites des trois participants de chaque
                    // session du protocole.
Na, Nb : number     // deux messages; ce seront les nonces du protocole.
keypair^alg PK, SK (principal)  // deux familles de clés:
                                // PK(X) sera la clé publique de X,
                                // SK(X) sera la clé secrète de X;
                                // la déclaration keypair déclare que
                                // PK(X) et SK(X) sont inverses l'une
                                // de l'autre pour l'algorithme alg.
everybody knows PK  // de plus, PK(X) est connu de tous, pour tout X.
A knows A, B, SK(A) // on suppose que A connaît sa propre identité,
                    // l'identité de B, et sa propre clé secrète SK(A).
B knows B, SK(B)    // B connaît sa propre identite et sa clé secrète SK(B).
S knows SK(S)       // S connaît sa clé secrète.
// La liste des messages du protocole lui-même:
{
  1. A -> S : A, B
  2. S -> A : {PK (B), B}_(SK (S))^alg
  3. A -> B : {Na, A}_(PK(B))^alg
  4. B -> S : B, A
  5. S -> B : {PK (A), A}_(SK (S))^alg
  6. B -> A : {Na, Nb}_(PK (A))^alg
  7. A -> B : {Nb}_(PK (B))^alg
}
```

Il sera traduit en trois descriptions de processus, correspondant aux rôles `A`, `B`, et `S`. Une version légèrement expurgée de la sortie du pretty-printer sur ce protocole produit :

```
process A(a:principal,b:principal) : init=1. {
  1. -> 2. : send (a,b)
  2. -> 3. : recv _x3
  3. -> %5. : new Na-1
  %5. -> 4. : send {(Na-1,a)}_(apply-pubk^(alg)(_kpf1,b))^(alg)
  4. -> 7. : recv {(<Na-1>,_x7)}_(apply-privk^(alg)(_kpf1,a))^(alg)
  7. -> end. : send {_x7}_(apply-pubk^(alg)(_kpf1,b))^(alg)
}
process B(b:principal) : init=1. {
  1. -> 4. : recv {(_x4,_x5)}_(apply-privk^(alg)(_kpf1,b))^(alg)
  4. -> 5. : send (b,_x5)
  5. -> 6. : recv _x6
  6. -> %12. : new Nb-1
  %12. -> 7. : send {(_x4,Nb-1)}_(apply-pubk^(alg)(_kpf1,_x5))^(alg)
  7. -> end. : recv <{Nb-1}_(apply-pubk^(alg)(_kpf1,b))^(alg)>
}
process S() : init=1. {
  1. -> 2. : recv (_x1,_x2)
  2. -> 3. : send {(apply-pubk^(alg)(_kpf1,_x2),_x2)}
                    _(apply-privk^(alg)(_kpf1,s))^(alg)
  3. -> 5. : recv <(_x2,_x1)>
  5. -> 6. : send {(apply-pubk^(alg)(_kpf1,_x1),_x1)}
                    _(apply-privk^(alg)(_kpf1,s))^(alg)
}
```

Le protocole lui-même est ensuite une mise en parallèle d'un nombre non borné de processus `A(a,b)`, `B(b)`, `S()`, pour différentes identités `a`, `b`. (Ceci s'appelle usuellement le mode *multi-sessions parallèles*.) Les algorithmes de vérification d'EVA ont pour but de vérifier que ces programmes parallèles vérifient les propriétés de sécurité visées.

Ce programme a été repris par Florent Jacquemard à partir de décembre 2002.

Voir http://www.lsv.ens-cachan.fr/~goubault/EVA.html.

5.4. Outils de vérification de protocoles cryptographiques CPV et CPV2

Participant : Jean Goubault-Larrecq [correspondant].

Mots clés : *EVA*, *protocole cryptographique*, *vérification*, *automate d'arbre*.

CPV et CPV2 sont deux logiciels de vérification de propriétés de confidentialité de protocoles cryptographiques, écrits par Jean Goubault-Larrecq. CPV a été développé au GIE Dyade puis au LSV à partir de 2000, et est fondé sur des techniques d'automates d'arbres. CPV2, commencé en 2002, est lui fondé sur des techniques de résolution, c'est-à-dire de démonstration automatique.

CPV est intégralement antérieur au projet SECSI. CPV2 est effectué dans le cadre du projet RNTL EVA, un projet où le LSV participe mais antérieur à SECSI.

L'originalité de CPV2 est d'être basé sur une bibliothèque de décision de clauses (correspondant à une certaine forme clausale issue de formules de la classe monadique, et qu'on appellera donc *clauses monadiques*), et une série de techniques d'abstraction et d'amélioration des abstractions, toutes automatiques. Certaines abstractions sont particulièrement adaptées à l'approximation de protocoles cryptographiques en clauses monadiques. On pourra consulter [30] pour la théorie sur laquelle se fonde cet outil.

CPV2 sera le troisième outil de vérification de protocoles cryptographiques qui s'insérera dans l'architecture EVA, après Hermès (Verimag) et Securify (Véronique Cortier, SECSI, LSV). CPV2 est réalisé à 70%, voir la courte description [47].

5.5. Outil de vérification de protocoles cryptographiques Securify

Participante : Véronique Cortier [correspondant].

L'outil Securify est entièrement dû à Véronique Cortier. Il s'agit d'une réalisation d'un algorithme qu'elle a inventé alors qu'elle était en visite au SRI auprès de Jon Millen en 2000, et qui a été poursuivi dans le cadre du projet RNTL EVA.

En tant que tel, ce logiciel a été développé indépendamment de SECSI.

Il s'agit d'un outil dédié à la preuve de propriétés de secret de protocoles cryptographiques. Il est décrit en [43], et est fondé sur un algorithme décrit dans un article présenté au Computer Security Foundations Workshop en 2001 [50].

Cet algorithme vérifie des propriétés de secret dans un cadre très général : nombre de sessions non borné, nombre de participants non borné, clés symétriques ou asymétriques. Par contre, il suppose que les données sont suffisamment typées et, pour le moment, il ne permet pas de traiter les protocoles à clés composées.

Le principe de cet outil est de faire la preuve de la correction du protocole testé par induction sur les règles utilisées. Aussi, l'algorithme consiste à vérifier qu'aucune des règles ne peut compromettre un secret. Pour cela trois tests de base permettent de conclure à la correction de la règle et si aucun de ces tests n'aboutit une recherche en arrière est appliquée pour obtenir plus d'informations et réappliquer, le cas échéant un des tests de base ou à nouveau une recherche en arrière.

L'outil Securify est d'autre part remarquablement rapide, comme le montre le tableau suivant :

Protocole	Preuve	# maximal de « back »	Temps ms
Otway-Rees	Oui	0	0,35
Woo and Lam	Oui	0	0,11
Denning-Sacco	Non	0	0,07
ISO Symmetric Key	Oui	0	0,15
Needham-Schroeder-Lowe	Oui	1	1,08
Needham-Schroeder	Non	1	1,23
Wide-Mouthed-Frog (modifié)	Oui	2	4,76
Kao-Chow	Oui	3	8,94
Yahalom	Oui	3	21,06

5.6. L'outil d'analyse statique CSur

Participants : Jean Goubault-Larrecq [correspondant], Fabrice Parrennes.

Mots clés : *Analyse statique*, *C*, *intervalles d'entiers*, *clause de Horn*.

