

Towards an Automatic Detection of Sensitive Information in a Database

Cédric du Mouza, Elisabeth Métais, Nadira Lammari, Jacky Akoka,
Tatiana Aubonnet, Isabelle Comyn-Wattiau, Hammou Fadili and
Samira Si-Saïd Cherfi

Lab. CEDRIC, CNAM Paris, France

Context and Problem

Context

- to test and validate new applications developers need realistic data
- final tests generally performed on excerpts from the on-going production databases
- recent phenomenon of the externalization of any development and test

Problem

- information in many databases is proprietary and must be protected
- existing proposals lack an automatic detection of the sensitive data

Motivating examples

Hospital database

- [data] all personal and medical information about patients
- [risk] any person developing an application on the medical data not to be able to extract any personal information about a patient



Motivating examples

Clients database

- [data] all information and coordinates of the different clients of a large company
- [risk] a leak of information can cause considerable business damage if transmitted to a competitor



Our Proposal

Main features of our approach

- Automatic detection of the values to be scrambled
- Automatic propagation to other semantically linked values

Techniques used

- a rule based approach implemented under an Expert System architecture
- a semantic graph to ensure the propagation of the confidentiality and the consistency with the other relations

Sensitive data

Sensitive attributes

The confidential attributes set, denoted $\mathcal{S}_c \subseteq \mathcal{S}$ is the set of attributes whose content is confidential, whatever the number of occurrences they have.

Identifying attributes

The identity attributes set, denoted $\mathcal{S}_i \subseteq \mathcal{S}$ is the set of attributes such that for any $x \in \mathcal{S}_i$ it exists a subset $s_i \subseteq \mathcal{S}_i$ within a single table \mathcal{T} and with $x \in s_i$, such that:

- (i) each instance of s_i occurs less than k times in the records from \mathcal{T}
- (ii) there is an attribute $y \in \mathcal{S}_c$ in \mathcal{T} .

Confidential attributes

The sensitive attributes set, denoted \mathcal{S}_s , is the set of identifying and confidential attributes, *i.e.*, $\mathcal{S}_s = \mathcal{S}_i \cup \mathcal{S}_c$.

Sensitive data

Sensitive attributes

The confidential attributes set, denoted $\mathcal{S}_c \subseteq \mathcal{S}$ is the set of attributes whose content is confidential, whatever the number of occurrences they have.

Identifying attributes

The identity attributes set, denoted $\mathcal{S}_i \subseteq \mathcal{S}$ is the set of attributes such that for any $x \in \mathcal{S}_i$ it exists a subset $s_i \subseteq \mathcal{S}_i$ within a single table \mathcal{T} and with $x \in s_i$, such that:

- (i) each instance of s_i occurs less than k times in the records from \mathcal{T}
- (ii) there is an attribute $y \in \mathcal{S}_c$ in \mathcal{T} .

Confidential attributes

The sensitive attributes set, denoted \mathcal{S}_s , is the set of identifying and confidential attributes, *i.e.*, $\mathcal{S}_s = \mathcal{S}_i \cup \mathcal{S}_c$.

Sensitive data

Sensitive attributes

The confidential attributes set, denoted $\mathcal{S}_c \subseteq \mathcal{S}$ is the set of attributes whose content is confidential, whatever the number of occurrences they have.

Identifying attributes

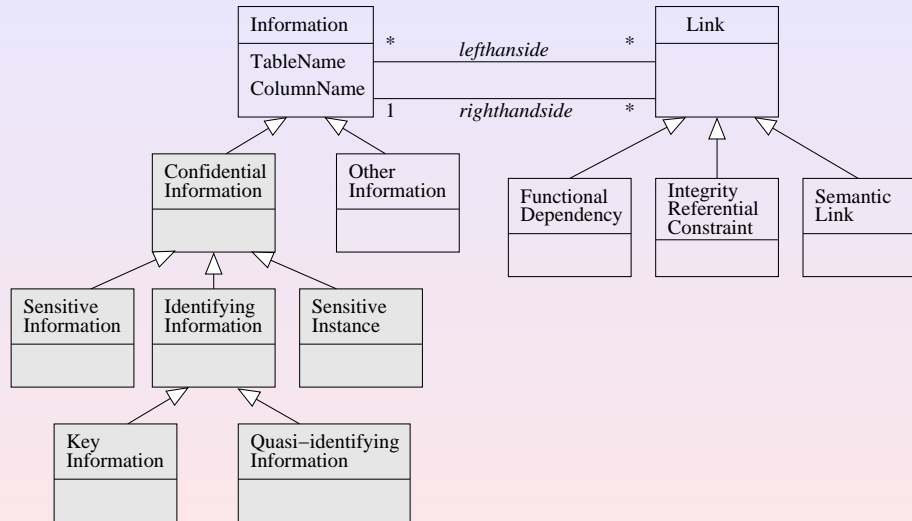
The identity attributes set, denoted $\mathcal{S}_i \subseteq \mathcal{S}$ is the set of attributes such that for any $x \in \mathcal{S}_i$ it exists a subset $s_i \subseteq \mathcal{S}_i$ within a single table \mathcal{T} and with $x \in s_i$, such that:

