Sécurisation des entrepôts de données contre les inférences en utilisant les réseaux Bayésiens

Salah TRIKI*, Hanene BEN-ABDALLAH*, Jamel FEKI*, Nouria HARBI**

* Laboratoire Mir@cl
Département d'Informatique,
Faculté des Sciences Economiques et de Gestion de Sfax,
Route de l'Aéroport Km 4 – 3018 Sfax, BP. 1088
{Salah.Triki, Hanene.BenAbdallah, Jamel.Feki}@Fsegs.rnu.tn
**Laboratoire ERIC
Université Lyon 2,
5 avenue P. Mendès France 69676 Bron, Cedex
Nouria.Harbi@univ-lyon2.fr

Résumé. Les entrepôts de données permettent aux analyste-décideurs d'établir des prévisions et de prendre des décisions stratégiques. La sécurisation de ces entrepôts est, par conséquent, importante afin de protéger les informations sensibles. Par ailleurs, cette sécurisation ne doit pas constituer une entrave à l'exploitation efficace et rapide de l'entrepôt, ni être trop souple induisant l'inférence des données interdites (i.e., données personnelles, confidentielles). Dans cet article, nous examinons la sécurisation des entrepôts de données à travers une approche basée sur les réseaux Bayésiens. Elle comporte deux avantages : d'une part, elle ne nécessite pas un traitement supplémentaire après chaque phase d'alimentation de l'entrepôt et, d'autre part, elle n'entraine pas l'altération des données originales.

1 Introduction

Les entrepôts de données (ED) occupent une place importante dans les organisations. Ils permettent aux décideurs d'explorer les données de celles-ci et de prendre des décisions stratégiques. Les données sont agrégées selon des dimensions formant ainsi un cube. L'agrégation permet de ressortir les corrélations entre les données et de définir les tendances. L'utilisation des opérations de forage vers l'avant rend possible l'exploration de ces corrélations et de ces tendances, offrant de l'information qui aide à la prise de décision.

Evidemment, les données de l'ED d'une organisation sont sensibles et ne doivent pas être cédées sans contrôles. Dans ce cadre, plusieurs gouvernements ont promulgué des lois pour la protection des vies privées de leurs citoyens. Parmi ces lois, HIPPA (« Health Insurance Portability and Accountability Act » HHS (1996)) vise à protéger les données médicales des patients américains en obligeant les établissements du secteur des soins de la santé de suivre des règles de sécurités strictes ; de même, GLBA (« Gramm-Leach-Bliley Act » GPO

(1999)) oblige les organismes financiers américains à protéger les données de leurs clients ; Safe Harbor (Export (2008)) permet aux entreprises s'y conformant de transférer et d'utiliser les données concernant les internautes européens ; Sarbanes-Oxley (Soxlaw (2002)) garantit la fiabilité des données financières des entreprises.

Par ailleurs, la sécurisation des entrepôts de données peut être abordée à deux niveaux : (i) niveau conception qui vise à concevoir un entrepôt de données sécurisé ; et (ii) niveau exploitation qui vise à renforcer les droits d'accès/habilitations des utilisateurs, et à interdire tout utilisateur malicieux *d'inférer* des données interdites à partir des données auxquelles il a accès. Pour le premier niveau de sécurité, plusieurs travaux ont proposé des méthodes de conception d'ED sécurisé (*cf,* Priebe et Pernul (2000), Rosenthal et Sciore (2000), Villarroel et al. (2006), Soler et al. (2007)) et des notations pour la modélisation des aspects de la sécurité (*cf.*, Soler et al. (2006), Soler et al. (2008)). Ces propositions permettent de modéliser les droits d'accès/habilitations des utilisateurs de l'ED, qui devraient être renforcés par le serveur OLAP et ce au deuxième niveau.

Quant au deuxième niveau de sécurité, le serveur OLAP est sensé assurer des accès en fonction des habilitations de chaque utilisateur. Le serveur OLAP peut refuser les accès aux données d'une mesure, d'une dimension, et/ou au-delà d'un niveau dans une hiérarchie. A l'instar des SGBD des systèmes d'information, les droits d'accès peuvent être explicitement spécifiés sur les tables/colonnes de tables de l'entrepôt. Cependant, le serveur OLAP tout seul ne peut pas protéger l'accès aux données interdites et ce par *inférence*.

