Contrôle d'accès basé sur la provenance

François Lesueur
francois.lesueur@insa-lyon.fr
Université de Lyon, CNRS
INSA-Lyon, LIRIS, UMR5205
F-69621, France

Romuald Thion
romuald.thion@univ-lyon1.fr
Université de Lyon, CNRS
Université Lyon 1, LIRIS, UMR5205
F-69622, France

Résumé

Cet article propose un mécanisme de contrôle d'accès pour les BD relationnelles basé sur la provenance. La source d'une donnée définit des autorisations sur ses tuples ; les autorisations sont ensuite propagées quand les tuples sont combinés par des requêtes. Chaque tuple possède ainsi une trace grâce à laquelle les autorisations résultantes peuvent être déterminées en accord avec l'ensemble des sources ayant contribué. Nous décrivons et formalisons ce modèle de sécurité et nous proposons une première conception sous certaines hypothèses de confiance. Des pistes permettant de relâcher ces hypothèses sont également proposées.
Mots-clefs : Sécurité des BD, Contrôle d'accès, Provenance des données

1 Introduction

Dans la société de l'information où la richesse devient immatérielle, il est crucial pour les producteurs de données de maintenir un certain contrôle sur la diffusion des données qu'ils produisent ou possèdent. Cependant, lorsque ces données sont traitées successivement conjointement avec d'autres, il devient difficile de déterminer les droits d'accès répondant à l'ensemble des exigences des sources initiales. Cet article de positionnement propose un travail préliminaire permettant aux producteurs de spécifier les règles de diffusion de leurs données.

Les droits d'accès sont représentés par des annotations associées à chaque tuple. À l'issue des opérations relationnelles successives réalisées entre des bases appartenant à différentes organisations, c'est-à-dire quand elles se transmettent des résultats de requêtes, une base ayant agrégé des tuples calcule les droits nécessaires à son accès et propage ainsi les autorisations définies par les producteurs.

Le mécanisme de gestion de données relationnelles annotées sur lequel nous définissons le modèle de sécurité est le cadre de *provenance* de Green *et al.* [7]. Dans cet article, nous utilisons les informations de provenance pour assurer la sécurité des données relationnelles ; nous n'étudions pas ici *la sécurité des informations de provenance*. La provenance est essentiellement utilisée comme un vecteur de propagation des politiques de sécurité.

Un cas d'usage paradigmatique du modèle proposé est celui de l'échange de données personnelles. Le modèle permet à une personne physique d'associer des groupes de sécurité

aux tuples le concernant (par exemple, données d'une table *Personne*, enregistrements de ses activités, ...). Ces groupes d'entités autorisées sont propagés par les requêtes et un SGBD peut ainsi déterminer si un accédant est légitime vis-à-vis des règles de diffusion fixées par les propriétaires.

2 État de l'art et positionnement

Les modèles à base de rôles sont couramment utilisés en bases de données [11]. Le contrôle repose uniquement sur l'identifiant de la ressource (le nom du fichier par exemple) et non pas sur le flux des informations ayant contribué à la constituer.

Le modèle multi-niveaux de Bell et LaPadula permet de contrôler les flux d'informations dans un système afin d'en garantir la confidentialité [4]. Les utilisateurs ont un niveau d'accréditation et chaque ressource est labélisée par un niveau de confidentialité (public, confidentiel, secret, très secret) : les utilisateurs ne peuvent lire que les données de niveau inférieur ou égal à leur propre niveau et ne peuvent écrire que dans des ressources de niveau supérieur. Les modèles multi-niveaux, extrêmement contraignants, sont principalement utilisés dans des environnements fortement contrôlés et centralisés (par exemple, militaires, bancaires).

Plus récemment, des modèles de contrôles de flux réseau [10, 12, 9] ont proposé d'assouplir la définition des flux autorisés. Les règles sont plus génériques mais le suivi des flux ne prend pas en compte les propriétés de l'algèbre relationnelle. En effet, le caractère déclaratif du calcul relationnel implique qu'un même tuple peut être obtenu de plusieurs façons différentes et permet d'analyser finement les données contribuant à un tuple.