- (i) each instance of s_i occurs less than k times in the records from \mathcal{T}
- (ii) there is an attribute $y \in \mathcal{S}_c$ in \mathcal{T} .

Confidential attributes

The sensitive attributes set, denoted \mathcal{S}_s , is the set of identifying and confidential attributes, *i.e.*, $\mathcal{S}_s = \mathcal{S}_i \cup \mathcal{S}_c$.

Meta-model of confidential data



Why considering all confidential attributes?

We observe that:

- The scrambling of the identity attributes preserves anonymity while confidential attributes keep their initial distribution.
- The scrambling of the sensitive attributes aims at protecting individual privacy by modifying the value of sensitive attributes while information that identifies persons remains unchanged.

Example

A HRD database storing information concerning employees: employee's id, name, city, department, name of the superior, wage, etc.

- the first two properties permit to identify an employee ($\mathcal{S}_i = \{id, name\}$), and thus to consult all his data
- one may avoid to reveal the highest salary or the average salary of a given department \Rightarrow considered as sensitive ($\mathcal{S}_c = \{wage\}$).
- in smaller companies, the couple (city,department) is sufficient to identify a small subset of employees \Rightarrow must be added to \mathcal{S}_i . For larger companies this information is not identifying enough.
- finally for our large company we have to scramble $\mathcal{S}_s = \{id, name, wage\}$.

The rule-based approach

Let Δ be the set of all possible domains of application, Θ the set of all possible table names, Φ the set of all possible attribute names and Ψ the set of all possible attribute values.

Rule condition

A *rule condition* $\chi = \chi_1 \boxplus \chi_2$ is a condition with $\chi_1 \in \{\text{domainName}, \text{tableName}, \text{attributeName}, \text{attributeValue}\}$, $\chi_2 \in \Delta \cup \Theta \cup \Phi \cup \Psi$, and \boxplus is an operator in $\{=, \neq, <, >, \leq, \geq, \text{contains}, \text{!contains}\}$.

Rule

A *rule* is composed by disjunctions and conjunctions of rule conditions along a rule sensitivity score $\sigma \in [0, 1]$, where σ permits to evaluate how sensitive is an attribute that satisfies the rule.

Rule example

Assume we consider that a column whose name contains “salar” if the domain is HRD and there are values greater than 15,000 or lower than 5,000 is highly sensitive (score=0.9). The corresponding rule is expressed by the following expression:

$$\begin{aligned} & ((\text{domainName} = 'HRD') \\ & \wedge (\text{attributeName contains 'salar'}) \\ & \wedge (\text{attributeValue} > 15000 \\ & \vee \text{attributeValue} < 5000)) , 0.9 \end{aligned}$$

Other detection techniques: statistics

The statistical computation

Some candidates for \mathcal{S}_i can be found thanks to:

- metabase (primary key and unique integrity constraints) for identifiers
- statistics, generally stored in the metabase for query optimization purpose, like attribute's selectivity, useful for pseudo-identifiers

but...

determining all the subsets of attributes that are quasi-identifiers is a *NP*-hard problem (but heuristics in literature)

Other detection techniques:NLP

Necessity of Natural Language Processing

- attributes may not have been named with exactly the same word that the one used in the rules
- matching using NLP techniques (currently only a semantic matching based on Wordnet)

$SIMILAR(att_name, att_desc, att_name_in_rule) \rightarrow s_{sim}$

- $s_{sim} = 0$ if antonyms
- $s_{sim} = 1$ if members of the same synset or same word in different languages
- $s_{sim} \in [0, 1]$ regarding the distance to a common ancestor

Propagation graph

Integrity and referential links

- foreign key attribute references a primary or secondary key attribute \Rightarrow any modification of the former must impact the latter
- same problem with attribute in a table with same semantics than another one in another table