CSur est un projet antérieur au projet SECSI (début 2002). Il est poursuivi par Fabrice Parrennes, sur CDD de l'ACI jeunes chercheurs « Sécurité informatique, protocoles cryptographiques et détection d'intrusions » attribuée à Jean Goubault-Larrecq en 2001, et qui, en tant que telle, est formellement indépendante de SECSI.

L'outil CSur est la racine de deux projets logiciels distincts :

- Le premier est un projet de programmation destiné aux élèves du cours « Analyse statique » du DESS « Développement de logiciels sûrs », et sur lequel est fondé leur note depuis 2002. Il s'agit d'un analyseur statique de programmes C, écrit en OCaml, et qui a pour but de détecter

des débordements arithmétiques, des accès illégaux à la mémoire (débordements de tableaux notamment). L'analyseur a été entièrement écrit par Jean Goubault-Larrecq, puis fourni aux élèves ; seuls certains modules n'ont été fournis qu'en version compilée, charge revenant aux élèves de réécrire les modules correspondants.
Voir http://www.lsv.ens-cachan.fr/~goubault/Csur/csur.html.

- Le deuxième projet est plus ambitieux, et est à la base du programme d'analyse statique du projet SECSI. Il est fondé originellement sur le code précédent, mais produit des listes de contraintes exprimant tant des propriétés approchées sur la forme des divers objets alloués en mémoire, que des propriétés de confidentialité de messages générés par le programme analysé.
Ces contraintes sont essentiellement des clauses de Horn, sur un langage de termes dédié à la description d'objets en mémoire. Ce langage de contraintes devrait d'autre part avoir un problème du vide décidable, ce qui reste encore à vérifier.
Ce deuxième projet est en cours de développement, et est écrit par Fabrice Parrennes.

5.7. logWeaver : un outil de détection d'intrusions fondé sur la logique temporelle

Participants : Jean Goubault-Larrecq [correspondant], Muriel Roger, Julien Olivain.

L'outil logWeaver répond à un triple besoin : celui de fournir un langage de spécification de signatures d'attaques qui soit *déclaratif*, pour des raisons de lisibilité et de maintenabilité, qui offre des capacités de reconnaissance d'attaques complexes impliquant en particulier des *corrélations* entre divers événements, notamment des corrélations temporelles, et finalement qui dispose d'un algorithme *efficace* de détection des attaques correspondant aux signatures spécifiées.

Le papier [9] introduit l'idée que la détection d'intrusions n'est en fait rien d'autre qu'une activité de *model-checking* de formules décrivant des attaques dans une logique temporelle adaptée, contre un modèle de Kripke consistant en le flux d'événements à surveiller. Le même papier étudie deux variantes de la logique du temps linéaire avec variables du premier ordre (pour enregistrer les valeurs de x ci-dessus, ou de l'utilisateur X ou de la machine cible dans l'exemple expliqué plus haut). La première, une adaptation directe de la logique du temps linéaire, est inadaptée tant pour des raisons de trop faible expressivité que d'inefficacité, notamment à cause de l'existence de constructions logiques inutiles ralentissant le moteur.

La deuxième, qui est la base de l'outil logWeaver [54], est spécialisée à des formules d'éventualité (une telle formule est vraie dans un modèle donné si et seulement si il existe un préfixe fini du modèle sur lequel elle est vraie). L'algorithme de model-checking en ligne proposé est rapide dans la plupart des situations pratiques, notamment grâce à l'utilisation d'optimisations obtenues par analyse statique des formules temporelles. De plus, l'algorithme retourne exactement les « shortest runs » (attaques minimales). La sémantique des attaques minimales est *correcte* : s'il existe une attaque, une attaque minimale existe aussi (on notera qu'il ne suffit pas comme en model-checking classique de retourner « oui, il existe une attaque » ou « non, il n'en existe pas ») ; cette sémantique est d'autre part *optimale* : il n'existe qu'une attaque minimale parmi toutes les attaques impliquant des valeurs données des variables du premier ordre commençant à un événement donné, alors même qu'il peut exister un nombre non borné d'attaques non minimales (ceci permet d'éviter de noyer l'officier de sécurité sous un nombre exagéré d'alertes, toutes liées à des attaques similaires).

Les logiques et les algorithmes sont décrits dans le papier [9]. La première logique et le premier algorithme étaient le sujet de DEA de Muriel Roger ; l'algorithme a été breveté [58]. La deuxième logique et le deuxième algorithme ont été conçus par Jean Goubault-Larrecq en 2000, sur de nouvelles idées, plus adaptées aux besoins pratiques en sécurité ainsi qu'en maintenance automatique de logiciels notamment, ainsi que sur de nouvelles idées d'algorithmes plus efficaces.

L'outil logWeaver est donc antérieur au projet SECSI, étant donné qu'il a été réalisé au GIE Dyade, puis maintenu au LSV par Jean Goubault-Larrecq. Il appartient toujours à Bull S.A. et à l'INRIA, les deux organismes qui ont formé le GIE Dyade. Une version détaillée de cet algorithme forme le sujet du rapport [39].

Julien Olivain, sur CDD au LSV pour un an sur le projet RNTL DICO, travaille depuis fin 2002 sur un nouvel outil qui prendra la suite de logWeaver, et sera plus performant, plus expressif, en un mot tirera les leçons des outils précédents. Ce nouvel outil sera refait depuis le début.

5.8. SPORE : une bibliothèque publique de protocoles cryptographiques

Participants : Florent Jacquemard [correspondant], Ralf Treinen, Hubert Comon-Lundh, Alexandre Boisseau, Véronique Cortier, (liste non limitative).

SPORE est une page Web disponible en http://www.lsv.ens-cachan.fr/spore/. Son but est de fournir une bibliothèque publique de protocoles cryptographiques, leurs différentes versions, les propriétés de sécurité qu'ils sont censés vérifier, celles qu'ils vérifient effectivement et sous quelles hypothèses, les attaques connues contre ces protocoles.

SPORE contient en particulier une liste de protocoles cryptographiques identifiés par Trusted Logic S.A. dans le cadre du projet EVA.

Une bibliothèque similaire avait été publiée en 1997 par John Clark et Jeremy Jacob (University of York), dans la seconde partie d'un survol très largement diffusé dans la communauté de la vérification de protocoles cryptographiques. En particulier, leur bibliothèque a souvent servi de base de cas d'étude pour les concepteurs d'outils de vérification automatique de protocoles.

Le but de la page SPORE est de poursuivre l'effort de Clark et de Jacob, d'une part par une mise à jour des protocoles que l'on trouve dans leur survol, et d'autre part avec l'ajout de nouvelles entrées, l'ensemble de la bibliothèque étant accessible en ligne, afin d'assurer une meilleure interactivité avec les utilisateurs et faciliter la réutilisabilité par les concepteurs d'outils.

Chaque entrée de la bibliothèque contient la description d'un protocole cryptographique (dans la syntaxe semi-formelle du papier de référence définissant la logique de Burrows, Abadi et Needham), les propriétés de sécurité que ce protocole est censé assurer ainsi que des commentaires et des liens vers des articles et pages traitant de ce protocole. On trouvera en particulier des références bibliographiques de travaux proposant des preuves formelles ou des attaques de protocoles. Les protocoles de la bibliothèque sont disponibles pour téléchargement dans plusieurs formats, incluant des formats imprimables (postscript, pdf) et des spécifications en texte des messages du protocole, à destination des concepteurs d'outils à la recherche de cas d'études.

