Stage de L3: Recherche automatique d'attaques sur les protocoles de sécurité

Hubert Comon-Lundh
LSV, École Normale Supérieure de Cachan
comon@lsv.ens-cachan.fr

January 13, 2014

Résumé

Le sujet peut prendre trois directions possibles de développement (voir la version détaillée pour la définition des termes utilisés):

1. Axiomatisation de propriétés de primitives cryptographiques et preuves de validité calculatoire de ces axiomes.
2. Recherche de stratégies de preuve automatique
3. Recherche d'attaques sur des exemples concrets

1 Contexte de travail

Contexte général. Le cadre général est la sécurité des protocoles cryptographiques. Ces protocoles sont des programmes distribués s'appuyant sur des primitives cryptographiques comme le chiffrement, les fonctions de hachage, les signatures etc. Ils sont utilisés dans de nombreuses applications: transactions sécurisées sur internet, téléphonie mobile, cartes à puce, cartes RFID ...

État de l'art. Depuis une vingtaine d'années, les méthodes formelles sont utilisées pour la vérification de protocoles cryptographiques. Ces méthodes formelles s'appuient nécéssairement sur des modèles formels des protocoles, typiquement utilisant des algèbres de processus.

Dans ces modèles formels, les primitives de sécurité sont idéalisées: les messages sont représentés par des termes et toutes les opérations possibles sont spécifiées, en général au moyen de règles de récriture sur ces termes. Par exemple on écrira que, pour une primitive de chiffrement symétrique, $\mathsf{dec}(k, \mathsf{enc}(x, k)) = x$.

Malheureusement, ces modèles (dits "de Dolev Yao") ne sont que des idéalisations des primitives. Il est arrivé que des protocoles soient prouvés corrects, puis qu'on trouve des attaques (dans un modèle plus réaliste). Il faut donc valider le modèle. C'est ce qu'on appelle *computational soundness* dans ce contexte.

Depuis une dizaine d'année et le travail fondateur de M. Abadi et Ph. Rogaway [1], plusieurs preuves de computational soundness ont été proposées dans divers cadres, mais elles supposent toutes des hypothèses que l'on peut juger irréalistes. De plus chacune d'elles ne

s'applique qu'à un ensemble de primitives fixé. Enfin, ces preuves sont en général extrèmement ardues et longues.

Dans un article plus récent [2], nous proposons de renverser la manière de voir le problème, de façon à réduire au minimum les preuves de computational soundness et d'en éviter tous les écueils sus-mentionnés. L'idée est de considérer un modèle d'attaquant dans lequel tout est permis à celui-ci, dès lors que ce qu'il fait est cohérent avec un certain nombre d'axiomes. Autrement dit, on spécifie ce qu'un attaquant ne peut pas faire au lieu de spécifier ce qu'il peut faire, comme dans tous les travaux antérieurs. L'existence d'une attaque revient ainsi à une preuve de cohérence entre les actions de l'attaquant, la négation de la propriété de sécurité et les axiomes spécifiant ce qui lui est interdit.

Contexte au LSV. Cette idée et l'automatisation de la recherche d'attaques suivant le principe ci-dessus est en cours de développement dans la thèse de Guillaume Scerri. (Voir par exemple [3] pour des résultats préliminaires).

2 Objectifs du stage

Le travail de G. Scerri s'est concentré sur le seul cas des primitives de chiffrement, en s'appuyant sur une axiomatisation proposée dans [2]. Il s'est de plus focalisé sur des techniques de recherche de preuve en temps polynômial.

Trois directions de travail **au choix** (pas toutes à la fois) peuvent être poursuivies dans le cadre d'un stage de L3:

1. **Nouvelles primitives** Il s'agit d'étendre le travail déjà réalisé à d'autres primitives cryptographiques, par exemple les signatures. Le travail d'un stagiaire de L3 dans ce contexte consistera à:

    1. Comprendre les définitions de sécurité des signatures
    2. Formaliser ces propriétés en logique du premier ordre
    3. Prouver la "computational soundness" des propriétés
    4. Le cas échéant (si le temps le permet) étudier l'impact sur les procédures de recherche automatique d'attaque

2. **Recherche de stratégies** Nous venons de mettre au point des axiomes pour les propriétés d'indistinguabilité calculatoire. Le travail d'un stagiaire de l'ENS consisterait à étudier l'automatisation des preuves de cohérence en présence de ces axiomes. Il ne s'agirait pas de partir de zéro, puisqu'une telle automatisation a été réalisée par G. Scerri dans le cas des propriétés d'accessibilité.

3. **Recherche d'attaque sur les protocoles existants** Il s'agit ici d'un sujet de nature différente, plus expérimentale. Le stagiaire de L3 devra considérer des protocoles classiques (par exemple Kerberos), les formaliser dans le prototype démonstrateur de G. Scerri et chercher des attaques. Le succès est assuré si on trouve une attaque. Mais il est probable que ceci demande de retoucher le code du prototype (en OCaml) car il se peut que la stratégie doive être adaptée pour être suffisamment efficace sur l'étude de cas choisie.

3 Compétences requises

Un élève de fin de première année d'ENS a en principe les compétences requises pour les 3 sujets.

Mais, selon ses goûts, les directions de travail seront plus ou moins bien adapées:

1. L'axiomatisation de nouvelles primitives demandera des preuves de correction, qui sont essentiellement des réductions de machines de Turing probabilistes en temps polynômial.
2. La recherche de stratégies demande plutôt de bien comprendre les mécanisme de preuve automatique (ici, résolution, réécriture).
3. La recheche d'attaques sur des études de cas demandera plutôt d'avoir du goût pour la programmation (en Ocaml).

References

[1] Martín Abadi and Phillip Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *J. Cryptology*, 15(2):103–127, 2002.

[2] Gergei Bana and Hubert Comon-Lundh. Towards unconditional soundness: Computationally complete symbolic attacker. In Pierpaolo Degano and Joshua D. Guttman, editors, *Proceedings of the 1st International Conference on Principles of Security and Trust (POST'12)*, volume 7215 of *Lecture Notes in Computer Science*, pages 189–208. Springer, March 2012.

[3] Hubert Comon-Lundh, Véronique Cortier, and Guillaume Scerri. Tractable inference systems: an extension with a deducibility predicate. In *Proceedings of the 24th International Conference on Automated Deduction (CADE'13)*, volume 7898 of *Lecture Notes in Artificial Intelligence*, Lake Placid, New York, USA, 2013. Springer.