We build for any set $P \subseteq \mathcal{S}$, the result set of links

$$\Gamma(P) = \bigcup_{x \in 2^{|\mathcal{P}|}} \gamma(x)$$

where $\gamma : 2^{|\mathcal{S}|} \rightarrow 2^{|\mathcal{S}|}$ is defined as

$$\forall x \in 2^{|\mathcal{S}|}, \gamma(x) = \begin{cases} \{y \mid y \in 2^{|\mathcal{S}|}, y \text{ referring or semantically linked to } x\} \\ \emptyset \text{ otherwise} \end{cases}$$

Propagation algorithm

We use the referential and semantical links between attributes to extend the set of attributes \mathcal{S}_s^{init} identified for scrambling:

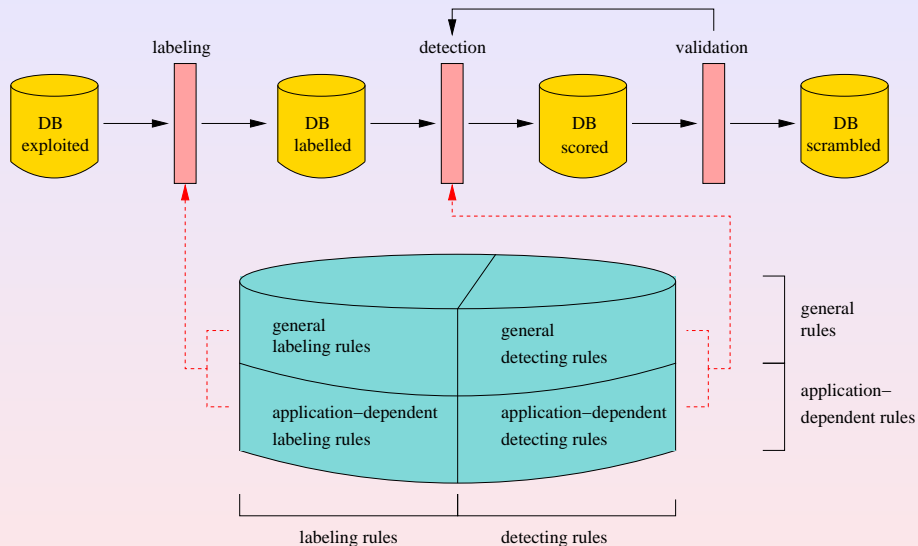
Propagation algorithm

- (i) $\mathcal{S}_s^{(0)} = \mathcal{S}_s^{init}$
- (ii) $\mathcal{S}_s^{(k+1)} = \mathcal{S}_s^{(k)} \cup \Gamma(\mathcal{S}_s^{(k)})$

Lemma (convergence)

The algorithm converges to \mathcal{S}_s with at most $|\mathcal{S}|$ iterations.

The scrambling process



Labelling the database

Problem

Detecting rules concern attributes whose name in rules may differ from the one in the database

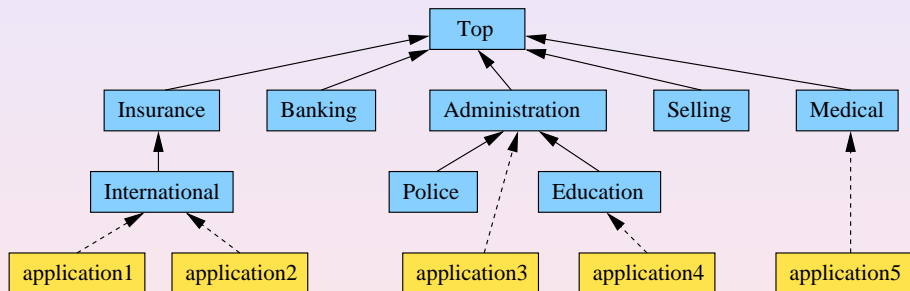
example: “salaries have to be scrambled”... but in the DB attribute is named “wage”

solutions

- write a rule for any synonyms of a word → many rules so detection becomes too costly
- label the database by re-naming any column with a “representant” name, that is also used when writing rules

Detection of confidential attribute

Detection rules applied on the labelled database. there are generic rules and domain/application-dependant rules.