La page SPORE, qui contient actuellement une cinquantaine de protocoles, a été conçue dans un esprit d'ouverture : les lecteurs peuvent commenter les entrées par email, entraînant une mise à jour, et surtout, peuvent soumettre de nouveaux protocoles.

6. Résultats nouveaux

6.1. Réduction de l'authentification au secret

Participants : Vincent Bernat, Hubert Comon-Lundh.

Vincent Bernat a travaillé dans le cadre de son travail de DEA, puis de thèse, sous la direction d'Hubert Comon-Lundh au problème suivant : peut-on faire vérifier des propriétés d'authentification par des outils qui savent vérifier des propriétés de secret, dans le cadre d'un modèle de Dolev-Yao ?

La question est importante, vu que la plupart des outils existant ont pour but de prouver le secret plutôt que l'authentification. Il est d'autre part intéressant pour des raisons tant théoriques que pratiques de ne pas réinventer une nouvelle technique de vérification de l'authentification.

Ce travail en est encore à un stade préliminaire. S'il semble qu'une telle réduction soit atteignable, il s'avère en revanche que la propriété d'authentification visée n'a souvent aucun intérêt, étant fausse pour tout protocole réel non trivial en mode multi-session parallèle. Il est remarquable que c'est aussi le cas d'un certain nombre d'autres propriétés d'authentification présentes dans la littérature, qui sont parfaitement raisonnables tant que l'on a un nombre totalement identifié de principaux obéissant à des rôles distincts, mais qui sont trivialement

fausses dans le cas d'un nombre quelconque de principaux jouant chaque rôle : un intrus peut en effet servir d'intermédiaire entre deux sessions du même protocole, et violer l'authentification.

6.2. Réduction du nombre d'agents pour les propriétés de sécurité

Participants : Véronique Cortier, Hubert Comon-Lundh.

Véronique Cortier a montré le résultat important suivant [36]. Soit Π un protocole cryptographique, et P une propriété de sécurité (secret, authentification notamment). S'il existe une attaque qui nécessite n principaux, alors il existe une attaque n'impliquant que deux agents.

Techniquement parlant, ce résultat suppose que chaque principal peut se parler à lui-même. Dans le cas contraire, Véronique Cortier a montré qu'il existe une attaque n'impliquant que $k + 1$ agents, où k est le nombre de participants honnêtes dans *une* session du protocole.

En particulier, on en déduit une réduction des propriétés de sécurité à des cas où l'une des sources d'infini du problème de vérification, le nombre possiblement infini d'identités de principaux, est fini et borné par une quantité calculable et en général faible (4 en général, même dans le cas où les principaux n'ont pas le droit de se parler à eux-mêmes). Ceci ne supprime bien sûr pas toutes les sources d'infini du problème de vérification, qui reste indécidable.

Ceci montre aussi que l'abstraction consistant à identifier les identités des agents honnêtes (resp. de ceux qui jouent le même rôle), et d'identifier toutes les identités des agents malhonnêtes, est en fait exacte.

6.3. Signature électronique de contrats

Participants : Alexandre Boisseau, Hubert Comon-Lundh.

Alexandre Boisseau a travaillé cette année sur les protocoles de signature électronique de contrats, et plus spécifiquement sur le protocole GJM, dû à J.A. Garay, M. Jakobsson, et P. MacKenzie [53] (http://www.lsv.ens-cachan.fr/spore/gjm.html). Il s'agit d'un protocole censé permettre à deux principaux A et B n'ayant pas nécessairement confiance l'un en l'autre de signer un contrat, en présence d'un tiers de confiance, appelé le TTP (« trusted third party »). Les propriétés de sécurité ne sont plus les propriétés usuelles de secret ou d'authentification, mais de :

- équité : si B a obtenu le contrat signé par A, alors A peut obtenir le contrat signé par B et réciproquement ;
- complétude : si le protocole de signature se déroule correctement, aucun protocole annexe, typiquement d'annulation de la procédure, n'est nécessaire ;
- absence d'abus de confiance (« abuse-freeness ») : ni A ni B ne peut prouver à un participant extérieur qu'il connaît l'issue de la négociation ; ceci a pour but d'interdire à l'un des participants, disons A, un avantage déloyal qui lui permettrait d'obtenir de meilleures conditions auprès d'un troisième participant C à qui il pourrait montrer qu'il va avoir un contrat signé avec B ;
- audit du TTP : si le TTP se conduit de façon malhonnête, alors A et B peuvent le prouver.

S. Kremer et J.-F. Raskin (Université libre de Bruxelles, Belgique) ont donné une modélisation du problème en utilisant la logique ATL (Alternating-Time Temporal Logic) et un modèle de systèmes de transitions alternées, dûs à Alur, Henzinger, et Kupfermann.

Le modèle d'Alexandre Boisseau étend ce dernier en en enlevant un certain nombre de simplifications dont la justification restait à trouver. Le modèle d'A. Boisseau, de ce fait, est plus complet, mais est trop gros pour les model-checkers actuels. A. Boisseau a montré, en collaboration avec S. Kremer et J.-F. Raskin, qu'on pouvait ramener le modèle complet au modèle simplifié par une technique automatique d'abstraction [31].

Il s'agit bien entendu de travaux commencés antérieurement à la création de SECSI.

6.4. Automates d'arbres alternants bidirectionnels modulo AC

Participants : Kumar Neeraj Verma, Jean Goubault-Larrecq.

Il est parfois besoin de raisonner modulo une *théorie équationnelle*, c'est-à-dire en considérant que certaines égalités entre termes doivent être valides. Dans le modèle de Dolev-Yao standard des protocoles cryptographiques, deux termes (deux messages) sont égaux si et seulement s'ils sont deux termes identiques, ce qui cadre bien avec une vision de cryptographie idéale. Cependant, un certain nombre de primitives cryptographiques vérifient des équations supplémentaires.

Or, il existe des cas où les primitives cryptographiques intéressantes obéissent naturellement à des équations spécifiques. C'est le cas notamment de l'exponentielle modulaire, telle qu'utilisée par exemple dans le protocole de Diffie-Hellman [51]. Une façon de modéliser ceci est de considérer que l'on dispose d'un symbole de fonction e à un argument, l'exponentielle, et d'une opération $\oplus$, qui est *associative* et *commutative*, et a un élément neutre 0. Autrement dit, on considère des messages qui sont des termes construits à l'aide des symboles de fonction e, $\oplus$ et 0, et modulo la théorie équationnelle AC1 :

$$(M_1 \oplus M_2) \oplus M_3 = M_1 \oplus (M_2 \oplus M_3) \qquad M_1 \oplus M_2 = M_2 \oplus M_1 \qquad M \oplus 0 = M$$

Ceci est décrit en détail dans [41], dans le cadre de la modélisation du protocole d'accord de clé initial IKA.1 [62] (anciennement GDH.2). Ce protocole est utilisé pour créer une clé de groupe initial dans la suite de protocoles CLIQUES.