Dans cet article, nous proposons de mettre en œuvre un contrôle d'accès pour les BD relationnelles prenant en compte les flux d'informations. Dans ce cadre, le contrôle prend en compte les droits définis originellement sur les données initiales : l'objectif est de permettre aux sources de données de spécifier à qui leurs données peuvent être diffusées. Lors d'une opération relationnelle, les droits d'accès aux tuples résultants sont calculés à partir des permissions sur les tuples sources. Comparé à l'exemple de contrôle multi-niveaux à l'aide de provenance présenté dans [7], le modèle présenté dans cet article s'appuie sur la notion de groupe d'identité, ce qui permet une gestion plus souple des autorisations et ne nécessite pas de consensus sur la sémantique des annotations de sécurité.

3 Modèle de sécurité

Dans cette section, nous présentons le modèle de sécurité qui est formalisé dans la section 4. Une conception du système sous hypothèse de confiance est décrite puis discutée en section 5.

Le système étudié est composé d'un ensemble de bases de données, placées sous des autorités différentes. Cet ensemble est d'une part composé des *sources de données* : ce sont des bases de données qui fournissent des données mais n'utilisent pas de données issues d'autres bases du système. Cet ensemble est d'autre part composé des *collecteurs de données* : ce sont des bases dont les données sont issues d'autres bases du système. Certaines bases peuvent à la fois apporter des données au système et utiliser des données issues d'autres bases et donc être à la fois *source* et *collecteur*.

Les utilisateurs du système demandent l'exécution de requêtes sur les bases de données : ce sont les *consommateurs* finaux des données. Les sources de données attribuent des labels de sécurité à leurs tuples, chaque label pouvant être attribué à un ensemble de tuples nécessitant les mêmes règles de sécurité. Ces sources maintiennent ensuite des accréditations accordées aux différents utilisateurs pour accéder aux tuples annotés par ces labels. L'objectif central de cet article est ainsi :

Assurer qu'un consommateur ne reçoit des informations concernant une source que si cette dernière l'autoriserait directement.

Pour le cas d'usage sur des données personnelles présenté en fin de la section 1, les sources sont les personnes physiques identifiables par les données, les collecteurs sont des personnes morales ou physiques qui recueillent des données personnelles, éventuellement en les échangeant entre elles. Les consommateurs sont les personnes qui font des demandes d'accès, par exemple à un annuaire de clients.

Chaque source est identifiée par un élément $s \in S$. Chaque source s définit un ensemble G_s d'identifiants de groupe d'utilisateurs. Chaque tuple est annoté par un label de la forme $s.g$, $g \in G_s$ supposé unique dans le système. On appelle G l'ensemble de tous les labels.

Quand un utilisateur u tente d'accéder à un tuple t annoté $s.g$, noté $(t : s.g)$, l'accès doit être autorisé *si et seulement si* s l'autorise, c'est-à-dire si u appartient au groupe g. Il s'agit du cas élémentaire où l'annotation est atomique. Les données pouvant être combinées par des requêtes, il faut donc, connaissant les annotations des tuples contribuant à un tuple du résultat d'une requête, déterminer quelle est l'annotation du tuple résultat (nous nous limitons ici à l'algèbre SPJRU) :

Renommage (ρ) chaque tuple conserve son annotation. En effet, le tuple ne changeant pas, ses permissions restent les mêmes.

Sélection (σ_P) chaque tuple conserve son annotation originale. En effet, le tuple est simplement filtré par le prédicat P.