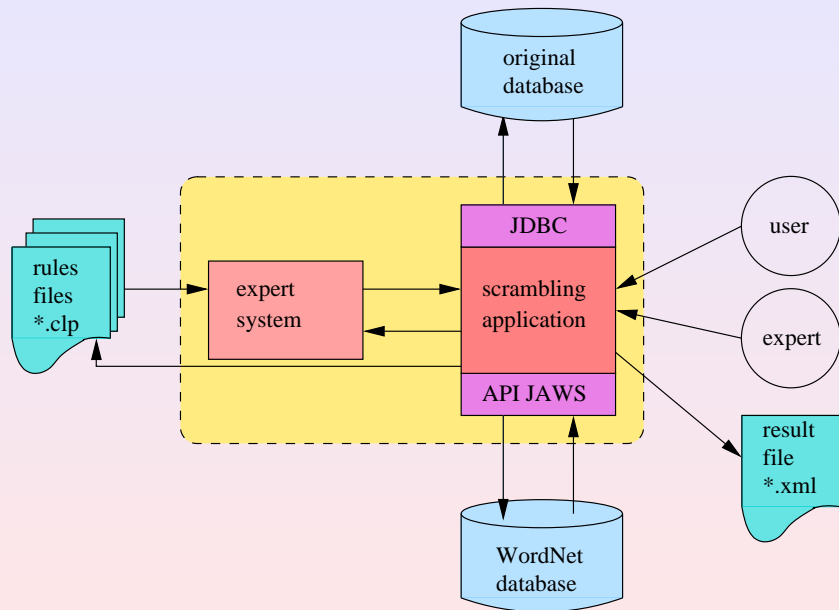


Expert validation

After automatic detection and automatic propagation, we propose to the expert:

- a direct access to the base of rules filtered according to his domain according to the hierarchy presented before;
- a clear vision of data samples (instances) across several tables with confidentiality scores deduced ← allows the expert to directly point to the attribute he disagrees with and to correct its level of confidentiality.

Prototype architecture



Prototype interface

Tool for detection of sensitive attributes in a database

File Expert System

Save & Exit Do Not Modify

List tables

- REGIONS
- LOCATIONS
- DEPARTMENTS
- JOBS
- EMPLOYEES
- JOB_HISTORY
- TABLETEST
- COUNTRIES

TABLES

- EMPLOYEES
- JOB_HISTORY
 - EMP_ID
 - START_DATE
 - END_DATE
 - AFFECTATION
 - DEP_ID

Display selected table

TABLE=>DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700

TABLE=>EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUM...	HIRE_DATE	JOB_ID	SALARY	COMMISSION	MANAGER_ID	DEPARTMENT...
100	Steven	King	SKING	515.123.4567	1987-06-17 00:00:00.0	AD_PRES	24000			90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	1989-09-21 00:00:00.0	AD_VP	17000		100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	1993-01-13 00:00:00.0	AD_VP	17000		100	90

TABLE=>JOB_HISTORY

EMP_ID	START_DATE	END_DATE	AFFECTATION	DEP_ID
102	1993-01-13 00:00:00.0	1998-07-24 00:00:00.0	IT_PROG	60
101	1989-09-21 00:00:00.0	1993-10-27 00:00:00.0	AC_ACCOUNT	110
101	1993-10-26 00:00:00.0	1997-03-15 00:00:00.0	AC_MGR	110

For a lonely visualisation : Enter the name of the table Launch research

Details of results

Semantic equivalence

Experiments: datasets

Database	nb of tables	nb of attributes	nb of rows
Dell store	8	52	172,716
IMDB	47	151	1,834,483
Media Wiki	45	289	756
Order	8	59	3459

Experiments: results

Database	total time (in s)	confidential attributes (nb and %)	
		mod. ($\sigma \leq 50$)	high ($\sigma > 50$)
Dell store	24.9	25 (48%)	13 (25%)
IMDB	566.8	44 (29%)	15 (10%)
Media Wiki	1.6	5 (11%)	2 (04%)
Order	1.9	25 (42%)	21 (36%)

Database	Experts evaluation			
	missed	f/p	inadeq.	approv.
Dell store	2	5	2	32
IMDB	3	9	8	42
Media Wiki	2	1	0	6
Order	3	4	4	38

Conclusion

Our proposal

- a meta-model of confidential information
- a rule-based approach for determining the attribute's sensitivity level
- integrity referential constraints and semantic links are used for the propagation of the sensitivity

Future work

- development of the NLP techniques
- automatically determining of the scrambling algorithms to use on confidential data
- more validation on larger databases thanks to experts

Thanks for your attention