Cette modélisation nécessite une variante des automates d'arbres classiques, les automates d'arbres *bidirectionnels* modulo AC1, et de tester la vacuité d'une intersection finie de tels automates. Kumar Neeraj Verma a montré que le problème de la vacuité des automates d'arbres bidirectionnels *alternants* modulo AC1 est indécidable, mais en absence d'alternance, c'est-à-dire de clauses intersection, et de clauses dites descendantes conditionnelles, ce problème est de nouveau décidable ; en fait, même le problème de la vacuité de l'intersection d'un nombre fini de langages définis sans alternance est décidable, ce qui permet de traiter le cas du protocole IKA.1. L'algorithme de décision est cependant non trivial, et de complexité non primitive récursive.

6.5. Relations logiques et équivalence observationnelle en présence de création de noms

Participants : Yu Zhang, David Nowak, Jean Goubault-Larrecq.

Une ligne de recherche originale a été lancée cette année sur la vérification de protocoles cryptographiques, à l'occasion de la venue de Zhang Yu en stage de DEA au LSV. Il s'agit d'étudier une approche différente, originellement due à Sumii et Pierce [63], fondée sur la notion d'équivalence observationnelle et plus précisément de relations logiques utilisées pour prouver l'équivalence observationnelle dans un λ-calculs avec création de noms et types abstraits, le *λ-calcul cryptographique*.

On sait depuis Abadi et Gordon [1] que des propriétés telles que le secret ou l'authentification ont des définitions élégantes en termes d'équivalence observationnelle de processus. Le travail de Sumii et Pierce [63] est similaire, mais se fonde sur une variante du λ-calcul avec création dynamique de noms et types abstraits (ces derniers représentant les chiffrements), et des techniques issues de la théorie du λ-calcul de relations logiques pour prouver l'équivalence observationnelle. Ceci peut permettre de trouver de nouvelles techniques algorithmiques originales de vérification de protocoles cryptographiques.

Or, les relations logiques de Sumii et Pierce sont encore relativement primaires, et ne déclareront notamment équivalents deux λ-termes que s'ils créent les mêmes noms dans le même ordre notamment. Or la création de noms (ou nonces) est un problème épineux dans toute méthode de vérification de protocoles cryptographiques.

Lorsqu'on se restreint à la problématique de l'équivalence observationnelle en présence de création de noms, une amélioration considérable est fournie par le travail antérieur de Pitts et Stark [57][61]. Le travail sur les relations logiques pour le méta-langage de Moggi qui a été effectué par Jean Goubault-Larrecq, David

Nowak, et Slawek Lasota [29] trouve ici une application naturelle (et prise en exemple dans l'article) : il suffit de choisir comme monade du méta-langage de Moggi une monade de création dynamique de noms, et la construction de [29] fournit immédiatement une notion de relation logique proche de celle de Pitts et Stark. Alors que la relation logique de Pitts et Stark est ad hoc, ne fonctionne que sur un λ-calcul en appel par valeur, et dépend fondamentalement de la normalisation de ce calcul, la relation logique issue de [29] est définie dans un cadre général - et s'accomode donc au passage fort bien de la récursion.

Zhang Yu a montré au cours de son stage de DEA que la relation logique de [29] est strictement plus faible que celle de Pitts et Stark, une fois ramenée au λ-calcul par valeur que ces derniers considèrent. Il a aussi montré comment obtenir une relation logique générale, équivalente à celle de Pitts et Stark dans le cas de leur langage. Aujourd'hui en thèse avec David Nowak et Jean Goubault-Larrecq, il travaille sur l'extension au λ-calcul cryptographique complet de Sumii et Pierce. On espère ainsi à terme trouver des techniques innovantes de preuve de sécurité de protocoles cryptographiques.

Une dernière remarque est de rigueur : les relations logiques pour le méta-langage de Moggi sont remarquablement similaires aux bisimulations, plus classiques dans le monde des calculs de processus ; cependant, pour l'instant la construction de [29] semble incomparable en généralité avec la construction classique de bisimulations par spans de morphismes ouverts [55], et les rapports avec les constructions classiques de bisimulations en algèbres de processus sont encore peu clairs.

8. Actions régionales, nationales et internationales

8.1. Actions Nationales

Toutes les actions listées ici sont antérieures à SECSI, et ne sont donc décrites qu'à titre indicatif. Il faut y voir ici, bien sûr, le fait que SECSI est un projet jeune.

8.1.1. Projet RNTL EVA (antérieur à SECSI)

Participants : Hubert Comon-Lundh, Jean Goubault-Larrecq, Florent Jacquemard, Véronique Cortier, Vincent Bernat, Alexandre Boisseau.

Ce projet exploratoire, financé par le réseau national des technologies du logiciel (RNTL), regroupe Trusted Logic S.A. (leader, startup de l'INRIA), le LSV, et Verimag. Notifié à l'automne 2000, il a une durée de 3 ans.

Ce projet est en particulier antérieur à la création de SECSI.

Le titre du projet, EVA, signifie « Explication et Vérification Automatique des protocoles cryptographiques ». Son but est de développer des méthodes de vérification automatique, et non pas seulement assistée par ordinateur, de protocoles cryptographiques. Les propriétés à vérifier sont pour l'instant diverses notions de confidentialité et d'authentification. On insiste particulièrement sur la vérification en mode multi-sessions parallèles, c'est-à-dire dans lequel il existe un nombre potentiellement non borné de principaux (participants) obéissant à chaque rôle.

Le « E » de EVA signifie « explication », et est une composante importante de ce projet. L'explication recouvre plusieurs thèmes de recherche possible. Un thème consiste à extraire soit un argument lisible de correction à partir d'une preuve de protocole, soit une description lisible d'une attaque (ou d'une trace indicative d'une attaque potentielle). Un thème plus actif est, dans le premier cas, d'extraire un script de preuve de protocole rejouable sous un outil de preuve tel que Coq (du projet LogiCal, UR Futurs).

Référence : http://www-eva.imag.fr/index_eva.html. Référence spécifique au LSV et au projet SECSI : http://www.lsv.ens-cachan.fr/~goubault/EVA.html.

8.1.2. Projet RNTL DICO (antérieur à SECSI)

Participants : Jean Goubault-Larrecq, Stéphane Demri, Julien Olivain, Muriel Roger.

Ce projet exploratoire, financé par le réseau national des technologies du logiciel (RNTL), regroupe la PME NetSecure Software (leader), France Télécom R&D, le LSV, l'IRISA, l'ONERA/DTIM, la FERIA/IRIT, SupÉlec. Notifié en décembre 2001, il a une durée de 2 ans.

Ce projet est en particulier antérieur à la création de SECSI.

Le titre du projet, DICO, signifie « Détection d'Intrusions COopérative ». Il s'agit donc en premier lieu d'un projet de détection d'intrusions, mêlant approches par signatures (comme celle développée chez SECSI [9]), approches comportementales (réseaux bayésiens par exemple), et corrélation d'alertes. Ce dernier point est la clé de voûte du projet, et consiste à fusionner ou inférer des alertes venant de différents outils, voire de différents serveurs.

Référence : http://dico.netsecuresoftware.com/. Référence spécifique au LSV et au projet SECSI : http://www.lsv.ens-cachan.fr/~goubault/DICO.html.

8.1.3. ACI cryptologie « VERNAM » (antérieure à SECSI)

Participants : Hubert Comon-Lundh, Jean Goubault-Larrecq, Ralf Treinen, Vincent Bernat, Véronique Cortier, Kumar Neeraj Verma.