L'objectif de cet article et de montrer la possibilité de protéger un entrepôt de données contre les inférences grâce l'ajout d'un module de contrôle qui complète le travail d'un serveur OLAP comme proposé dans l'architecture présentée dans (Salah *et al*. (2009)),

Le module de contrôle vise à interdire à un utilisateur d'inférer des données protégées à partir des données qui lui sont accessibles. Dans ce travail, nous nous intéressons aux cas des requêtes utilisant les fonctions d'agrégations Max et Min. Nous proposons une approche basée sur les réseaux Bayésiens qui ne nécessite pas un traitement supplémentaire après chaque phase d'alimentation de l'entrepôt de données et qui n'entraine pas l'altération des données originales.

Le reste de l'article est organisé comme suit : dans la section 2, nous présentons un état de l'art des travaux effectués dans le domaine de la sécurisation des entrepôts de données. La section 3 constitue une introduction aux réseaux Bayésiens, à la section 4 nous détaillons notre approche et nous étudions sa complexité. La section 5 présente un exemple illustrant l'utilisation de notre approche. Finalement nous concluons l'article par la section 6.

2 Etat de l'art

Sung et al. (2006) signalent qu'il existe des similitudes entre les bases de données statistiques et les entrepôts de données ; vu leur ancienneté, les premières peuvent aider à trouver des solutions pour la prévention des inférences dans les entrepôts de données. Dans les bases de données statistiques, seules les requêtes utilisant les fonctions d'agrégations sont autorisées. Au niveau des entrepôts de données le besoin de sécurisation s'est fait ressentir depuis longtemps (Shoshani et al 1997, Bhargava 2000, Pernul et al 2000). Plusieurs travaux ont été réalisés couvrant les niveaux métier (Soler et al. 2008), conception (Villarroel et al. 2006) et niveau logique (Soler et al. 2006) d'une approche MDA (« Model Driven Approach »). La sécurisation au niveau exploitation, comporte deux classes d'approches.

La première consiste à interdire les requêtes des utilisateurs malicieux à l'aide de ses anciennes requêtes. Cette façon d'aborder le problème a été faite dans le cadre des bases de données statistiques. Ainsi, Denning et Schlörer (1983) proposent de fixer le nombre minimal de tuples utilisés dans une requête, Dobkin et al (1979). proposent de fixer le nombre de valeurs communes utilisés dans les requêtes, Dobkin et al (1979) proposent une approche permettant d'auditer les requêtes utilisant les fonctions de type moyenne et médiane, tandis que Chin et Ozsoyoglu (1982), proposent une approche permettant d'auditer les requêtes contenants la fonction d'agrégations Max. De même dans le cadre des entrepôts de données, Zhang et al (2004) proposent une approche consistant à compter le nombre de cellules utilisées dans les requêtes précédentes afin de décider est ce que la nouvelle requête peut être répondue, Malvestuto et al (2006) proposent une approche utilisant ILP (« Integer Linear Programming »).

La deuxième classe d'approches consiste à ajouter des perturbations aux données originales. Au niveau des bases de données statistiques, Traub et al (1984) proposent d'effectuer des perturbations aléatoires sur les données, tandis que Beck (1980) proposent de les porter sur les réponses des requêtes et Schlërer (1981) proposent de les faire sur les structures de la base de données. Au niveau des entrepôts de données, Sung et al (2006) proposent d'altérer aléatoirement les données, Agrawal et al (2005) proposent une approche répartie dans laquelle chaque client porte des perturbations aux données et le serveur rétablit les distributions, Hua et al (2005) proposent une approche permettant de cacher les données à partir des quelles les risques d'inférences existent.

Au sujet des travaux utilisant les perturbations, nous avons recensé deux inconvénients. Le premier inconvénient est dû au fait que ces perturbations entrainent un traitement supplémentaire après la phase d'alimentation de l'entrepôt de données. Le second est la perte totale des données originales une fois les perturbations sont appliquées. L'approche que nous proposons ci-après pallie à ces deux insuffisances.

3 Les réseaux Bayésiens

Un réseau Bayésien est un graphe orienté sans circuit (DAG : Direct Acyclic Graph) (Stuart 2000). Chaque nœud possède plusieurs *états* annotés par leur probabilité. Celles-ci sont indiquées dans une table de probabilités conditionnelles (CPT : Conditional Probabilities Table).

