available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

The inference problem: Maintaining maximal availability in the presence of database updates☆

Tyrone S. Toland [a,*], Csilla Farkas [b], Caroline M. Eastman [b]

[a] Department of Informatics, University of South Carolina Upstate, 800 University Way, Spartanburg, SC 29303, USA
[b] Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208, USA

ARTICLE INFO

Article history:
Received 19 February 2009
Received in revised form
18 July 2009
Accepted 21 July 2009

Keywords:
Disclosure inference
Confidentiality
Availability
Updates
Access control
Collaboration
Collusion
Database

ABSTRACT

In this paper, we present the Dynamic Disclosure Monitor (D^2Mon) architecture to prevent illegal inferences via database constraints. D^2Mon extends the functionality of Disclosure Monitor (DiMon) to address database updates while preserving the soundness and completeness properties of the inference algorithms. We study updates from the perspective of increasing data availability. That is, updates on tuples that were previously released may affect the correctness of the user inferences over these tuples. We develop a mechanism, called *Update Consolidator* (UpCon), that propagates updates to a history file to ensure that no query is rejected based on inferences derived from outdated data. The history file is used by the Disclosure Inference Engine (DiIE) to compute inferences. We show that UpCon and DiIE working together guarantee *confidentiality* (completeness property of the data-dependent disclosure inference algorithm) and maximal *availability* (soundness property of the data-dependent disclosure inference algorithm) even in the presence of updates. We also present our implementation of D^2Mon and our empirical results.

1. Introduction

During the last couple of decades, our society became increasingly dependent on computerized information resources. Electronic databases contain information with sensitivity levels ranging from public (e.g., airline schedules, phone numbers and addresses) to highly sensitive (e.g., financial and medical information, military research). The aim of information security policies is to protect the *confidentiality* (secrecy) and *integrity* of data, while ensuring data *availability*. Access control models, such as discretionary access control (DAC), mandatory access control (MAC), and role-based access control (RBAC), prevent direct unauthorized accesses to data. However, these models are unable to protect against indirect data accesses when unauthorized information is obtained via inference channels. An inference channel is the ability to determine sensitive data using non-sensitive data (Jajodia and Meadows, 1995). Most of the inference channels in relational databases are established by combining *metadata* (e.g., database constraints) with non-sensitive data to obtain sensitive information.

Techniques to detect and remove inference channels can be organized into two categories. The first category includes techniques that detect inference channels *during database design* time (Buczkowski, 1990; Dawson et al., 1999b; Goguen and Meseguer, 1984; Hinke, 1988; Hinke et al., 1995; Marks, 1996; Morgenstern, 1988; Smith, 1990; Su and Ozsoyoglu, 1991; Yip and Levitt, 1998). Inference channels are removed by

☆ This work was partially supported by the National Science Foundation under grant numbers IIS-0237782 and P200A000308-02.
* *Corresponding author*. Tel.: +1 864 503 5310; fax: +1 864 503 5408.
E-mail addresses: ttoland@uscupstate.edu (T.S. Toland), farkas@cse.sc.edu (C. Farkas), eastman@cse.sc.edu (C.M. Eastman).
0167-4048/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cose.2009.07.004

modifying the database design or by increasing the classification levels of some of the data items. Techniques in the second category seek to eliminate inference channel violations during *query processing time* (Denning, 1985; Keefe et al., 1989; Marks et al., 1996; Mazumdar et al., 1988; Stachour and Thuraisingham, 1990; Thuraisingham, 1987). If an inference channel is detected, the query is either refused or modified to avoid security violations. While, in general, database design time approaches are computationally less expensive and may provide better on-line performance than query processing time approaches, database design time approaches often result in over-classification of data items, thus reducing data availability. Query processing time approaches allow greater availability by analyzing the data released to the user.

In this paper, we study database updates from the perspective of increasing data availability. For this, we propose a system that uses a query processing time algorithm to control inferences that reveal sensitive data. Our work extends the Disclosure Monitor (DiMon) model presented by Brodsky et al. (2000). DiMon detects and eliminates inference channels based on database constraints. A Disclosure Inference Engine (DiIE) is proposed, that generates all information that can be disclosed based on a user's previous query results, the current query results, and a set of Horn-clause constraints (Ullman, 1988). However, Brodsky et al. do not consider dynamic databases where updates may violate the soundness property of the data-dependent disclosure inference algorithm. The following are two examples of an execution of DiMon. The first example satisfies both the soundness and completeness property, while the second example violates the soundness property.

