Contrôle d'accès dans les bases de données : conception et vérification

Résumé : Vérifier qu'une spécification de contrôle d'accès sur des bases de données ne fuite des données permettant de déduire un secret.

Encadrants : Pierre Bourhis pierre.bourhis@univ-lille1.fr, Sophie Tison sophie.tison@univ-lille1.fr

Localisation : Equipe Links Batiment B INRIA

Pré-Requis : Bases en théorie des automates et Bases des données

Contexte

Le problème de la sécurité dans les bases de données est un enjeu primordial. Une approche classique pour protéger les données est de définir des vues pour masquer les données sensibles tout en autorisant d'en exhiber une autre partie. Ces vues appelées vues de sécurité sont définies par des requêtes sur le schéma initial. Par exemple pour anonymiser des données, les colonnes contenant les noms et prénom des personnes ont été cachées. Malheureusement, ces vues de sécurités ne sont pas suffisantes pour assurer que les données sensibles soient complètement cachées d'un attaquant malveillant. En effet, en utilisant les définitions des vues ainsi que les propriétés sur les bases de données liant les données entre elles, comme par exemple les clés étrangères et des données visibles, un attaquant malveillant peut déduire des données sensibles. Par exemple, dans un exemple simulant Facebook, les photos des différents utilisateurs sont stockées dans une relation Photo et les utilisateurs qui sont amis sont stockés dans la relation Amis. Dans cet exemple les amis et les photos des autres utilisateurs sont les données sensibles. La relation Amis est cachée et seules mes photos et celles de mes amis et des amis de mes amis me sont visibles. En utilisant les photos qui me sont montrées, je peux ainsi déduire qui sont les amis de mes amis ce qui est une fuite de données sensibles.

Problématique

Dans un travail récent [2], nous avons démontré qu'il est possible de vérifier que pour un ensemble de vues et un ensemble de contraintes satisfaites par la base données, il n'existe pas de base de données à partir de laquelle, un attaquant peut déduire des données sensibles. Si dans le cadre général, le problème a une grande complexité, il existe un cadre réaliste dans lequel la complexité est raisonnable (PSPACE). Nous avons de plus démontré que ce problème peut se réduire à une généralisation d'un problème classique pour les bases de données : le problème d'inclusion de requêtes conjonctives sous des contraintes d'inclusions [1].

Travail à faire

Le but de ce projet est d'implémenter l'algorithme pour chercher la fuite de données sensibles. Pour cela, nous chercherons un algorithme pour résoudre la généralisation du problème d'inclusion de requêtes lorsque les bases de données sous-jacentes satisfont des contraintes. A notre connaissance, il n'existe pas d'implémentation efficace de ce problème. Une approche semble envisageable en

réduisant ce problème à des problèmes liés aux automates (test du vide, inclusion de deux automates) et en utilisant les récentes optimisations des algorithmes pour ceux-ci [3].

Références

[1] Serge Abiteboul, Rick Hull, Victor Vianu: Foundations of Databases

[2] Michael Benedikt, Pierre Bourhis, Balder ten Cate, Gabriele Puppis:
Querying Visible and Invisible Tables in the Presence of Integrity Constraints.
http://arxiv.org/abs/1509.01683

[3] Filippo Bonchi, Damien Pous: Hacking Nondeterminism with Induction and Coinduction, Communications of the ACM, 2015