Analyse Automatique de Protocoles de Sécurité avec CASRUL*

M. Bouallagui, Y. Chevalier, M. Rusinowitch, M. Turuani, L. Vigneron
LORIA, Nancy, France.

30 avril 2002

Les protocoles cryptographiques comme SET, TLS, Kerberos sont conçus pour sécuriser les transactions électroniques. L'expérience a montré que la conception de ces protocoles est souvent erronée, même en admettant que la cryptographie est parfaite, c'est-à-dire qu'un message codé ne peut être décrypté sans la clé. Un adversaire peut intercepter, analyser et modifier les messages échangés avec peu de moyens de calculs et causer par exemple des dégats économiques importants. Même dans ce modèle abstrait, dit de Dolev-Yao, l'analyse des protocoles cryptographiques est complexe car l'ensemble de configurations à analyser est immense voire **infini**: il faut prendre en compte un nombre quelconque de sessions, une taille quelconque des messages, l'entrelacement des sessions.

Nous avons développé des techniques originales de traduction et de résolution de contraintes pour automatiser l'analyse des protocoles cryptographiques dans le modèle de Dolev Yao à partir de leurs spécifications sous forme de règles d'envoi de messages. Le système CASRUL [5, 1] que nous développons compile les spécifications avant de les soumettre à des procédures de décisions.

Compilateur: Le compilateur de protocoles développé permet de donner une sémantique opérationnelle à une description déclarative d'un protocole. Cette sémantique est donnée sous la forme de règles de réécriture, format générique qui permet l'utilisation de ce compilateur par différents outils d'étude de protocoles. Cette généricité est utilisée, par exemple, dans le cadre du projet européen AVISS[1]. Il est projeté, dans le futur, d'élargir le nombre d'outils pouvant s'interfacer avec le compilateur.

*soutenu par ACI Cryptologie, projet VERNAM

1. www.informatik.uni-freiburg.de/ softech/research/projects/aviss

A la différence d'autres compilateurs de ce type, les connaissances des participants à un protocole sont gérées lors de la compilation. Cela permet de détecter des erreurs sémantiques, par exemple que des participants n'ont pas les connaissances pour construire les messages qu'ils doivent émettre.

Vérification par résolution de contraintes. Le problème de détection d'une attaque dans un environnement hostile se ramène à la résolution de contraintes sur un espace de messages. L'intrus doit être capable de répondre aux messages de manière conforme au protocole, donc de composer des messages acceptables par les participants honnêtes. Il doit être capable à partir de ses connaissances initiales et des réponses suscitées de dériver une information supposée secrète. Pour construire ses messages l'intrus peut décomposer les messages complexes, décrypter les textes lorsqu'il possède la clé inverse, encrypter lorsqu'il possède la clé conforme. La problème de détection d'une attaque est NP-complet [6]. L'implantation des règles de résolution de contraintes dans CASRUL est cependant très efficace en pratique [1].

Cas d'un nombre non-borné de sessions. Les méthodes traditionnelles de model-checking sont limitées à la recherche d'erreurs sur un nombre fini de sessions du protocole. Nous avons développé, dans le cadre de l'outil CASRUL, une extension de la méthode de résolution de contrainte permettant de considérer un nombre de sessions arbitraire. Par abstraction sur les principaux pouvant conduire un nombre non-borné de sessions en parallèle, il est possible de se ramener dans ce cas, à un espace fini. Cette méthode a en outre l'avantage d'accélérer la recherche d'attaques sur les protocoles classiques décrits dans [3]. Elle est détaillée dans [2].

Expérimentations Le système actuel permet de compiler 46 des 51 protocoles de la bibliothèque de référence de Clark et Jacob [3]. D'autre part 35 des 51 protocoles admettent des failles et notre outil peut en détecter 32. En particulier l'une de ces failles (concernant le protocole de Denning Sacco) n'était pas connue auparavant [2]. CASRUL est accessible et exécutable en ligne à l'adresse www.loria.fr/equipes/protheo/SOFTWARES/CASRUL/.

Par rapport à d'autres outils de vérification tels CASPER [4] fondé sur le model-checking la couverture des failles est supérieure. Ainsi la plupart des erreurs dues au typage dans [3] sont découvertes automatiquement par CASRUL.

Références

[1] Y. Chevalier and L. Vigneron. A Tool for Lazy Verification of Security Protocols. In *Proc. of ASE'01*. IEEE CS Press, 2001.

[2] Y. Chevalier and L. Vigneron. Automated Unbounded Verification of Security Protocols In *Int. Conference on Automated Verification '02* to appear in LNCS, Copenhagen, Denmark, July 27-31, 2002

[3] J. Clark and J. Jacob. A Survey of Authentication Protocol Literature: Version 1.0, 17. Nov. 1997. www.cs.york.ac.uk/ jac/papers/drareview.ps.gz.

[4] B. Donovan, P. Norris, and G. Lowe, Analyzing a library of protocols using Casper and FDR. In *Proc. of FMSP'99*.

[5] F. Jacquemard, M. Rusinowitch, and L. Vigneron. Compiling and Verifying Security Protocols. In *Proc. of LPAR'00*, LNCS 1955, pp. 131–160. Springer, 2000. In M. Parigot and A. Voronkov, editors, *Proceedings of LPAR 2000*, LNCS 1955, pages 131–160. Springer, 2000.

[6] M. Rusinowitch and M. Turuani. Protocol insecurity with finite number of sessions is NP-complete. In *Proceedings of 14th IEEE Computer Security Foundations Workshop*, Cape Breton, Nova Scotia, Canada, June 2001.