For both examples, we use the *Employee* relation containing information about the name, rank, salary, and department of employees (See Table 1). The relation satisfies the functional dependency (FD) *RANK* & *SALARY*. The security requirement is that the employees' salaries can only be accessed by authorized users; that is, partial tuples over attributes *NAME* and *SALARY* can only be accessed by authorized users. However, to increase data availability, unauthorized users are allowed to access *NAME* and *SALARY* separately.

Example 1.1. Suppose an unauthorized user submits the following two queries:

Query 1: ''List the name and rank of the employees working in the toy department.'' ($\Pi_{NAME,\ RANK}\sigma_{DEPARTMENT='Toy'}$)
Query 2: ''List the salaries of all clerks in the appliance department.'' ($\Pi_{SALARY}\sigma_{RANK='Clerk'\ \wedge\ DEPARTMENT='Appliance'}$)

The answer to the queries is as follows: **Query 1** = {⟨*John, Clerk*⟩,⟨*Mary,Secretary*⟩} and **Query 2** = {⟨*Clerk*, $38,000⟩}. Since the *Employee* relation satisfies the FD *RANK* → *SALARY*, these query results reveal that John's salary is $38,000. These types of inferences are correctly detected and prevented by DiIE. That is, DiIE would reject **Query 2** to prevent the user from inferring the unauthorized information.

Example 1.2. Now, suppose the following is submitted:

Query 1: ''List the name and rank of the employees working in the toy department.'' ($\Pi_{NAME,\ RANK}\sigma_{DEPARTMENT='Toy'}$)
Update: ''John is promoted to the rank of manager.''
Update: ''The salaries of all the clerks are increased by 4%.''
Query 2: ''List the salaries of all clerks in the appliance department.''
($\Pi_{SALARY}\sigma_{RANK='Clerk'\wedge DEPARTMENT='Appliance'}$)

The answer to **Query 1** is the same as in Example 1.1. As a result of the database updates shown in Table 1, the answer to **Query 2** is {⟨*Clerk*, $39,520⟩}. In this case, DiIE would indicate that ⟨*John*, $39,520⟩ is disclosed; and, since it is unauthorized, **Query 2** would be rejected. However, this sub-tuple is not contained in the updated relation and has never been present in any of the earlier versions. Therefore, the second query could be safely answered without revealing any unauthorized information.

The reason that the second query can be released in Example 1.2 is that the FD was applied on outdated data values. We call inferences based on outdated data values ''wrong'' inferences. That is, the ''wrong'' inference that John's salary is $39,520 does not reveal unauthorized data under the security requirement specified in this example. We will permit queries that generate ''wrong'' inferences in order to increase data availability.

We evaluate the disclosure inference algorithms in our model from the perspective of *soundness* and *completeness*. Intuitively, soundness means that only existing disclosure inferences are generated by the algorithm (data availability); completeness means that all existing disclosure inferences are generated (secrecy).

Our preliminary work, Farkas et al. (2001), formalizes our initial results and presents our conceptual framework, called Dynamic Disclosure Monitor (D^2Mon). D^2Mon guarantees data confidentiality and maximal availability even in the presence of inferences and updates. We do not propose to replace an existing access control mechanism. It is our intention to complement an existing access control mechanism (e.g., DAC, MAC, RBAC) to address the inference problem. To this end, we assume the existence of an access control mechanism that utilizes a predefined security classification.

Table 1 – Employee relation.

NAME	RANK	SALARY	DEPT.
Original			
John	Clerk	38,000	Toy
Mary	Secretary	28,000	Toy
Chris	Secretary	28,000	Marketing
Joe	Manager	45,000	Appliance
Sam	Clerk	38,000	Appliance
Eve	Manager	45,000	Marketing
Updated			
John	**Manager**	**45,000**	Toy
Mary	Secretary	28,000	Toy
Chris	Secretary	28,000	Marketing
Joe	Manager	45,000	Appliance
Sam	Clerk	**39,520**	Appliance
Eve	Manager	45,000	Marketing