Jointure naturelle ($\bowtie$) chaque tuple joint combine les tuples sources. L'accès à un tuple joint nécessite donc le droit d'accéder *à l'ensemble des tuples sources.*

Projection (π_V) ou Union ($\cup$) chaque tuple obtenu peut dériver de plusieurs tuples différents. L'accès au tuple résultant doit donc être autorisé pour tout utilisateur ayant accès *à l'un des tuples initiaux.*

4 Expression formelle du modèle de sécurité

Green *et al.* [7] étendent le modèle relationnel en considérant des K-relations où les tuples sont annotés par une valeur d'un ensemble K muni de deux lois $\oplus$ et $\otimes$ avec deux éléments distingués 0 et 1. Informellement, $\oplus$ permet de combiner les annotations pour l'union et la projection, $\otimes$ combine dans le cas de la jointure, 0 et 1 jouent le rôle de vrai et faux. Nous utilisons ce cadre pour formaliser la sémantique de sécurité décrite en section précédente.

Une K-relation r de schéma U est formellement une fonction $r : (U \to \mathbb{D}) \to K$ dont le support $supp(r) = \{t | r(t) \neq 0\}$ est fini. Green *et al.* étendent l'algèbre relationnelle SPJRU aux K-relations et montrent que la structure $\langle K, \oplus, \otimes, 0, 1\rangle$ doit être un

semi-anneau commutatif, c'est-à-dire telle que $\langle K, \oplus, 0\rangle$ et $\langle K, \otimes, 1\rangle$ soient des *monoïdes commutatifs* (loi de composition *associative* et *commutative* avec un *élément neutre*) avec 0 l'élément *absorbant* pour $\otimes$ et $\otimes$ qui *distribue* sur $\oplus$: $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$.

Théorème 1 (issu de [7])**.** *Les identités algébriques (sauf l'idempotence de l'union et de la jointure) de SPJRU sont respectées par l'algèbre relationnelle SPJRU étendue aux K-relations* ssi $\langle K, \oplus, \otimes, 0, 1\rangle$ *est un semi-anneau commutatif.*

D'après le théorème 1, nous devons munir l'ensemble G des annotations d'une structure de semi-anneau commutatif. Informellement, il s'agit de proposer une représentation et une gestion des labels de sécurité compatible avec l'algèbre SPJRU.

Nous définissons la structure $\langle \mathcal{P}(\mathcal{P}(G)), \cup, \Cup, \emptyset, \{\emptyset\}\rangle$ où $\mathcal{P}(G)$ désigne l'ensemble des sous-ensembles de G et où $\Cup$ est l'union paire à paire : $X \Cup Y = \{x \cup y \mid x \in X \wedge y \in Y\}$. Les éléments de $\mathcal{P}(\mathcal{P}(G))$ sont des ensembles d'ensembles d'annotations, par exemple $\{\{s_0.g_0, s_0.g_1\}, \{s_2.g_0\}, \{s_0.g_1, s_2.g_1\}\}$. Les annotations initiales définies par les sources sont de la forme $\{\{s.g\}\}$. Cette structure a été proposée dans [5] pour capturer la *why provenance*, c'est-à-dire une collection de collection de *témoins* qui indique quels tuples sources ont contribué à la formation d'un tuple.

Proposition 1 (issue de [5])**.** $\langle \mathcal{P}(\mathcal{P}(G)), \cup, \Cup, \emptyset, \{\emptyset\}\rangle$ *est un semi-anneau commutatif.*

Pour prendre la décision d'accès, un collecteur doit comparer un ensemble d'accréditations $Cr \subseteq G$ fournies par le consommateur (typiquement, des certificats) à l'annotation X de chaque tuple demandé. La fonction $eval : \mathcal{P}(G) \times \mathcal{P}(\mathcal{P}(G)) \to \mathbb{B}$, définie par $eval(Cr, X) = \exists C \in X.C \subseteq Cr$, permet de déterminer si l'accès au tuple annoté par X est autorisé.

On note $eval_{Cr} : \mathcal{P}(\mathcal{P}(U)) \to \mathbb{B}$ le curryfié de *eval* dont le premier argument est fixé. Cette fonction respecte les propriétés du modèle de sécurité décrites en section 3. Informellement, l'utilisation alternative (union et projection), représentée par $\cup$, correspond à une disjonction logique des permissions et l'utilisation conjointe (jointure naturelle), représentée par $\Cup$, à une conjonction de ces permissions. La proposition 2 formalise cette intuition et garantit ainsi l'atteinte de l'objectif fixé : un consommateur ne reçoit des informations concernant des sources que si ces dernières l'autoriseraient directement.

