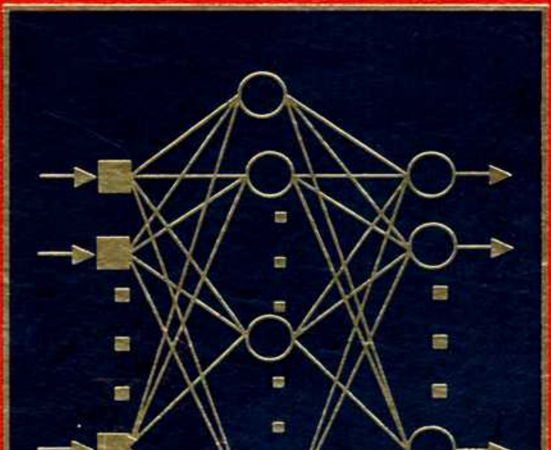
Advances in

COMPUTERS

Volume 38



Edited by

MARSHALL C. YOVITS

Advances

in COMPUTERS

VOLUME 38

Contributors to This Volume

B. Chandrasekaran
Jose A. B. Fortes
Kumar N. Ganapathy
John M. Long
Abbe Mowshowitz
Günther Pernul
Weijia Shang
Benjamin W. Wah

Advances in COMPUTERS

EDITED BY

MARSHALL C. YOVITS

Purdue School of Science
Indiana University—Purdue University at Indianapolis
Indianapolis, Indiana

VOLUME 38

ACADEMIC PRESS
Boston San Diego New York
London Sydney Tokyo Toronto

This book is printed on acid-free paper. ∞

Copyright © 1994 by Academic Press, Inc.

All rights reserved.
No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission in writing from the publisher.

ACADEMIC PRESS, INC.
A Division of Harcourt Brace & Company
525 B Street, Suite 1900, San Diego, CA 92101-4495

United Kingdom Edition published by
ACADEMIC PRESS LIMITED
24–28 Oval Road, London NW1 7DX

Library of Congress Catalog Card Number: 59–15761
International Standard Serial Number: 0065-2458
International Standard Book Number: 0-12-012138-7

Printed in the United States of America

94 95 96 97 BC 9 8 7 6 5 4 3 2 1

Contents

Database Security

Günther Pernul

Functional Representation and Causal Processes

B. Chandrasekaran

Computer-Based Medical Systems

John M. Long

Algorithm-Specific Parallel Processing with Linear Processing Arrays

Jose A. B. Fortes, Benjamin W. Wah, Weijia Shang, and Kumar N. Ganapathy

Information as a Commodity: Assessment of Market Value

Abbe Mowshowitz

Contributors

Numbers in parentheses refer to the pages on which the authors' contributions begin.

B. Chandrasekaran (73) *Laboratory for AI Research, The Ohio State University, Columbus, Ohio 43210*

Jose A. B. Fortes (198) *School of Electrical Engineering, Purdue University, West Lafayette, Indiana 47907*

Kumar N. Ganapathy (198) *Coordinated Science Laboratory, University of Illinois, Urbana, Illinois 61801*

John M. Long (146) *2676 Manson Pike, Murfreesboro, Tennessee 37129*

Abbe Mowshowitz (248) *Department of Computer Science, The City College (CUNY), New York, New York 10031*

Günther Pernul (1) *Institute of Applied Computer Science, Department of Information Engineering, University of Vienna, A-1010 Vienna, Austria*

Weijia Shang (198) *Department of Computer Engineering, Santa Clara University, Santa Clara, California 95053*

Benjamin W. Wah (198) *Coordinated Science Laboratory, University of Illinois, Urbana, Illinois 61801*

This Page Intentionally Left Blank

Preface

The publication of Volume 38 of *Advances in Computers* continues the in-depth presentation of subjects of both current and continuing interest in computer and information science. Contributions have been solicited from highly respected experts in their fields who recognize the importance of writing substantial review and tutorial articles in their areas of expertise. *Advances in Computers* permits the publication of survey-type articles written from a relatively leisurely perspective. By virtue of the length of the chapters included, authors are able to treat their subjects both in depth and in breadth. The *Advances in Computers* series began in 1960 and now continues in its 35th year with this volume. During this period, in which we have witnessed great expansion and dynamic change in the computer and information fields, the series has played an important role in the development of computers and their applications. The continuation of the series over this lengthy period is a tribute to the reputations and capabilities of the authors who have contributed to it.

Included in Volume 38 are chapters on database security, causal processes, computer-based medical systems, parallel processing with linear arrays, and information treated as a commodity.