L'ACI « VERNAM » regroupe l'université de Provence/projet INRIA MIMOSA, le LSV, et le projet INRIA PROTHEO, sur le thème de la recherche de sous-classes décidables de protocoles cryptographiques, en relation avec des sous-classes décidables de la logique du premier ordre. Il s'agit d'une action concertée incitative « cryptologie » du ministère de la recherche, qui a démarré à l'automne 2000 pour 3 ans.

Ce projet est en particulier antérieur à la création de SECSI.

Référence spécifique au LSV et au projet SECSI : http://www.lsv.ens-cachan.fr/~goubault/VERNAM.html.

8.1.4. ACI cryptologie « PSI-Robuste » (antérieure à SECSI)

Participants : Jean Goubault-Larrecq, Stéphane Demri, Fabrice Parrennes, Julien Olivain, Muriel Roger, Yu Zhang.

L'ACI « PSI-Robuste » est une action concertée incitative de cristallisation au LSV, sur les thèmes de la protection des systèmes informatiques. Il porte plus spécifiquement sur les approches de détection d'intrusion, d'analyse statique de code, et l'interaction entre les deux approches. Il s'agit d'une action concertée incitative « cryptologie » du ministère de la recherche, qui a démarré à l'automne 2001 pour 3 ans.

Ce projet est en particulier antérieur à la création de SECSI.

Référence spécifique au LSV et au projet SECSI : http://www.lsv.ens-cachan.fr/~goubault/PSIrobuste.html.

8.1.5. ACI jeunes chercheurs « Sécurité informatique, protocoles cryptographiques et détection d'intrusions » (antérieure à SECSI)

Participants : Jean Goubault-Larrecq, Fabrice Parrennes, Stéphane Demri, Alexandre Boisseau, Véronique Cortier, Muriel Roger, Yu Zhang.

Il s'agit d'une ACI « jeunes chercheurs » attribuée à Jean Goubault-Larrecq, qui fournit un financement pour 3 ans, à partir de l'automne 2001. Elle a servi et sert encore à développer les thèmes de vérification de protocoles cryptographiques et de détection d'intrusions principalement, ainsi que d'analyse statique de code. Elle finance notamment Fabrice Parrennes, en CDD pour un an (septembre 2002-septembre 2003).

Ce projet est en particulier antérieur à la création de SECSI.

Référence spécifique au LSV et au projet SECSI : http://www.lsv.ens-cachan.fr/~goubault/JC.html.

9. Diffusion des résultats

9.1. Enseignement

Jean Goubault-Larrecq a donné un cours de « Logique et Informatique », portant sur le λ-calcul, les langages fonctionnels et les preuves, au second semestre 2001-2002. Il s'agit d'un cours de première année commun entre les magistères MMFAI de l'ENS et Math-Info de l'ENS Cachan ; le volume est de 20 h. de cours. Les TD ont été données par Maribel Fernández, ENS.

Jean Goubault-Larrecq a donné un cours de démonstration automatique de théorèmes au DEA « Programmation » ; volume : 16 h. Ian Mackie (CNRS, LIX), y a participé pour 4 heures supplémentaires.

Jean Goubault-Larrecq a donné un cours d'analyse statique de code au DESS « Développement de Logiciels Sûrs » ; volume : 20 h.

Jean Goubault-Larrecq a donné un cours de programmation en première année au magistère STIC de l'ENS Cachan, au premier semestre 2002-2003 ; volume : 40 h. équivalent TD.

Hubert Comon-Lundh a donné un cours de logique et automates au DEA « Programmation » ; volume : 20 h.

Hubert Comon-Lundh et Ralf Treinen ont donné un cours de logique en première année du magistère STIC de l'ENS Cachan ; volume total : 80 h. équivalent TD, répartis de septembre 2002 à janvier 2003.

Stéphane Demri a donné un cours sur la complexité du model-checking au DEA « Algorithmique » ; volume : 20 h.

Véronique Cortier a donnée des TP de C++ au premier semestre 2002 en première année du magistère d'ingénierie électrique de l'ENS Cachan, et des TD de calculabilité au deuxième semestre 2002 en première année du magistère de mathématiques et informatique (MathInfo) de l'ENS Cachan.

Alexandre Boisseau a donné des TD d'algorithmique en première année a l'ISTY, une école d'ingénieurs en informatique interne a l'Université de Versailles St-Quentin en Yvelines ; il y est moniteur. Le volume est de 32 h. au second semestre 2001-2002, et 32 h. au premier semestre 2002-2003. Il a aussi rédigé un recueil d'exercices partiellement corrigés.

9.2. Encadrement

Jean Goubault-Larrecq a encadré, avec David Nowak (CNRS, LSV), le stage de DEA « Programmation » de Zhang Yu [46], portant sur les relations logiques et le λ-calcul cryptographique de Sumii et Pierce [63]. Ce travail se poursuit dans une thèse à l'ENS Cachan, de titre « Critères décidables et/ou complets de sécurité des protocoles cryptographiques », financé sur une bouse MENRT sur l'ACI « cryptologie ».

Jean Goubault-Larrecq a supervisé le stage de DEA « Programmation » de Mathieu Baudet chez Gemplus, portant sur « contrôle de ressource et évitement des interblocages sur la mémoire » (encadrant : Antoine Galland, GemPlus). Il s'agit d'optimiser les allocations mémoire sur les architectures multi-applications de cartes à microprocesseur.

Hubert Comon-Lundh a encadré le stage de DEA « Algorithmique » de Vincent Bernat, sur le sujet de la réduction de la propriété d'authentification à celle de secret dans les protocoles cryptographiques. Ce stage se poursuit en une thèse à l'ENS Cachan.

9.3. Participations à des jurys de thèse ou d'habilitation

Hubert Comon-Lundh a participé comme rapporteur au jury d'habilitation de Joachim Niehren, université de Saarbrücken, juillet 2002.

Hubert Comon-Lundh a participé comme rapporteur au jury de thèse de Benjamin Monate, université Paris 11, janvier 2002.

Jean Goubault-Larrecq a participé comme directeur de thèse au jury d'Eva Rose, université Paris 7, septembre 2002 : « Vérification de code d'octet de la machine virtuelle Java : formalisation et implantation ».

Jean Goubault-Larrecq a participé comme rapporteur au jury de thèse de Fabrice Parrennes, université Paris 6, mai 2002.

Jean Goubault-Larrecq a participé comme examinateur aux jurys de thèse de Sébastien Praud, université Paris 11 Orsay, février 2002, et d'Olivier Brunet, université Joseph Fourier, Grenoble, octobre 2002.

9.4. Participation à des comités de programme de conférences, ou comités de rédaction de revues

Jean Goubault-Larrecq a édité un numéro spécial sur les modèles et méthodes de vérification de protocoles cryptographiques du *Journal of Telecommunications and Information Technology*, à paraître en décembre 2002

[22]. Ce journal international est publié par l'Instytut Łączności (Institute of Telecommunications), à Varsovie en Pologne. Il s'agit d'un journal international, donc en anglais.

Jean Goubault-Larrecq a participé comme président du comité de programme du 1er workshop international sur la sécurité des communications sur Internet/1st International Workshop on Security of Communications on the Internet (SECI'02), Tunis, Tunisie. (Bilingue anglais/français.) Les actes sont publiés par l'INRIA, collection didactique [12].

