Security-Control Methods for Statistical Databases: A Comparative Study

NABIL R. ADAM

Rutgers, The State University of New Jersey, Newark, New Jersey 07102

JOHN C. WORTMANN

Department of Industrial Engineering & Management Science, Eindhoven University of Technology, The Netherlands

This paper considers the problem of providing security to statistical databases against disclosure of confidential information. Security-control methods suggested in the literature are classified into four general approaches: conceptual, query restriction, data perturbation, and output perturbation.

Criteria for evaluating the performance of the various security-control methods are identified. Security-control methods that are based on each of the four approaches are discussed, together with their performance with respect to the identified evaluation criteria. A detailed comparative analysis of the most promising methods for protecting dynamic-online statistical databases is also presented.

To date no single security-control method prevents both exact and partial disclosures. There are, however, a few perturbation-based methods that prevent exact disclosure and enable the database administrator to exercise "statistical disclosure control." Some of these methods, however introduce bias into query responses or suffer from the 0/1 query-set-size problem (i.e., partial disclosure is possible in case of null query set or a query set of size 1).

We recommend directing future research efforts toward developing new methods that prevent exact disclosure and provide statistical-disclosure control, while at the same time do not suffer from the bias problem and the 0/1 query-set-size problem. Furthermore, efforts directed toward developing a bias-correction mechanism and solving the general problem of small query-set-size would help salvage a few of the current perturbation-based methods.

Categories and Subject Descriptors: H.2.0 [**Database Management**]: General—*security, integrity, and protection*

General Terms: Protection, Security

Additional Key Words and Phrases: Compromise, controls, disclosure, inference, security

INTRODUCTION

A database consists of a model of some part of the real world. Such a model is made up of entities (the elements of the part of the real world that is modeled), attributes (characteristics of the entities), and relationships among the different entities. Entities with identical attributes constitute a particular entity type. In a hospital

Supported by a grant from Rutgers GSM Research Resources Committee.

CONTENTS

database, for example, patients and treatments are system entities, and name, Social Security number, and diagnosis type are attributes of the entity type patient.

A statistical database (SDB) system is a database system that enables its users to retrieve only aggregate statistics (e.g., sample mean and count) for a subset of the entities represented in the database. An example is Ghosh's [1984, 1985] description of an SDB that is made up of test data for a manufacturing process. Another example is the database maintained by the U.S. Census Bureau. These examples are special-purpose databases, since providing aggregate statistics is their only purpose. In other situations, a single database may serve multiple applications, including a statistical application. A hospital database, for example, might be used by physicians to support their medical work as well as by statistical researchers of the National Health Council. In that case, the statistical researchers are authorized to retrieve only aggregate statistics; the physicians, on the other hand, can retrieve anything from the database.

Many government agencies, businesses, and nonprofit organizations need to collect, analyze, and report data about individuals in order to support their short-term and long-term planning activities. SDBs therefore contain confidential information such as income, credit ratings, type of disease, or test scores of individuals. Such data are typically stored online and analyzed using sophisticated database management systems (DBMs) and software packages. On the one hand, such database systems are expected to satisfy user requests of aggregate statistics related to nonconfidential and confidential attributes. On the other hand, the system should be secure enough to guard against a user's ability to infer any confidential information related to a specific individual represented in the database. The problem here is "the inevitable conflict between the individual's right to privacy and the society's need to know and process information" [Palley 1986; Palley and Simonoff 1987]. Dalenius [1974] presents an overview of this problem as does the following citation from Miller [1971, p. 136]:

> Some deficiencies inevitably crop up even in the Census Bureau. In 1963, for example, it reportedly provided the American Medical Association with a statistical list of one hundred and eight-eight doctors residing in Illinois. The list was broken down into more than two dozen categories, and each category was further subdivided by medical specialty and area residence; as a result, identification of individual doctors was possible . . . .

The recent proliferation of computerized-information systems has added to the growing public concern about threats to individuals' privacy. It is therefore not surprising that the problem of securing SDBs has become an important one in recent years. As we move further into the information age and see expert and knowledge-based systems used in conjunction with SDBs, the security problem is expected to become even more important. Trueblood [1984], for example, illustrates how knowledge-based and expert systems could use nonconfidential information to infer confidential information.