In the first chapter, Günther Pernul points out that the general concept of database security is very broad and embraces such areas as the moral and ethical issues imposed by public and society, legal issues in which laws are passed regulating the collection and disclosure of stored information, and more technical issues such as ways of protecting stored information from loss or unauthorized access, destruction, use, modification, or disclosure. He proposes models and techniques that provide a conceptual framework in the effort to counter the possible threats to database security. Emphasis is given to techniques primarily intended to assure a certain degree of confidentiality, integrity, and availability of the data. Privacy and related legal issues of database security are also discussed.

In the second chapter, B. Chandrasekaran states that cognitive agents that are organized to achieve goals in the world have three fundamental activities to perform, namely, making sense of the world, planning actions to achieve goals, and predicting consequences. He reviews over a decade of work on device understanding from a functional perspective. He believes that research on causal and functional representations is just beginning. In his chapter he describes a research agenda for the immediate future, discusses the logic of "understanding," and also discusses the phenomena of "reasoning."

In Chapter 3, John Long indicates that the notion of computer-based medical systems embraces the full range of computer systems—both hardware and software—that are designed and built for use in a medical environment. These include embedded computers (hardware and software) found in medical devices. He shows that many of the areas of medicine are changing due to the impact of computers. Computer-based medical systems are revolutionizing medicine and moving it into the information age. The pace is deliberate as is appropriate for an area that deals with human health. The potential for great benefits exists and many have already been accomplished. By the same token, the changes being brought about because of computers create new problems and exacerbate existing ones.

In the next chapter Fortes, Wah, Shang, and Ganapathy point out that applications of digital signal processing, scientific computing, digital communications, and control are characterized by repeated execution of a small number of computationally intensive operations. In order to meet performance requirements it is often necessary to dedicate hardware with parallel processing capabilities to these specialized operations. Processor arrays, due to their structural regularity and consequent suitability for VLSI implementation, are frequently used for this purpose. They then show that algorithm-specific parallel processing with linear processor arrays can be systematically achieved with the help of the techniques discussed. In particular, they are ideally suited to the algorithms described as affine recurrences or loop nests.

Abbe Mowshowitz in the final chapter considers that the evolution of the marketplace for information appears to be governed by impulses stemming from the displacement of information, knowledge, or skill from persons to artifacts. This process of displacement is an extension of the commoditization of labor, a process that began in earnest with the industrial revolution. The information commodity is to contemporary organizations what the labor commodity was to the pre-industrial workshop—a vehicle for the radical reorganization of production. Triggered by advances in computers and telecommunications, he believes that this displacement process is gaining momentum with the integration of these technologies. Computer-based communications networks will soon reach virtually every organization and person in the industrialized world. Such networks will stimulate an explosive growth in the production and use of information commodities, and support a global marketplace of gigantic proportions.

I am pleased to thank the contributors to this volume. They have given extensively to make this book an important and timely contribution to their profession. Despite the considerable time and effort required, they have recognized the importance of writing substantial review and tutorial contributions in their areas of expertise; their cooperation and assistance are greatly appreciated. Because of their efforts, this volume achieves a high level of excellence and should be of great value and substantial interest for many years to come. It has been a pleasant and rewarding experience for me to edit this volume and to work with the authors.

Marshall C. Yovits

This Page Intentionally Left Blank

Database Security

GÜNTHER PERNUL

Institute of Applied Computer Science
Department of Information Engineering
University of Vienna

1. Introduction

Information stored in databases is often considered a valuable and important corporate resource. Many organizations have become so dependent on the proper functioning of their systems that a disruption of service or a leakage of stored information may cause outcomes ranging from inconvenience to catastrophe. Corporate data may relate to financial records; may be essential to the successful operation of an organization, may represent trade secrets, or may describe information about persons whose privacy must be protected. Thus, the general concept of database

Copyright © 1994 by Academic Press, Inc.
All rights of reproduction in any form reserved.

ISBN 0-12-012138-7

security is very broad and embraces such areas as the moral and ethical issues imposed by public and society and legal issues in which laws are passed regulating the collection and disclosure of stored information, or more technical issues such as ways of protecting stored information from loss or unauthorized access, destruction, use, modification, or disclosure.

More generally, database security is concerned with ensuring the secrecy, integrity, and availability of data stored in a database. To define our terms, *secrecy* denotes the protection of information from unauthorized disclosure either by direct retrieval or indirect logical inference. In addition, secrecy must deal with the possibility that information may also be disclosed by legitimate users acting as an "information channel" by passing secret information to unauthorized users. This may be done intentionally or without the knowledge of the authorized user. By *integrity* we understand the need to protect data from malicious or accidental modification, including insertion of false data, contamination of data, and destruction of data. Integrity constraints are rules that define the correct states of a database and thus can protect the correctness of the database during operation. By *Availability* we understand the characteristic according to which we may be certain that data are available to authorized users when they need them. Availability includes the "denial of service" of a system, as occurs when a system is not functioning in accordance with its intended purpose. Availability is closely related to integrity because "denial of service" may be caused by unauthorized destruction, modification, or delay of service as well.

