

Proposition de stage de Master

Lieu :

Laboratoire Spécification et Vérification
École Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan CEDEX

Titre : Vérification d'APIs de sécurité

Description du stage :

Durant les dernières années de nombreuses techniques dans le domaine des méthodes formelles ont été développées pour analyser et prouver la correction de protocoles cryptographiques. Une API ¹ de sécurité peut être vu comme un ensemble de protocoles et de normes informatiques utilisés pour échanger des données entre les applications. De nombreuses applications utilisent des APIs pour sécuriser les transactions. Un module particulièrement critique est celui utilisé par exemple pour la sécurisation des transactions sur les réseaux financiers (réseaux de guichets automatiques, systèmes de virements interbancaires, échanges d'actions, ...). On peut citer également le standard PKCS#11 utilisé à l'heure actuelle dans de nombreuses applications. Les APIs de sécurité devraient bientôt être embarquées dans les voitures, pour sécuriser l'authentification des réseaux voiture-à-voiture.

Dans ces applications particulièrement critiques, de nombreuses attaques ont été découvertes. Citons par exemple des attaques découvertes par Mike BOND sur deux APIs utilisées à l'heure actuelle par la majorité des réseaux financiers [1]. Le besoin de méthodes formelles pour effectuer la vérification de ces APIs se fait sentir [2]. D'une part en raison des applications critiques utilisant ces APIs, et aussi en raison du déploiement à grande échelle de ces protocoles et de la difficulté à les corriger lorsqu'ils ont été déployés.

Récemment, des efforts ont été fait pour vérifier ces APIs à l'aide de méthodes formelles (e.g. [3]). Pour mener à bien cette étude, il est important de développer des techniques spécifiques. Les procédures et outils existants et permettant de vérifier les protocoles cryptographiques classiques ne sont pas adaptés [5]. Une API est généralement composé de nombreux protocoles. Ces derniers sont souvent des requêtes assez simples ne nécessitant pas plusieurs échanges de messages. En revanche, il est important de tenir compte de l'état global du système.

Au cours de ce stage, on pourra mener l'étude d'une ou plusieurs APIs. On pourra s'intéresser au fragment SSL/TLS du standard PKCS#11 [6, §12.31] qui n'a pas fait l'objet d'étude jusqu'à présent ou à l'API de sécurité MiFare embarquée dans les cartes à puces [4]. L'objectif du stage sera de proposer une modélisation de l'API ou du fragment de l'API considéré et de développer une procédure de décision (ou semi-décision) pour vérifier (ou attaquer) cette API. On pourra s'intéresser à différents types de propriété de sécurité en fonction de la nature

¹Application Programming Interface

de l'API choisie. La vérification devra prendre en compte autant que possible les propriétés algébriques des primitives cryptographiques impliqués dans cette APIs.

Personne(s) encadrant le stage :

S. Delaune & G. Steel

Tél : 01 47 40 75 63 & 01 47 40 75 80

Email : {delaune|steel}@lsv.ens-cachan.fr

Références

- [1] M. Bond. Attacks on cryptoprocessor transaction sets. In *Proc. of Cryptographic Hardware and Embedded Systems (CHES'01)*, volume 2162 of *LNCS*, pages 220–234. Springer, 2001.
- [2] M. Bond and J. Clulow. Extending security protocol analysis : New challenges. In *Proc. of Automated Reasoning and Security Protocols Analysis (ARSPA'04)*, Cork (Ireland), 2004.
- [3] S. Delaune, S. Kremer, and G. Steel. Formal analysis of PKCS#11. In *Proceedings of the 21st IEEE Computer Security Foundations Symposium (CSF'08)*, pages 331–344, Pittsburgh, PA, USA, June 2008. IEEE Computer Society Press.
- [4] F. D. Garcia, G. de Koning Gans, R. Muijers, P. van Rossum, R. Verdult, R. W. Schreur, and B. Jacobs. Dismantling mifare classic. In S. Jajodia and J. López, editors, *ESORICS*, volume 5283 of *Lecture Notes in Computer Science*, pages 97–114. Springer, 2008.
- [5] J. Herzog. Applying protocol analysis to security device interfaces. *IEEE Security & Privacy Magazine*, 4(4) :84–87, July-Aug 2006.
- [6] RSA Security Inc., v2.20. *PKCS #11 : Cryptographic Token Interface Standard.*, June 2004.