La topologie d'un réseau Bayésiens dans un domaine donné, décrit l'interdépendance des variables entre elles. *L'état* observé d'un nœud, appelé *évidence*, peut avoir une répercussion sur les états des autres nœuds. Les arcs entre les nœuds représentent les relations de causes à effets existantes entre ceux-ci. On dit que le nœud X est le père d'un nœud Y, s'ils sont reliés par un arc orienté allant du nœud *X* vers le nœud *Y* ; inversement, Y est l'enfant de X. Un nœud peut avoir plusieurs parents ainsi que plusieurs enfants. L'ensemble des parents d'un nœud X_i est noté $Pa(X_i)$. La taille de la CPT d'un nœud dépend du nombre de ses états, du nombre *n* de ses parents ainsi que du nombre des états de ses parents. La figure 1 indique un réseau Bayésien composé de deux nœuds parents Pa1 et Pa2 et un nœud enfant P où chacun possède 2 états. La taille du CPT des nœuds parents est 2, et celle du nœud enfant est 16.

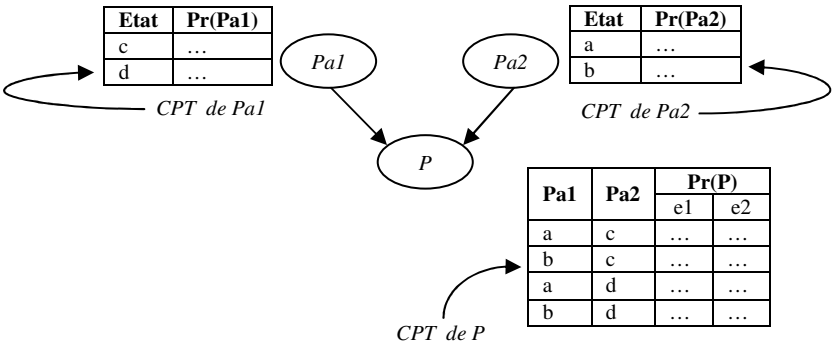

FIG. 1 – La CPT d'un nœud dépend du nombre d'états des nœuds de ses parents.

4 Présentation de l'approche

Dans cette section, nous présentons une approche basée sur les réseaux Bayésiens pour la prévention des inférences à travers des requêtes utilisant les fonctions d'agrégations *Min* et *Max*. Nous détaillons le cas des requêtes utilisant les fonctions de type Max. Cependant, le cas des requêtes utilisant les fonctions d'agrégations de type Min est traité de la même façon.

4.1 Définitions

Définition 1 : $Q = \{S_1 , S_2, \dots , S_n\} : S_i = \{Val_1, Val_2 \dots Val_n\}$
Le résultat d'une requête Q est un ensemble de sous-ensembles de mesures. Chaque ensemble regroupe les mesures correspondantes aux mêmes valeurs des paramètres. Ceux-ci appartiennent aux dimensions utilisées dans la requête Q.

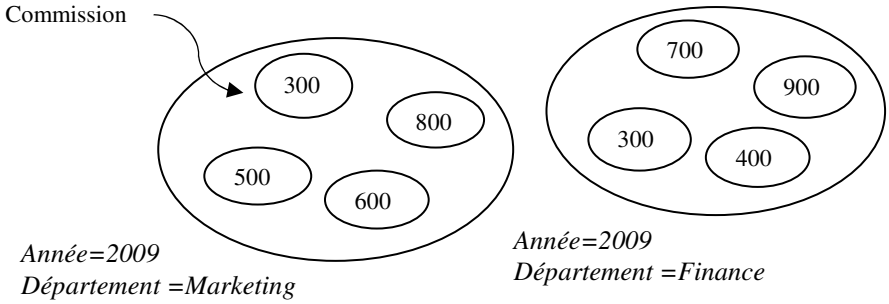

FIG. 2 – Le résultat de la requête les commissions des employés par département et par année.

La figure 2 montre le résultat de la requête *les commissions des employés par département et par année*.

Définition 2 : *Le résultat d'une requête contenant une fonction d'agrégation Max est un ensemble de mesures. Chaque mesure de cet ensemble est le maximum d'un ensemble de mesures. Celui-ci appartient au résultat de la même requête sans la fonction d'agrégation Max.*

$Q = \{ S_1 , S_2, \dots , S_n\} : S_i = \{Val_1, Val_2 \dots Val_n\}$

$Max\ (Q) = \{M_1, M_2, \dots , M_n\} : M_i = Max(S_i)$

La figure 3 montre le résultat de la requête *le maximum des commissions des employés par département et par année*.