Hubert Comon-Lundh a fait partie des comités de programme des conférences Constraint Programming (CP'2002) et Infinity 2002 (workshop satellite d'ICALP).

Jean Goubault-Larrecq a fait partie des comités de programme des conférences 10th International Conference on Theorem Proving with Analytic Tableaux and Related Methods (Tableaux 2002, partie de FLoC 2002, Copenhague, Danemark), 18th International Conference on Automated Deduction (CADE 2002, partie de FLoC 2002, Copenhague, Danemark), 9th International Conferences on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2002, Tbilissi, Géorgie), 2nd International Workshop on Intuitionistic Modal Logics and Applications (IMLA 02, partie de FLoC 2002, Copenhague, Danemark).

Véronique Cortier a fait partie du comité de programme du workshop Foundations of Computer Security (FCS'02), workshop satellite de FLoC 2002, Copenhague, Danemark, 2002.

9.5. Participation à des colloques, séminaires, invitations

Hubert Comon-Lundh a organisé les journées en l'honneur de Zohar Manna, à l'occasion de la remise de son doctorat Honoris Causa de l'ENS Cachan, mars 2002. Voir http://www.lsv.ens-cachan.fr/manna-dhc.php.

Jean Goubault-Larrecq a organisé la demi-journée « sécurité du code », Cachan, septembre 2002. Ceci coincidait avec la soutenance de thèse d'Eva Rose sur la vérification à faible empreinte mémoire de bytecode Java, qui en a été le point culminant.

Jean Goubault-Larrecq a donné un exposé à la réunion de l'ACI cryptologie « VERNAM », au LORIA à Nancy en janvier 2002, et un exposé invité aux journées Cachan-Bordeaux, Cachan, mars 2002, les deux exposés sur un travail avec Kumar Neeraj Verma : « Alternating 2-Way AC-Tree Automata » ; un exposé invité au séminaire CATIA, université de Montpellier, mars 2002 : « Sur la géométrie des preuves de la logique S4 intuitionniste » ; un séminaire invité aux rencontres DCSSI-INRIA Logical, Rocquencourt, septembre 2002 : « Vérification de protocoles cryptographiques : la logique à la rescousse ! » ; et le même séminaire à l'IIE (Institut d'Informatique d'Entreprise, Évry), décembre 2002.

Hubert Comon-Lundh a donné un exposé invité à CiAD (Complexity in automated deduction, workshop satellite de FloC) : « How difficult is it to retrieve a secret ? » ; un exposé invité aux journées du LIX, septembre 2002 : « Sécurité des protocoles cryptographiques : au-delà du chiffrement parfait » ; un séminaire invité au groupe de travail PPS, université Paris 7, janvier 2002 : « Vérification de protocoles cryptographiques ».

Stéphane Demri a donné un séminaire au RWTH, Aachen, Allemagne, février 2002, et à l'université McGill, Montréal, Canada : « A parametric analysis of the state explosion problem in model-checking ».

Véronique Cortier a donné un exposé à la réunion de l'ACI cryptologie « VERNAM », au LORIA à Nancy en janvier 2002, de titre « un pont entre le spi-calcul et les systèmes de transitions » ; un exposé invité à la journée SLOVO en l'honneur de Moshe Vardi, à l'occasion de la remise de son doctorat Honoris Causa de l'Université d'Orléans, septembre 2002, titre : « Security properties : Two agents are enough » ; un exposé au séminaire SIF (Séminaire d'Informatique Fondamentale) au LORIA à Nancy, octobre 2002 : « Vérification automatique des protocoles cryptographiques » ; un exposé au séminaire 68NQRT, IRISA, Rennes, octobre 2002 : « Vérification des protocoles cryptographiques ».

Alexandre Boisseau a effectué des visites à l'unversité libre de Bruxelles, où il a travaillé avec S. Kremer et J.-F. Raskin sur les formalisations de signatures de contrats électroniques en avril, mai, octobre et novembre 2002.

Florent Jacquemard a participé aux journées nationales du RNTL, Toulouse, octobre 2002 : « projet EVA : Explication et Vérification Automatique des protocoles cryptographiques », présentation de poster.

Ralf Treinen a passé une semaine en octobre-novembre 2002 à l'Universität des Saarlandes, Saarbrücken, Allemagne, pour continuer ses travaux communs avec Joachim Niehren et Tim Priesnitz sur les contraintes de sous-typage.

Zhang Yu a participé aux cérémonies du centenaire de l'université de Nanjing, mai 2002. Il a participé à l'école de printemps « Sémantique des langages de programmation », Agay, mars 2002.

9.6. Maintenance de listes de problèmes ouverts

Ralf Treinen maintient, avec Nachum Dershowitz (université de Tel-Aviv, Israël), la liste des problèmes ouverts de la série de conférences « Rewriting Techniques and Applications » (RTA). La liste est disponible à l'adresse http://www.lsv.ens-cachan.fr/rtaloop/.

10. Bibliographie

Bibliographie de référence

[1] M. ABADI, A. GORDON. *A Calculus for Cryptographic Protocols : The Spi Calculus.* in « Information and Computation », numéro 1, volume 148, janvier, 1999, pages 1-70.

[2] H. COMON, V. CORTIER, J. MITCHELL. *Tree automata with one memory, set constraints and ping-pong protocols.* in « Proc. 28th Int. Coll. Automata, Languages, and Programming (ICALP'01), Crete, Greece, July 2001 », volume 2076, Springer, pages 682-693, 2001.

[3] E. A. EMERSON. *Temporal and Modal Logic.* éditeurs J. VAN LEEUWEN., in « Handbook of Theoretical Computer Science », volume B, Elsevier, 1990, chapitre 16, pages 996-1072.

[4] A. FREIER, P. KARLTON, P. KOCHER. *The SSL protocol. Version 3.0.*. http://home.netscape.com/eng/ssl3/.

[5] J. GOUBAULT-LARRECQ. *A method for automatic cryptographic protocol verification (extended abstract).* in « Proc. Int. Workshop on formal methods in parallel programming, techniques and applications », série Lecture Notes in Computer Science, volume 1800, Springer Verlag, pages 977-984, 2000.

[6] G. LOWE. *An Attack on the Needham-Schroeder Public-Key Authentication Protocol.* in « Information Processing Letters », numéro 3, volume 56, 1996, pages 131-133.

[7] LSV. *System and Software Verification. Model-Checking Techniques and Tools.* Springer Verlag, 2001.

[8] L. PAULSON. *The inductive approach to verifying cryptographic protocols.* in « Journal of Computer Security », volume 6, 1998, pages 85-128.

[9] M. ROGER, J. GOUBAULT-LARRECQ. *Log auditing through model-checking.* in « Proc. 14th IEEE Computer Security Foundations Workshop », 2001.

[10] R. WILHELM, M. SAGIV, T. REPS. *Shape Analysis.* in « Proc. of the 9th Int. Conf. on Compiler Construction (CC'2000) », Springer-Verlag LNCS, 2000.

Livres et monographies

[11] S. DEMRI, E. ORŁOWSKA. *Incomplete Information : Structure, Inference, Complexity.* série EATCS Monographs, Springer, 2002, http://www.springer.de/cgi/svcat/search_book.pl?isbn=3-540-41904-7.