Database security cannot be seen as an isolated problem as it is influenced by the other components of a computerized system. The security requirements of a system are specified by means of a security policy that is then enforced by various security mechanisms. For databases, the security requirements can be classified in the following categories:

- *Identification, Authentication.* Usually, before gaining access to a database, each user has to identify himself to the computer system. Authentication is a way of verifying the identity of a user at log-on time. Most of the common authentication methods are passwords but more advanced techniques like badge readers, biometric recognition techniques, or signature analysis devices are also available.
- *Authorization, Access Controls.* Authorization consists in the specification of a set of rules that declare who has a particular type of access to a particular type of information. Authorization policies, therefore, govern the disclosure and modification of information. Access controls are procedures that are designed to control authorization by limiting access to stored data to authorized users only.

- *Integrity, Consistency.* An integrity policy gives a set of rules (i.e., semantic integrity constraints) that define the correct states of the database during database operation and, therefore, can protect against malicious or accidental modification of information. Closely related issues are concurrency control and recovery. Concurrency control policies protect the integrity of the database in the presence of concurrent transactions. If these transactions do not terminate normally due to system crashes or security violations, recovery techniques may be used to reconstruct correct or valid database states.
- *Auditing.* The requirement to keep records of all security-relevant actions issued by a user is called auditing. The resulting audit records are the basis for further reviews and examinations in order to test the adequacy of system controls and to recommend changes in a security policy.

In this chapter our approach will not involve this type of broad perspective of database security. Instead, the main focus will be on aspects of authorization and access controls. This is a legitimate concern, since identification, authentication, and auditing[1] normally fall within the scope of the underlying operating system and integrity and consistency policies are subject to the closely related topic of "semantic data modeling" or are dependent on the physical design of the database management system (DBMS) software, namely, the transaction and recovery manager. Because most research in database security has concentrated on the relational data model, the discussion in this chapter will focus on the framework of relational databases. However, the results described may generally be applicable to other database models as well. For an overall discussion on basic database security concepts consult the surveys by Jajodia and Sandhu (1990a), Lunt and Fernandez (1990), and Denning (1988). For references to further readings consult the annotated bibliography compiled by Pernul and Luef (1992).

In the remainder of the opening section we briefly review the relational data model, introducing a simple example that will be used throughout the chapter, present the basic terminology used in computer security, and describe the most successful methods of penetrating a database. Because of the diversity of application domains for databases different security models and techniques have been proposed so far. In Section 2 we review, evaluate, and compare the most prominent examples of these security models and techniques. Section 3 contains an investigation of secure (trusted) database management systems. By a secure DBMS we understand special-purpose

[1] However, audit records are often stored and examined by using DBMS software.

systems that support a level-based security policy and are designed and implemented with the main focus on the enforcement of high security requirements. Section 4 focuses on one of the major problems of level-based security-related database research. In this section we address the problem of classifying the data stored in a database so that the security classifications reflect the security requirements of the application domain proper. What is necessary here is to have a clear understanding of all the security semantics of the database application and an appropriate clever database design. A semantic data/security model is proposed in order to arrive at a conceptualization and clear understanding of the security semantics of the database application. Database security (and computer security in general) is subject to many national and international standardization efforts. These efforts are aimed at developing metrics for evaluating the degree of trust that can be placed in the computer products used in the processing of sensitive information. In Section 5 we briefly review these proposals. In Section 6 we point out research challenges in database security and attempt to forecast the direction of the field over the next few years. Section 7 concludes the chapter.

1.1 The Relational Data Model Revisited

The relational data model was invented by Codd (1970) and is described in most database textbooks. A relational database supports the relational data model and must have three basic components: a set of relations, a set of integrity rules, and a set of relational operators. Each relation consists of a state-invariant relational schema $RS(A_1, \ldots, A_n)$, where each A_i is called an attribute and is defined over a domain $dom(A_i)$. A relation R is a state-dependent instance of RS and consists of a set of distinct tuples of the form $(a_1, \ldots, a_n)$, where each element a_i must satisfy $dom(A_i)$ (i.e., $a_i \in dom(A_i)$).