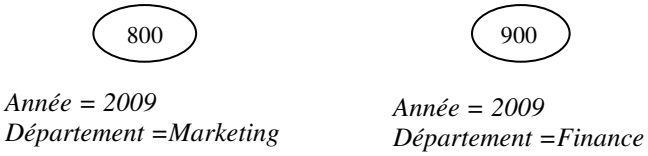

FIG. 3 – Le résultat de la requête le maximum des commissions des employés par département et par année

4.2 Construction du réseau Bayésien

Les définitions 1et 2 nous permettent de créer un réseau Bayésien pour chaque mesure qui existe dans le résultat d'une requête de type Max (Min). Le réseau Bayésien contient un ensemble de nœuds parents et un nœud enfant : Les nœuds parents contiennent les mesures utilisées pour calculer la valeur maximale et leur nœud enfant contient celle-ci. Chaque nœud parent possède deux états : *=Mesure Maximale*, et *<Mesure Maximale*.

La figure 4 montre les réseaux créés en réponse à la requête *le maximum des commissions des employés par département et par année*.

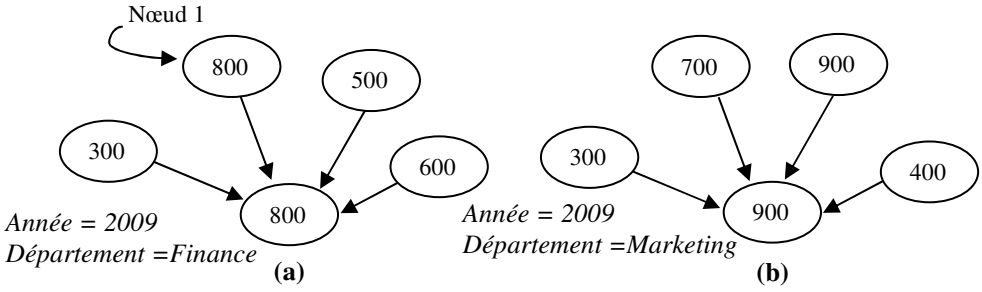

FIG. 4 – Les réseaux Bayésiens créés en réponse à la requête le maximum des commissions des employés par département et par année

Définition 3 :

Soit :

$Q = \{ S_1 , S_2, \dots , S_n\}$,

$Val_i \in S_i$

$Max\ (Q) = \{M_1, M_2, \dots , M_n\}$,

$Card\ (S_i)$ = *Nombre d'élément de* S_i,

Occ (v,S_i) = *Nombre d'occurrences de v dans* S_i

La probabilité d'inférer Val_i *est :*

$$Pr\ (\ Val_i = v \mid Max(S_i) = v\ ) = Occ\ (v,S_i) / Card\ (S_i)$$

En appliquant la définition 3 au Noeud1 du réseau Bayésien de la figure 4 (a) on obtient les probabilités indiquées dans le tableau 1. Pour le nœud 1, la valeur 1/4 de probabilité de chacun de ses deux états ('inférieur à 800' et 'égal à 800') provient du fait que les quatre nœuds ont des valeurs différentes (*cf.* TAB. 1).

Max = 800	**Pr (Nœud1 = 800)**	**Pr (Nœud1 < 800)**
Vrai	1/4	1/4

TAB. 1 – La CPT du Nœud 1

4.3 Prévention contre les inférences

Dans cette section, nous présentons les algorithmes permettant la prévention contre les inférences. Ces algorithmes utilisent les structures de données illustrées dans TAB. 2 .

L'algorithme AutoriserRequête opère sur trois étape pour autoriser ou interdire une requête de type Max qu'il reçoit en paramètre : 1) élimination de la clause Max de la requête ; 2) création du réseau Bayésien correspondant à la requête en utilisant la fonction CréerBayesNet ; enfin 3) vérification que les probabilités des différents nœuds du réseau sont inférieures au seuil donné en paramètre, si c'est le cas alors la requête est autorisée si non elle est interdite.

L'algorithme EstimerProbabilitésBayesNet permet de calculer les probabilités d'un réseau issu de l'union de deux autres réseaux. Le premier réseau Bayésien est celui qui correspond à la dernière requête lancée par l'utilisateur et le deuxième correspond à l'union de l'ensemble des réseaux Bayésiens correspondants aux anciennes requêtes autorisées. Cet algorithme utilise la procédure MettreAJourProbabilités afin de mettre à jour les probabilités du nouveau réseau. Si les probabilités sont inférieures à un seuil, alors la valeur retournée est Vrai sinon la valeur retournée est Faux. La procédure utilise l'algorithme Clustering (Stuart et al. 2003) afin de mettre à jour les probabilités du réseau Bayésien donné en paramètre. L'idée de base de Clustering est de réduire le nombre de nœuds en groupant ceux-ci en clusters. Les nœuds de chaque cluster seront remplacés un seul nœud possédant les états de ceux-ci.