[12] *Actes du 1er workshop international sur la sécurité des communications sur Internet (SECI'02), Tunis, Tunisie, Sep. 2002.* éditeurs J. GOUBAULT-LARRECQ., INRIA, 2002, http://www.lsv.ens-cachan.fr/~goubault/SECI-02/Final/actes-seci02/index.html.

Articles et chapitres de livre

[13] H. COMON, Y. JURSKI. *Counter Automata, Fixed Points and Additive Theories.* in « Theoretical Computer Science », 2002, à paraître.

[14] H. COMON, P. NARENDRAN, R. NIEUWENHUIS, M. RUSINOWITCH. *Deciding the Confluence of Ordered Term Rewrite Systems.* in « ACM Trans. Computational Logic », 2002, à paraître.

[15] H. COMON, V. SHMATIKOV. *Is it Possible to Decide whether a Cryptographic Protocol is Secure or not ?.* in « Journal of Telecommunications and Information Technology », volume 4, décembre, 2002.

[16] V. CORTIER. *About the Decision of Reachability for Register Machines.* in « Theoretical Informatics and Applications », 2002, à paraître, probablement en 2003.

[17] S. DEMRI. *A polynomial space construction of tree-like models for logics with local chains of modal connectives.* in « Theoretical Computer Science », 2002, http://www.lsv.ens-cachan.fr/Publis/PAPERS/demri-tcs02.ps, à paraître.

[18] S. DEMRI, R. GORÉ. *Cut-free display calculi for nominal tense logics.* in « Journal of Logic and Computation », numéro 5, volume 12, 2002, http://www.lsv.ens-cachan.fr/Publis/PAPERS/DemGor-jlc02-1.ps, à paraître.

[19] S. DEMRI, R. GORÉ. *Theoremhood Preserving Maps Characterising Cut Elimination for Modal Provability Logics.* in « Journal of Logic and Computation », numéro 5, volume 12, 2002, pages 861-884, http://www.lsv.ens-cachan.fr/Publis/PAPERS/DemGor-jlc02-2.ps.

[20] S. DEMRI, P. SCHNOEBELEN. *The Complexity of Propositional Linear Temporal Logics in Simple Cases.* in « Information and Computation », numéro 1, volume 174, 2002, pages 84-103, http://www.lsv.ens-cachan.fr/Publis/PAPERS/DS-ICOMP2001.ps.

[21] J. GOUBAULT-LARRECQ. *Extensions of Valuations.* in « Math. Struct. in Comp. Science », 2002, à paraître. Aussi disponible en rapport LSV LSV-02-17, novembre 2002..

[22] J. GOUBAULT-LARRECQ. *Special Issue on Models and Methods for Cryptographic Protocol Verification.* in « Journal of Telecommunications and Information Technology », volume 4, décembre, 2002.

[23] J. GOUBAULT-LARRECQ. *Sécurité, modélisation et analyse de protocoles cryptographiques.* in « Phœbus, la

revue de la sûreté de fonctionnement », volume 20, 2002, http://www.lsv.ens-cachan.fr/Publis/PAPERS/GL-Phoebus2002.doc, Numéro spécial sur la sécurité des systèmes d'information.

[24] J. GOUBAULT-LARRECQ, E. GOUBAULT. *On the Geometry of Intuitionistic S4 Proofs.* in « Homology, Homotopy and Applications », 2002, http://www.lsv.ens-cachan.fr/Publis/PAPERS/S4G.ps, Accepté en version finale, à paraître.

Communications à des congrès, colloques, etc.

[25] S. DEMRI, D. D'SOUZA. *An automata-theoretic approach to Constraint LTL.* in « Proc. 22nd Conf. Found. of Software Technology and Theor. Comp. Sci. (FST&TCS'2002), Kanpur, India, Dec. 2002 », série Lecture Notes in Computer Science, volume 2556, Springer, 2002, http://www.cmi.ac.in/techreps/ps/tcs-02-1.ps.gz, à paraître.

[26] S. DEMRI. *Modal Logics with Weak Forms of Recursion : PSPACE specimens.* in « Advances in Modal Logics, selected papers from 3rd Workshop on Advances in Modal Logics (AIML'2000), Leipzig, Germany, Oct. 2000 », World Scientific, éditeurs M. DE RIJKE, H. WANSING, F. WOLTER, M. ZAKHARYASCHEV., pages 113-138, 2002, http://www.lsv.ens-cachan.fr/Publis/PAPERS/sd-aiml00.ps.

[27] S. DEMRI, F. LAROUSSINIE, P. SCHNOEBELEN. *A Parametric Analysis of the State Explosion Problem in Model Checking (Extended Abstract).* in « Proc. 19th Ann. Symp. Theoretical Aspects of Computer Science (STACS'2002), Antibes Juan-les-Pins, France, Mar. 2002 », série Lecture Notes in Computer Science, volume 2285, Springer, pages 620-631, 2002, http://www.lsv.ens-cachan.fr/Publis/PAPERS/DLS-stacs2002.ps.

[28] J. GOUBAULT-LARRECQ. *Higher-Order Positive Set Constraints.* in « CSL'02 », Springer Verlag LNCS, 2002, Disponible en rapport de recherche LSV-02-6, Lab. Specification and Verification, ENS de Cachan, juillet 2002, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2002-6.rr.ps..

[29] J. GOUBAULT-LARRECQ, S. LASOTA, D. NOWAK. *Logical Relations for Monadic Types.* in « CSL'02 », Springer-Verlag LNCS, 2002.

[30] J. GOUBAULT-LARRECQ. *Vérification de protocoles cryptographiques : la logique à la rescousse !.* in « Actes du 1er workshop international sur la sécurité des communications sur Internet (SECI'02), Tunis, Tunisia, Sep. 2002 », INRIA, éditeurs J. GOUBAULT-LARRECQ., pages 119-152, 2002, http://www.lsv.ens-cachan.fr/Publis/PAPERS/JGL-seci.ps.

Rapports de recherche et publications internes

[31] A. BOISSEAU. *Signatures électroniques de contrats.* Research Report, numéro LSV-02-4, Lab. Spécification et Vérification, ENS de Cachan, avril, 2002, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2002-4.rr.ps.

[32] S. CHHABRA. *Extraction of Intrusion Detection Signatures from Failed Proofs Of Cryptographic Protocols : Work in Progress.* rapport technique, LSV/CNRS UMR 8643 & ENS Cachan, 2002.

[33] V. CORTIER. *Observational equivalence and trace equivalence in an extension of Spi-calculus. Application to cryptographic protocols analysis. Extended version.* Research Report, numéro LSV-02-3,

Lab. Specification and Verification, ENS de Cachan, Cachan, France, mars, 2002, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2002-3.rr.ps, 33 pages.

[34] J. GOUBAULT-LARRECQ. *Programmation.* Magistère STIC, ENS Cachan, Cachan, 2002, http://www.lsv.ens-cachan.fr/~goubault/cours.html.

[35] J. GOUBAULT-LARRECQ, K. N. VERMA. *Alternating Two-Way AC-Tree Automata.* Research Report, numéro LSV-02-1, Lab. Spécification and Vérification, ENS de Cachan, Cachan, France, janvier, 2002, 15 pages. Version étendue et corrigée, septembre 2002 ; disponible auprès des auteurs goubault@lsv.ens-cachan.fr, verma@lsv.ens-cachan.fr..