The following example of a hospital database discusses the problem of securing an SDB. The database contains these data about patients:

{Age, Sex, Employer, Social Security Number, Diagnosis Type}

In the hospital environment, physicians may be given access to patients' entire medical records, whereas statistical researchers may only be allowed to obtain aggregate statistics for subsets of the patient population. A subset of patients whose data are included in the computation of the response to a query is referred to as the *query set.* Statistics are calculated for subsets of patients having common attribute values (e.g., Age = 42 and Sex = male). Such a subset can be specified by a characteristic formula, C, which is a logical formula over the values of the attributes using the Boolean operators AND (&), OR (+), and NOT ($\neg$). For example,

$$C = (\text{Age} = 42) \ \& \ (\text{Sex} = \text{Male}) \ \& \ (\text{Employer} = \text{ABC})$$

is a characteristic formula that specifies the subset of male patients, age 42, employed by the ABC company. The size of the query set that corresponds to a characteristic formula C is denoted by $|C|$.

Suppose there is a malevolent researcher who wants to obtain information about the diagnosis type of a given patient, Mr. X. A malevolent user who wants to compromise the database is referred to as a *snooper.* In our example, assume that the snooper knows the age and employer of Mr. X. He can then issue the query

Q1: COUNT (Age = 42) & (Sex = Male) & (Employer = ABC).

If the answer is 1, the snooper has located Mr. X and can then issue such queries as

Q2: COUNT (Age = 42) & (Sex = Male) & (Employer = ABC) & (Diagnosis Type = Schizophrenia).

If the answer to Q2 is 1, the database is said to be positively compromised and the user is able to infer that Mr. X has the diagnosis type schizophrenia. If the answer is 0, the database is said to be partially compromised, because the user was able to infer that the diagnosis type of Mr. X is *not* schizophrenia. Partial compromise refers to the situation in which some inference about a confidential attribute of an entity can be made, even if the exact value cannot be determined. It may take the form of a negative compromise, that is, it is inferred that an attribute of a certain entity does not lie within a given range.

As we have seen from the above example, there are basically three types of authorized users: the nonstatistical user (e.g., the physician in the hospital example) who is authorized to issue queries and update the database, the researcher who is authorized to retrieve aggregate statistics from the database, and the snooper who is interested in compromising the database. There is also a database administrator (DBA) who, by definition, is the guardian of the database. In the remainder of this paper we will use the terms *user* and *researcher* interchangeably.

In this paper we assume that the following elements of access control [Turn and Shapiro 1978] have been implemented and are effective in preventing unauthorized access to the system:

- Authorization of people to access the computer facility and the database system.
- Identification of a person seeking access to the computer facility and the database system.
- Authentication of the user's identity and access authorization.

As such we will use the term *snooper* to refer to a person who is making improper use of the data normally available to him or her as an authorized user. Usually it is assumed that a snooper has supplementary (i.e., a priori) knowledge about a target. For example, in queries Q1 and Q2, the snooper knows a priori that the target is male, 42 years old, and employed by the ABC Company. Assumptions regarding the a priori knowledge of the snooper are crucial to the development of an effective security-control method: The more the security-control method is aware of each user's supplementary knowledge, the more effective it will be in reducing the likelihood of compromising the database.

An SDB that serves multiple applications can be structured as a hierarchical, network, or relational database. An SDB whose only purpose is to provide aggregate statistics can be structured in a tabular form. The relational and tabular forms of SDBs are widely discussed in the literature [Ghosh 1986].

In the relational form, each entity in the real world is represented by a tuple consisting of attribute values of that entity. A set of tuples with similar attributes constitutes a relation. Such a relation is usually depicted as a two-dimensional table, where the rows correspond to tuples and the columns correspond to attributes. In statistical analysis, the domain of attribute values is often subdivided into classes, which are used as categories. Based on these categories, all entities are classified into elementary cells. If two entities belong to the same categories for all attributes, they are in the same elementary cell. This process is referred to as microaggregation.

If M attributes are used for categorization, the data can be represented by an M-dimensional table consisting of elementary cells. This type of table representation is called the tabular form. The cells of such tables contain summary statistics (e.g., the number of entities) computed over the entities contained in the cell [Denning 1983]. Many statistical databases (e.g., census data) are published in tabular form.

An SDB in relational form is easily translated into an equivalent one in tabular form. The reverse, however, is not true; that is, it is not always possible to deduce the relational form from the tabular form. This is because there is information loss when attribute values of entities are categorized in order to obtain the tabular representation.

The remainder of this section discusses the problem of securing SDBs and gives a brief overview of the solution approaches. Then Section 1 identifies a set of criteria that can be used to evaluate the performance of a security-control method, and Sections 2 through 5 examine the security-control methods that have been proposed to date that are related to the solution approaches discussed in the Introduction. A detailed comparative analysis of the most promising security-control methods is given in Section 6, followed by a discussion of new types of threats that are starting to be explored in the literature in Section 7. Section 8 presents our conclusions.

The Security Problem of Statistical Databases

The objective of an SDB is to provide researchers with aggregate statistics (e.g., mean and count) about a collection of entities while protecting confidentiality of any individual entity represented in the database. It is the policy of the system as set by the DBA that determines the criterion for defining confidential information [Denning and Schlörer 1983].