Nom	**Les éléments**	**Description**
BayesNet	Nœuds : Nœud [ ]	Cette Structure de données permet de mémoriser les réseaux Bayésiens
Nœud	Valeur : Réel Probs : Probabilité [ ] Parents : Nœud [ ] Enfants : Nœud [ ]	Cette structure de données décrit un nœud du réseau Bayésien. -Valeur : Valeur du nœud -Probs : contient les probabilités d'inférer la valeur du nœud. -Parents : Les nœuds parents

		-Enfants : Les nœuds enfants
Probabilité	Valeur : Réel Max : Réel	Cette structure de données permet de mémoriser la probabilité d'avoir la valeur d'un nœud égale à une valeur Max.

TAB. 2 – Les nouvelles structures de données

Algorithme 1 : AutoriserRequête (Qm : Requête MDX, VAR BNS : BayesNet , Seuil : Réel) : Booléen

```
1:  VARIABLES
2:      Qc : Requête MDX
3:      Rst : Résultat d'une requête MDX
4:      BN : BayesNet
5:      aut : Booléen
6:  DEBUT
7:      Qc ← SupprimerMaxClause ( Qm )
8:      Rst ←  ExécuterRequête ( Qc )
9:      BN ← CréerBayesNet ( Rst )
10:     SI (Noeud_i ∈ BN , j ≤ taille (Probs): Probs[j].Valeur < Seuil ) ALORS
11:         aut ← faux
12:     SINON
13:         aut ← EstimerProbabilitésBayesNet (BN ,BNS ,Seuil)
14:     FIN SI
15:     RETOUR aut
16: FIN
```

Algorithme 2 : CréerBayesNet (Rst : Résultat d'une requête MDX) : BayesNet

```
1:  VARIABLES
2:      Bn : BayesNet
3:      n : Entier
4:  DEBUT
5:      POUR CHAQUE m DANS Mesures ( Rst )
6:          Bn.Nœuds [ i ].Valeur ← m
7:      FIN POUR
8:      n ← Nombre des Mesures
9:      Bn. Nœuds [ n ].Value ← Max (Mesures ( Rst ) )
10:     POUR CHAQUE n DANS Bn.Nodes
11:         n.Probs [ lastIndex+1 ].Max ← Max (Mesures ( Rst ) )
12:         n.Probs [ lastIndex+1 ].Valeur  ← Nombre de ( n.Valeur ) dans Rst / n
13:     FIN POUR
14:     RETOUR Bn
15: FIN
```

Algorithme 3 : EstimerProbabilitésBayesNet (BN : BayesNet , VAR BNS : , Seuil : Réel) : Booléen

```
1:  VARIABLES
2:      newBNS : BayesNet
3:
4:  DEBUT
5:      SI (BNS est Vide ) ALORS
6:          newBNS ← BN
7:      SINON
8:          newBNS ← BNS ∪ BN
9:      FIN SI
10:     MettreAJourProbabilités ( newBNS )
11:     SI (Noeud_i ∈ BN , j ≤ taille (Probs): Probs[j].Valeur < Seuil ) ALORS
12:         RETOUR faux
13:     SINON
14:         BNS ← newBNS
15:         RETOUR Vrai
16:     FIN SI
17: FIN
```

Algorithme 4 : MettreAJourProbabilités (VAR BNS : BayesNet)

```
1:  DEBUT
2:      Mettre à jour les probabilités de BNS en utilisant l'algorithme Clustering
3:  FIN
```

L'algorithme CréerBayesNet permet de créer un réseau Bayésien correspondant à une requête utilisant la fonction d'agrégation Max. L'instruction permettant la création du réseau est la plus coûteuse. Elle est linéaire en nombre de mesures utilisées par la requête. Ainsi, la complexité de l'algorithme est O(n).

L'algorithme MettreAJourProbabilités permet de mettre à jour les probabilités d'un réseau Bayésien en utilisant l'algorithme Clustering. Ce dernier ayant une complexité O(n) (Stuart et al 2003), la complexité de l'algorithme MettreAJourProbabilités est de ce fait O(n).

