Abstract Non-Interference on Databases (Abstract)

Isabella Mastroeni and Rosalba Rossato

Dipartimento di Informatica - Università degli Studi di Verona
Ca' Vignal 2 - Strada Le Grazie 15 - 37134 Verona (Italy)
{isabella.mastroeni@, rossato@sci.}univr.it

One of the main tasks of information security policies is to enforce confidentiality of data. Confidentiality, or non-interference, refers to the protection of data against unauthorized disclosure of sensitive information, in particular we will focus on confidentiality for multilevel secure databases. The database management system (DBMS) must provide techniques to enable certain users to access selected portions of a database without gaining access to the rest of the database. The standard approach to protect multilevel secure databases from violations of confidentiality consists in the mechanism of *mandatory access control*. The idea is to classify data and users in terms of *security classes*, *i.e.,* a user has a certain clearance and he can access/modify only those data whose security level is dominated by this clearance. Unfortunately, this kind of mechanisms does not completely guarantee information confidentiality, since unauthorized releases of information may occur due to "implicit dependencies" between private and public information (so called *inference channels* [2]).
In the following we consider the relational data model and we aim to provide a *semantic* characterization of security policies in order to avoid *inference* of sensitive information from queries on public data. In particular we look for a general semantic model where the security policies are *parametric* on *properties* both of protected and observed data.

The relational model represents the database as a collection of *relations*, each one representing a collection of related data values [3]. In the formal relational model terminology, a row is called *tuple*; a column header *attribute*, representing the time-invariant property of each column; the table *relation*. Consider, for instance, a database composed by the relation *Employee*, which stores information about employees of an organization. *Employee* is defined on the attributes set {Name, SSN, BDate, Address, Sex, Salary, Dno}. It keeps trace of the name, the social security number, the birthdate, the address, the sex, the salary, and the department number where the employe works. Table 1 shows an instance of the relations *Employee*.

In the following we consider the framework of Abstract Interpretation (AI) [4, 5] for modeling the concept of *property*. A property is intended as the set of all the concrete elements having the property, *e.g.,* on natural numbers, the property "even" corresponds to the set of all the even numbers, analogously the property $[1500, 2000[$ denotes the set $\left\{ x \in \mathbb{N} \,\middle|\, 1500 \leq x < 2000 \right\}$. Abstract Non-Interference (ANI) states that *properties* of observable data don't depend on *properties* of sensitive data.

Applying ANI to Database Security. Consider the attribute SALARY *private* in Table 1, *i.e.,* it is not allowed to retrieve/use information about it. We can differently interpret the term *private* in the relational database context:

Direct Access: Given the name of the employee, it is not possible to retrieve his/her salary. In the standard security approach this corresponds exactly to a *view* which cuts the column SALARY. In our model it would correspond to *abstract* the information SALARY, observing the property "I don't know". This approach can be applied also when it is *possible* to retrieve *some* information about SALARY, *e.g.,* for statistics, but it is not possible to associate it with the corresponding employee, *abstracting* the information NAME to the property "I don't know".

Partial Access: It is possible to retrieve *only* partial information about SALARY. To the best of our knowledge this is not considered in the standard security approach. In our model this would correspond to *abstracting* the values of SALARY to the *observable property*. In Table 2 we suppose that only intervals of salaries are observable, *i.e.,* it is avoided to know the value of the salary, but the interval of values is considered public.

Our aim is to exploit this semantic characterization of security policies for certifying the security level of a given multilevel secure database from different points of view following the ideas proposed for ANI in language-based security [6]:

- Given a set of queries on the database, we want to characterize the *maximal amount* of sensitive information that can be inferred by these queries.
- Given the database, we want to characterize the "most concrete", *i.e.,* the less manipulated, view which avoids inference of sensitive information through any possible query.
- We want to generalize the model in order to abstract data not only by means of *attribute independent* properties but also by using *attribute dependent*[1] abstractions, *i.e.,* the same value of salary can be abstracted in different properties *depending on*, for example, the department number (DNO).

[1] These are the terms used in the standard framework of abstract interpretation [4, 5].

Name	SSN	BDate	Address	Sex	Salary	Dno
Smith	104	250570	5^{th} Avenue	M	1.500	4
Benson	124	100468	Castle Spring	F	1.800	5
Alicia	345	190768	980 Dallas	F	2.000	5
Borg	555	101137	450 Stone	M	2.500	1

Table 1. An instance of the relation *Employee*.

Name	SSN	BDate	Address	Sex	Salary	Dno
Smith	104	250570	5^{th} Avenue	M	[1.500,2000[	4
Benson	124	100468	Castle Spring	F	[1.500,2000[	5
Alicia	345	190768	980 Dallas	F	[2.000,2500[	5
Borg	555	101137	450 Stone	M	[2.500,3000[	1

Table 2. An interval abstraction of Salary.

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[2] A. Brodsky, C. Farkas, and S. Jajodia. Secure databases: Constraints, inference channels, and monitoring disclosures. *IEEE Trans. on Knowledge and Data Engineering*, 12(6):900–919, 2000.

[3] E.F. Codd. A relation model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.

[4] P. Cousot and R. Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proc. of Conf. Record of the 4th ACM Symp. on Principles of Programming Languages* (*POPL '77*), pages 238–252, New York, 1977. ACM Press.

[5] P. Cousot and R. Cousot. Systematic design of program analysis frameworks. In *Proc. of Conf. Record of the 6th ACM Symp. on Principles of Programming Languages* (*POPL '79*), pages 269–282, New York, 1979. ACM Press.

[6] R. Giacobazzi and I. Mastroeni. Abstract non-interference: Parameterizing non-interference by abstract interpretation. In *Proc. of the 31st Annual ACM SIGPLAN-SIGACT Symp. on Principles of Programming Languages (POPL '04)*, pages 186–197, New York, 2004. ACM-Press.

[7] J. A. Goguen and J. Meseguer. Unwinding and inference control. In *Proc. IEEE Symp. on Security and Privacy*, pages 75–86, Los Alamitos, Calif., 1984. IEEE Comp. Soc. Press.

[8] A. Sabelfeld and A.C. Myers. Language-based information-flow security. *IEEE J. on selected ares in communications*, 21(1):5–19, 2003.