[36] H. COMON-LUNDH, V. CORTIER. *Security Properties : Two Agents Are Sufficient.* Research Report, numéro LSV-02-10, Lab. Specification and Verification, ENS de Cachan, Cachan, France, août, 2002, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2002-10.rr.ps, 26 pages.

[37] J. GOUBAULT-LARRECQ. *A Note on the Completeness of Certain Refinements of Resolution.* Research Report, numéro LSV-02-8, Lab. Specification and Verification, ENS de Cachan, Cachan, France, juillet, 2002, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2002-8.rr.ps, 16 pages.

[38] J. GOUBAULT-LARRECQ. *Reading Notes : Why is Cpo Cocomplete ?.* Research Report, numéro LSV-02-15, Lab. Specification and Verification, ENS de Cachan, Cachan, France, octobre, 2002, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2002-15.rr.ps, 8 pages.

[39] J. GOUBAULT-LARRECQ. *Un Algorithme pour l'Analyse de Logs.* Research Report, numéro LSV-02-18, Lab. Specification and Verification, ENS de Cachan, Cachan, France, novembre, 2002, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2002-18.rr.ps, 33 pages.

[40] J. GOUBAULT-LARRECQ. *SKInT Labels.* Research Report, numéro LSV-02-7, Lab. Specification and Verification, ENS de Cachan, Cachan, France, juillet, 2002, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2002-7.rr.ps, 15 pages.

[41] J. GOUBAULT-LARRECQ, K. N. VERMA. *Alternating Two-Way AC-Tree Automata.* Research Report, numéro LSV-02-11, Lab. Specification and Verification, ENS de Cachan, Cachan, France, septembre, 2002, http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/rr-lsv-2002-11.rr.ps, 21 pages.

Divers

[42] H. COMON. *How Difficult is it to Retrieve a Secret ?.* Invited talk. Workshop on Complexity in Automated Deduction, FloC, Copenhaguen, juillet, 2002.

[43] V. CORTIER. *Outil de vérification SECURIFY.* Rapport numéro 7 du projet RNTL EVA, mai, 2002, http://www.lsv.ens-cachan.fr/~cortier/EVA-TR7.pdf, 6 pages.

[44] J. GOUBAULT-LARRECQ. *L'outil Csur.* 2002, http://www.lsv.ens-cachan.fr/~goubault/Csur/csur.html, Outil distribuable gratuitement, sous conditions : certains modules ne peuvent être distribués qu'en version compilée, l'outil servant de projet du cours « Analyse statique de code » du DESS « Développement de Logiciels Sûrs ». En OCaml, 12648 lignes..

[45] J. GOUBAULT-LARRECQ. *The EVA Parser and Translator.* 2002, http://www.lsv.ens-cachan.fr/~goubault/EVA.html, Démarré en 2001. Écrit en C (1327 lignes), OCaml (361 lignes), HimML (3454 lignes)..

[46] Y. ZHANG. *Logical Relations For Names.* Mémoire de DEA, DEA Programmation, Paris, septembre, 2002, http://www.lsv.ens-cachan.fr/Publis/PAPERS/ZY-dea02.ps, 30 pages.

[47] J. GOUBAULT-LARRECQ. *Outils CPV et CPV2.* Rapport numéro 8 du projet RNTL EVA, mai, 2002, 7 pages.

[48] J. GOUBAULT-LARRECQ, J.-P. POUZOL, S. DEMRI, L. MÉ, P. CARLE. *Langages de Détection d'Attaques par Signatures.* Sous-projet 3, livrable 1 du projet RNTL DICO. Version 1, juin, 2002, 30 pages.

Bibliographie générale

[49] J. CLARK, J. JACOB. *A survey of authentication protocol literature : Version 1.0..* 1997, Available via http://www.cs.york.ac.uk/~jac/papers/drareview.ps.gz.

[50] V. CORTIER, J. MILLEN, H. RUESS. *Proving Secrecy is Easy Enough.* in « Proc. 14th IEEE Computer Security Foundations Workshop », 2001.

[51] W. DIFFIE, M. HELLMAN. *New Directions in Cryptography.* in « IEEE Transactions on Information Theory », numéro 6, volume IT-22, novembre, 1976, pages 644-654.

[52] A. FREIER, P. KARLTON, P. KOCHER.. *The SSL protocol. Version 3.0..* http://home.netscape.com/eng/ssl3/.

[53] J. A. GARAY, M. JAKOBSSON, P. MCKENZIE. *Abuse-Free Optimistic Contract Signing.* in « Advances in Cryptology : Proceedings of Crypto'99 », série Lecture Notes in Computer Science, volume 1666, Springer Verlag, pages 449-466, 1999.

[54] J. GOUBAULT-LARRECQ. *An Introduction to logWeaver (v2.8).* GIE Dyade & LSV, septembre, 2001, Disponible sur la page Web du projet DICO au LSV, http://www.lsv.ens-cachan.fr/~goubault/DICO/tutorial.ps, 39 pages.

[55] A. JOYAL, M. NIELSEN, G. WINSKEL. *Bisimulation and Open Maps.* in « Proc. 8th Annual Symposium on Logic in Computer Science (LICS'93) », pages 418-427, 1993, Voir aussi *Information and Computation*, vol. 127, 1996, pages 164-185.

[56] G. LOWE. *An Attack on the Needham-Schroeder Public-Key Authentication Protocol.* in « Information Processing Letters », numéro 3, volume 56, 1996, pages 131-133.

[57] A. PITTS, I. STARK. *Observable Properties of Higher Order Functions that Dynamically Create Local Names, or : What's new ?.* in « MFCS'93 », Springer-Verlag LNCS 711, pages 122-141, 1993, http://www.dcs.ed.ac.uk/~stark/publications/obsphown.html.

[58] M. ROGER, J. GOUBAULT-LARRECQ. *Procédé et Dispositif de Résolution de Modèles, Utilisation pour la Détection des Attaques contre les Systèmes Informatiques.* Dépôt français du 13 sep. 1999, correspondant Dyade, demandeurs : 1. INRIA 2. Bull S.A. Numéro de publication : 2 798 490. Numéro d'enregistrement

national : 99 11716. Classification : G 06 F 19/00. Date de mise à la disposition du public de la demande : 16 mars 2001, bulletin 01/11., 1999.

[59] V. SHMATIKOV, J. C. MITCHELL. *Analysis of Abuse-Free Contract Signing.* in « Financial Cryptography », pages 174-191, 2000.

[60] V. SHMATIKOV, J. C. MITCHELL. *Analysis of a fair exchange protocol.* in « Proceedings of the 1999 FLoC Workshop on Formal Methods and Security Protocols », Trento, Italy, 1999.

[61] I. STARK. *Names, Equations, Relations : Practical Ways to Reason about new.* in « Fundamenta Informaticae », numéro 4, volume 33, April, 1998, pages 369-396, http://www.dcs.ed.ac.uk/~stark/publications/namerp-fi.html.

[62] M. STEINER, G. TSUDIK, M. WAIDNER. *Key Agreement in Dynamic Peer Groups.* in « IEEE Transactions on Parallel and Distributed Systems », numéro 8, volume 11, 2000, pages 769-780.

[63] E. SUMII, B. C. PIERCE. *Logical Relations for Encryption.* in « Computer Security Foundations Workshop », 2001, To appear in *Journal of Computer Security*.