L'algorithme EstimerProbabilitésBayesNet permet dans une première étape de créer un nouveau réseau Bayésien à partir de l'union de deux réseaux et dans une deuxième étape de mettre à jour les probabilités du nouveau réseau en appelant la procédure MettreAJourProbabilités ; en troisième étape l'algorithme retourne la valeur booléenne Vrai dans le cas où des valeurs des probabilités sont inférieures à un seuil et Faux pour le cas contraire. Le temps de traitement concernant la première étape dépend du nombre de nœuds des deux réseaux et celui de la troisième étape du nombre de nœuds du nouveau réseau ; la complexité de l'algorithme est alors est O(n).

L'algorithme AutoriserRequête permet d'autoriser ou de refuser la requête d'un utilisateur. La première étape de l'algorithme consiste à créer le réseau Bayésien correspondant à la requête en appelant la fonction CréerBayesNet ; La complexité de cette étape est donc O(n). En deuxième étape l'algorithme compare les valeurs des probabilités par rapport à un seuil ; dans le cas où les valeurs des probabilités sont inferieures au seuil, il appel la fonction EstimerProbabilitésBayesNet et retourne le résultat de celle-ci ; dans le cas

contraire il retourne la valeur booléenne Faux. Le temps de traitement concernant cette étape dépend du nombre des nœuds du réseau, donc sa complexité est O(n).

5 Exemple

Dans cette section nous présentons un exemple démontant comment notre approche peut prévenir contre les inférences.

5.1 Exemple d'un cas d'inférence

La figure 5 contient un cube fictif permettant d'analyser les performances d'une société de commerce international. Il contient le fait commission et les deux dimensions : département, temps. Le tableau 3 regroupe les descriptions des différents paramètres.

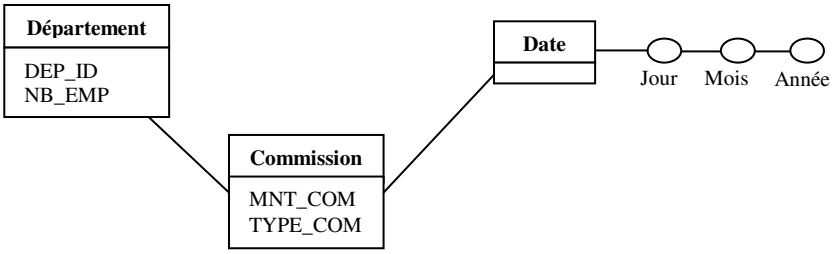

FIG. 5 – Le schéma en étoile de l'exemple

Paramètre	Dimension/Fait	Description
MNT_COM	Commission	Montant de la commission
TYPE_COM	Commission	Le type de transaction sur lequel l'employé a obtenu la commission. Les valeurs que peut prendre ce paramètre sont : Nationale et Internationale
DEP_ID	Département	Nom du département
NB_EMP	Département	Nombre d'employés du département
Jour	Date	Jour de la date de l'obtention de la commission
Mois	Date	Mois de la date de l'obtention de la commission
Année	Date	Année de la date de l'obtention de la commission

TAB. 3 – Les paramètres des dimensions et du fait

Supposons que : la société possède deux département Finance et Marketing. ; le nombre d'employés de ce dernier est 4 ; Alice et Bob sont les seuls employés du département Marketing habilités à faire des transactions internationales ; et que Alice n'a pas travaillé en décembre 2009 parce qu'elle a eu un congé de maladie. Est-il possible d'inférer le nom de l'employé du département Marketing ayant la commission maximale ?

A partir du résultat de la première requête (*cf.* TAB. 4) : maximum des commissions par nombre d'employés et par département, et le résultat de la deuxième requête (*cf.* TAB. 5)

maximum des commissions par année et par mois, il possible d'inférer que la commission maximale du département Marketing a été obtenue au mois de décembre. A partir du résultat de la troisième requête (*cf.* TAB. 6) : maximum des commissions par année et par type de commission, il est possible d'inférer que l'employé ayant eu la commission maximale est Bob.