Proposition 2. *$eval_{Cr}$ est une transformation qui préserve la structure des annotations, c'est-à-dire un homomorphisme formalisé par les quatre équations suivantes :*

$$eval_{Cr}(X \cup Y) = eval_{Cr}(X) \vee eval_{Cr}(Y) \qquad eval_{Cr}(\emptyset) = \bot$$
$$eval_{Cr}(X \Cup Y) = eval_{Cr}(X) \wedge eval_{Cr}(Y) \qquad eval_{Cr}(\{\emptyset\}) = \top$$

5 Mise en œuvre du modèle

5.1 Réalisation sous hypothèse de confiance

Dans cette première conception, nous considérons que les collecteurs se font confiance ou s'exécutent sur du matériel de confiance certifié par une autorité tierce, comme proposé par exemple dans TrustedDB [3] ou dans les PDS [2]. En revanche, ils ne font pas confiance aux utilisateurs finaux et les collecteurs recevant des requêtes d'utilisateurs doivent donc pouvoir calculer la légitimité de leurs accès.

L'hypothèse de confiance entre les collecteurs implique que les collecteurs appliquent correctement les règles de sécurité définies par les sources. La fonction *eval* permet alors aux collecteurs de savoir décider de l'autorisation d'accès à un tuple. Tous les tuples sont stockés en clair et chaque collecteur peut ainsi réaliser les opérations relationnelles de manière usuelle. Les annotations sont calculées fidèlement tout au long de ces opérations et l'application des règles définies par les sources est appliquée par les collecteurs finaux.

Lors du transfert à l'utilisateur final, le collecteur applique le contrôle d'accès défini par les annotations des tuples. u envoie la requête Q au collecteur C qui, sur sa base annotée $\mathcal{I}$, calcule le résultat annoté $Q(\mathcal{I})$. u demande alors les accréditations aux sources (sous forme de certificats), les présente au collecteur et ce dernier filtre les résultats selon ces accréditations. Pour Cr le jeu d'accréditations présenté par u, C envoie à u le résultat $\{(t : X) \mid (t : X) \in Q(\mathcal{I}) \wedge eval_{Cr}(X) = \top\}$.

5.2 Vers la relaxation des hypothèses de confiance

La première conception permet un contrôle d'accès distribué entre des bases collaborant de manière fiable. Nous souhaitons dépasser ce cadre et diminuer fortement ces hypothèses de confiance, en limitant la confiance accordée aux collecteurs. Pour ce faire, nous souhaitons mettre à profit des mécanismes cryptographiques : la source ne transmettra que *des données chiffrées* et permettra leur utilisation par chaque acteur de manière limitée. Les annotations peuvent ainsi être vues comme des « recettes cryptographiques » représentant les opérations de chiffrement réalisées à chaque transmission d'un résultat de requête.

Dans ce cas, chaque groupe $s.g$ est associé à un couple de clefs asymétriques. Pour chaque tuple $(t : X) \in Q(\mathcal{I})$, avec par exemple $X = \{x_1 = \{s_0.g_0, s_0.g_1\}, x_2 = \{s_1.g_0\}, x_3 = \{s_0.g_1, s_1.g_1\}\}$, le clair de t n'est pas stocké mais un exemplaire est stocké chiffré par les clés publiques correspondant à chaque partie x_i de l'annotation. Pour x_1, t est chiffré avec successivement les clés correspondant à $s_0.g_0$ puis $s_0.g_1$. Pour obtenir les clés $s_0.g_0$ et $s_0.g_1$, un consommateur doit les demander à la source s_0.

Les opérations relationnelles devront être exécutées sur des données chiffrées, en se basant par exemple sur les mécanismes issus des BD privées [1, 8]. Le problème que nous proposons est toutefois plus complexe car le système devra permettre de réaliser des opérations relationnelles sur des données chiffrées successivement par de multiples clés, succession décrite dans la formule de provenance annotant le tuple.