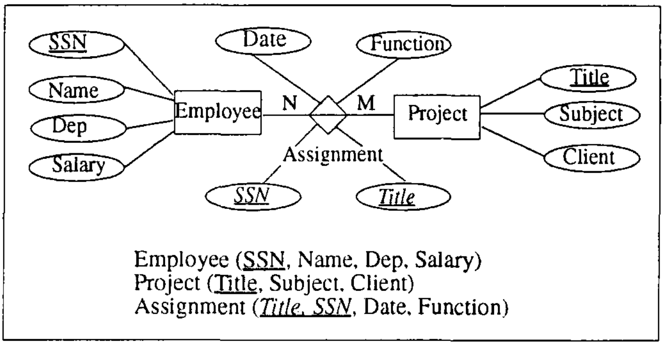
Integrity constraints restrict the set of theoretically possible tuples (i.e., $dom(A_1) \times dom(A_2) \times \cdots \times dom(A_n)$) to the set of practically meaningful tuples. Let X and Y denote sets of one or more of the attributes A_i in a relational schema. We say Y is *functionally dependent* on X, written $X \rightarrow Y$, if and only if it is not possible to have two tuples with the same value for X but different values for Y. Functional dependencies represent the basis of most integrity constraints in the relational model of data. Since not all possible relations are meaningful in an application, only those that satisfy certain integrity constraints are considered. From the large set of proposed integrity constraints two are of major relevance for security: the **key property** and the **referential integrity property**. The key property states that each tuple must be uniquely identified by a key and a key attribute must not have the null value. Consequently, each real-world event can be represented in the database only once. Referential integrity states that tuples referenced in one relation must exist in others and is expressed by means of foreign keys. These two rules are application-independent and must be valid in each relational database. In addition, many application-dependent semantic constraints may exist in different databases.

Virtual-view relations (or views) are distinguished from base relations. While the former are the result of relational operations and exist only virtually, the latter are actually present in the database and hold the stored data. Relational operations consist of the set operations, a select operation for selecting tuples from relations that satisfy a certain predicate, a project operation for projecting a relation onto a subset of its attributes, and a join operation for combining attributes and tuples from different relations.

The relational data model was first implemented as System R by IBM and as INGRES at U. C. Berkeley. The two projects provided the principal impetus for the field of database security research and also considerably advanced the field as well as forming the basis of most commercially available products.

A few words on the design of a database are in order. The design of a relational database is a complicated and difficult task and involves several phases and activities. Before the final relation schemas can be determined a careful requirements analysis and conceptualization of the database is necessary. Usually this is done using a conceptual data model powerful enough to allow the modeling of all application-relevant knowledge. The conceptual model is used as an intermediate representation of the database and ultimately transferred into corresponding relation schemas. It is very important to use a conceptual data model at this stage since it is only with such a high-level data model that a database can be created that properly represents all the application-dependent data semantics. The *de facto* standard for conceptual design is the Entity Relationship (ER) approach (Chen, 1976) or any one of its variants. In its graphical representation and in simplest form ER regards the world as consisting of a set of entity types (boxes), attributes (connected to the boxes), and relationship types (diamonds). Relationship types are defined between entity types and are either of degree $\langle 1:1 \rangle$, $\langle 1:n \rangle$, or $\langle n:m \rangle$. The degree describes the maximum number of participating entities.

Following is a short example of a relational database. This example will be used throughout the chapter. It is a very simple example yet sufficiently complex for presenting many of the security-relevant questions and demonstrating the complexity of the field. Figure 1 contains a conceptualization of the database in the form of an ER diagram and corresponding



Employee (SSN, Name, Dep, Salary)
Project (Title, Subject, Client)
Assignment (*Title*, *SSN*, Date, Function)

FIG. 1. Representations of a sample database.

relational schemas (key attributes are underlined, foreign keys are in italics). The database represents the fact that projects within an enterprise are carried out by employees. In this simple example there are three security objects. First, *Employee* represents a set of employees each of which is uniquely described by a characteristic SSN (Social Security Number). Next are Name (of employee), Department (in which the employee is working), and Salary (of employee). Second, *Project* refers to a set of projects carried out by the enterprise. Each project has an identifying Title, Subject, and Client. Finally, the security object *Assignment* contains the assignments of employees to projects. Each Assignment is characterized by the Date of the Assignment and the Function the employee has to perform while participating in the project. A single employee can be assigned to more than one project and a project may be carried out by more than one employee.

1.2 The Vocabulary of Security and Major Database Security Threats

Before presenting the details of database security research it is necessary to define the terminology used and the potential threats to database security. As we have already pointed out, security requirements are stated by means of a *security policy* which consists of a set of laws, rules, and practices that regulate how an organization manages, protects, and distributes sensitive information. In general, a security policy is stated in terms of a set of security objects and a set of security subjects. A *security object* is a passive entity that contains or receives information. It might be a structured concept like an entire database, a relation, a view, a tuple, an attribute, an attribute value, or even a real-world fact represented in the database.