NB_EMP	DEP_ID	MNT_COM
4	Marketing	900
	Finance	950

TAB. 4 – Le résultat de la requête 1

Année	Mois	Max commissions
2009	Octobre	850
	Novembre	720
	Décembre	900

TAB. 5 – Le résultat de la requête 2

Année	TYPE_COM	Max commissions
2009	Nationale	840
	Internationale	900

TAB. 6 – Le résultat de la requête 3

5.2 Prévention contre le cas d'inférence

Nous fixons la valeur du seuil à 1/2. La figure 6 (a) présente le réseau Bayésien correspondant à la première requête (par soucis de clarté uniquement, les commissions du département marketing sont représentées). Les nœuds correspondent aux différentes commissions. Le nombre de ceux-ci est 10 donc la probabilité d'inférer la commission de Bob est 1/10. La figure 6 (b) montre le réseau Bayésien correspondant à la deuxième requête. En faisant l'union du réseau Bayésien de la première requête et celui de la deuxième, nous obtenons le réseau de la figure 7. L'union fait croitre la probabilité d'inférer la commission de Bob à 1/4. La figure 8 montre le réseau Bayésien correspondant à la troisième requête. L'union de ce dernier réseau (*cf.* FIG. 9) avec les réseaux des deux premières requêtes fait croitre la probabilité d'inférer la commission de Bob à 1/2. Puisque celle-ci devient égale au seuil, le résultat de la troisième requête ne sera pas délivré à l'utilisateur.

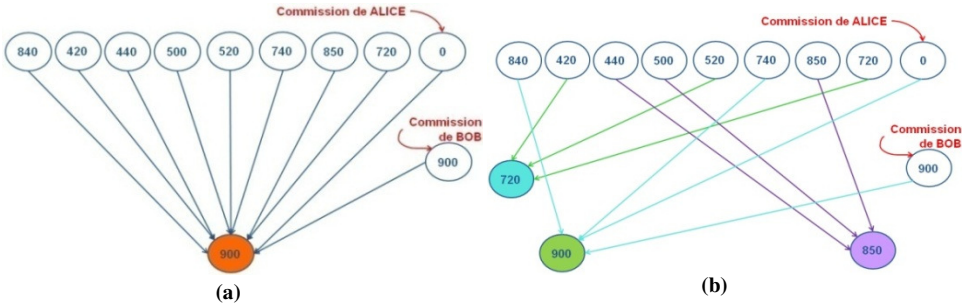

FIG. 6 – Les réseaux Bayésiens correspondant aux deux premières requêtes

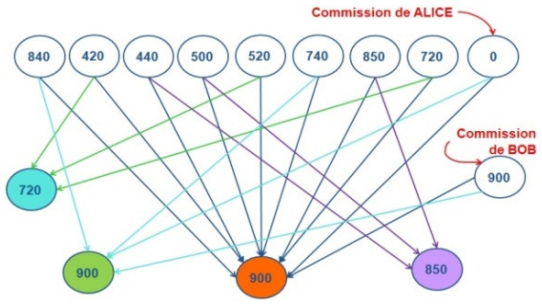

FIG. 7 – Le résultat de l'union des réseaux Bayésiens de la première et la deuxième requête

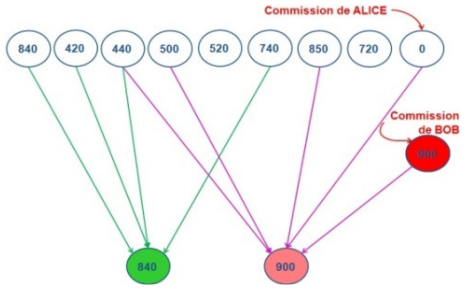

FIG. 8 – Le réseau Bayésien correspondant à la troisième requête

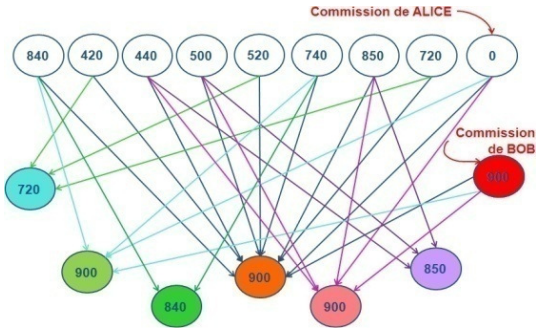

FIG. 9 – Le résultat de l'union des réseaux Bayésiens correspondant aux trois requêtes

6 Conclusion

Dans cet article, nous avons présenté une approche permettant la prévention contre les inférences dans les entrepôts de données sur la base des réseaux Bayésiens.

Un ensemble de définitions pour traduire une requête en réseau Bayésiens a été proposé. A partir de ces définitions des algorithmes ont été élaborés en vu de construire les réseaux Bayésiens correspondants aux requêtes des utilisateurs et d'interdire celles qui sont susceptibles de générer des inférences. La complexité de ces algorithmes est linéaire en nombre de valeur dans l'entrepôt.