6 Conclusion

Dans cet article, nous avons présenté un modèle de sécurité permettant aux sources de données de définir les droits autorisés sur des données relationnelles. Chaque tuple est annoté, les annotations sont composées lors des opérations relationnelles et un utilisateur final n'obtient le résultat d'une requête que s'il est habilité à lire au moins l'un des ensembles de tuples initiaux composant le tuple demandé.

Ce travail préliminaire présente une solution partielle aux problèmes de transmission de droits d'accès définis par des sources. Une telle technique peut par exemple être envisagée dans le cadre des bases de données hippocratiques, pour lesquelles les techniques de gestion de provenance relationnelle n'ont, à notre connaissance, pas encore été utilisées.

Nous souhaitons dans un premier temps diminuer les hypothèses de confiance nécessaires dans le première réalisation présenté. Dans le système que nous envisageons, décrit dans la section 5, les différentes bases de données composant les tuples ne se font pas confiance et le contrôle d'accès doit donc être contraint. Nous prévoyons d'exploiter pour cela des mécanismes cryptographiques dont il faudra évaluer le surcoût.

Dans un second temps, nous prévoyons d'étendre le langage de requête. Dans cet article, nous nous sommes limités à l'algèbre SPJRU. Des résultats théoriques récents sur les annotations nous permettent d'envisager l'intégration de langages d'interrogation plus riches que SPJRU [6]. Le cas des agrégats est particulièrement intéressant dans le cas de données personnelles par exemple car il permet d'anonymiser les résultats.

Enfin, les politiques de sécurité descriptibles sont ici limitées à une correspondance directe entre l'annotation de sécurité finale et les sources. Nous nous intéresserons à l'extension de ces politiques, ce qui impliquera une évolution du langage des annotations et de la fonction $eval_{Cr}$.

Références

[1] R. Agrawal, A. Evfimievski, and R. Srikant. Information sharing across private databases. *SIGMOD'03 : International Conference on Management of Data*, Jun 2003.

[2] T. Allard, N. Anciaux, L. Bouganim, Y. Guo, L. L. Folgoc, B. Nguyen, P. Pucheral, I. Ray, I. Ray, and S. Yin. Secure personal data servers : a vision paper. *Proceedings of the VLDB Endowment*, 3(1-2) :25–35, 2010.

[3] S. Bajaj and R. Sion. Trusteddb : a trusted hardware based database with privacy and data confidentiality. *Proceedings of the 2011 international conference on Management of data*, pages 205–216, 2011.

[4] D. E. Bell and L. J. LaPadula. Secure computer systems : Mathematical foundations and model. *MITRE CORP BEDFORD MA*, 1(M74-244), 1973.

[5] P. Buneman, J. Cheney, W.-C. Tan, and S. Vansummeren. Curated databases. *PODS'08 : 27th symposium on Principles Of Database Systems*, Jun 2008.

[6] F. Geerts and A. Poggi. On database query languages for k-relations. *Journal of Applied Logic*, 8(2) :173 – 185, 2010. Selected papers from the Logic in Databases Workshop 2008.

[7] T. Green, G. Karvounarakis, and V. Tannen. Provenance semirings. *PODS'07 : 26th symposium on Principles Of Database Systems*, Jun 2007.

[8] C. Hazay and Y. Lindell. Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries. *Journal of cryptology*, 23(3) :422–456, 2010.

[9] J. Liu, M. George, K. Vikram, X. Qi, L. Waye, and A. Myers. Fabric : a platform for secure distributed computation and storage. *SOSP'09 : 22nd Symposium on Operating Systems Principles*, Oct 2009.

[10] A. Myers and B. Liskov. A decentralized model for information flow control. *SOSP'97 : 16th Symposium on Operating Systems Principles*, Dec 1997.

[11] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *Computer*, 29(2) :38–47, 1996.

[12] N. Zeldovich, S. Boyd-Wickizer, and D. Mazières. Securing distributed systems with information flow control. *NSDI'08 : 5th Symposium on Networked Systems Design and Implementation*, Apr 2008.