L'approche présentée tient compte des requêtes utilisant les fonctions d'agrégation Max et Min. Des investigations concernant les autres types de requêtes sont envisageables ainsi que l'allégement du traitement effectué par les réseaux Bayésiens et la détermination de la valeur du seuil.

Références

Agrawal, R., Srikant, R. et Thomas, D. (2005). Privacy-Preserving OLAP. In: ACM SIGMOD, pp.251–262.

Beck L.L. (1980). A security mechanism for statistical databases. ACM Trans. on Database Systems, 5(3):316–338.

Bhargava B. (2000). Security in data warehousing (invited talk). In Proceedings of the 3rd Data Warehousing and Knowledge Discovery (DaWak'00).

Chin, F.Y., et Ozsoyoglu, G. (1982). Auditing and Inference Control in Statistical Databases. IEEE Transactions on Software Engineering 8(6), 574–582.

Denning D.E. et Schlörer J. (1983). Inference controls for statistical databases. IEEE Computer, 16(7):69–82.

Dobkin D., Jones A.K., et Lipton R.J. (1979). Secure databases: protection against user influence. ACM Trans. on Database Systems, 4(1):97–106.

Export (2008). http://www.export.gov/safeharbor/

HHS (1996). http://www.hhs.gov/ocr/privacy/index.html

Hua, M., Zhang, S., Wang, W., Zhou, H. et Shi, B. (2005). FMC: An Approach for Privacy Preserving OLAP. In: Tjoa, A.M., Trujillo, J. (eds.) DaWaK 2005. LNCS, vol. 3589, pp. 408–417.Springer, Heidelberg

Malvestuto F.M., Mezzani M. et Moscarin M. (2006). Auditing Sum-Queries to Make a Statistical Database Secure. ACM Transactions on Information and System Security 9(1), 31–60.

Pernul G., Priebe T. (2000). Towards olap security design - survey and research issues. In Proceedings of 3rd ACM International Workshop on Data Warehousing and OLAP (DOLAP'00), pages 114–121.

GPO (1999). http://www.gpo.gov/fdsys/pkg/PLAW-106publ102/content-detail.html

Salah T., Jamel F., Hanene BEN-ABDALLAH, Nouria H, (2009). Sécurisation des entrepôts de données : Etat de l'art et proposition d'une architecture. Quatrième Atelier sur les Systèmes Décisionnels. 10-11 novembre 2009, Jijel, Algérie

Schlërer J. (1981). Security of statistical databases: multidimensional transformation. ACM Trans. Database Systems, 6(1):95–112.

Soxlaw (2002). http://www.soxlaw.com/

Shoshani A. (1997). OLAP and statistical databases: Similarities and differences. In Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'97), pages 185–196.

Soler, E., V. Stefanov, J.-N. Mazón, Trujillo J., Fernández-Medina E., et Piattini M. (2008). Towards comprehensive requirement analysis for data warehouses : Considering security requirements. In ARES, pp. 104–111. IEEE Computer Society.

Soler, E., Villarroel R., Trujillo J., Fernández-Medina E., et Piattini M. (2006). Representing security and audit rules for data warehouses at the logical level by using the common warehouse metamodel. In ARES, pp. 914–921. IEEE Computer Society.

Stuart J. Russell N., Peter N., (2003). *Artificial Intelligence A Modern Approach Second Edition*. New Jersey : Pearson Education, Inc.

Sung, S.Y., Liu, Y., Xiong, H. et Ng, P.A. (2006). Privacy Preservation for Data Cubes. Knowledge and Information Systems 9(1), 38–61.

Traub J.F., Yemini Y., et Wozniakowski H. (1984). The statistical security of a statistical database. ACM Trans. on Database Systems, 9(4):672–679.

Zhang, N., Zhao, W. et Chen, J. (2004). Cardinality-based Inference Control in OLAP Systems: An Information Theoretic Approach. In: ACM DOLAP, pp. 59–64.

Summary

Data warehouses provide business leaders and analysts to make strategic decisions and make predictions. Securing data warehouse is therefore important. Moreover, securing a data warehouse should not be very restrictive preventing effective use of the data warehouse, nor too loose allowing the inference of prohibited data (i.e. personal, confidential). In this article we examine the security of data warehouses using an approach based on Bayesian networks. This has two advantages: it requires no further treatment after each feeding phase of the data warehouse and does not involve alteration of the original data.