Threats to data security arise from a snooper's attempt to infer some previously unknown, confidential data about a given entity. Threats to security may result in exact or partial disclosure. A disclosure is said to occur (or, equivalently, an SDB is said to have been compromised) if through the answer to one or more queries a snooper is then able to infer the exact value of (exact disclosure) or a more accurate estimate of a confidential attribute of an individual entity. In this paper, we use the terms *compromise* and *disclosure* interchangeably.

Overview of Solution Approaches

Several methods for protecting the security of SDBs have been suggested in the literature. These methods can be classified under

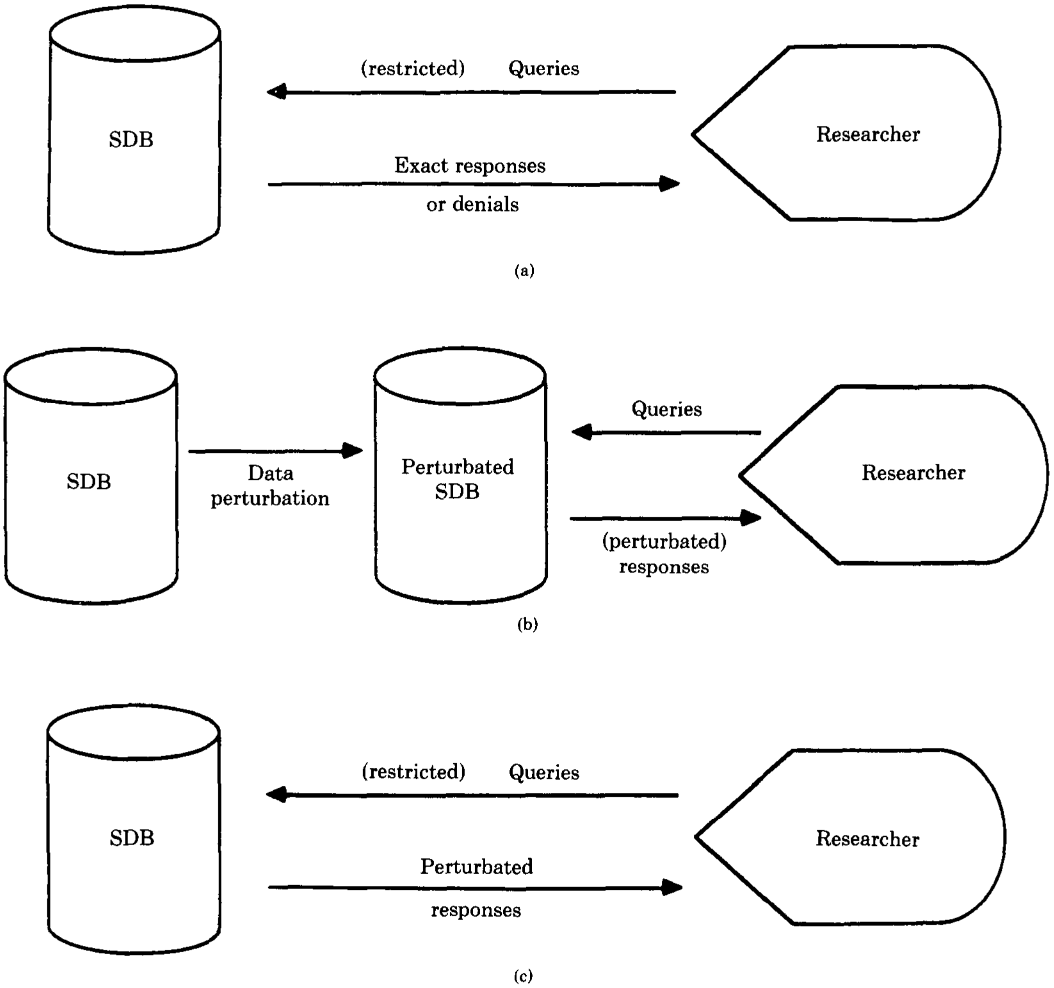

Figure 1. (a) Query set restriction; (b) data perturbation; (c) output perturbation.

four general approaches: conceptual, query restriction, data perturbation, and output perturbation.

Two models are based on the conceptual approach: the conceptual model [Chin and Özsoyoğlu 1981] and the lattice model [Denning 1983; Denning and Schlörer 1983]. The conceptual model provides a framework for investigating the security problem at the conceptual-data-model level. The lattice model constitutes a framework for data represented in tabular form. Each of these models presents a framework for better understanding and investigating the security problem of SDBs. Neither presents a specific implementation procedure. A discussion of these models is given in Section 2.

Security-control methods that are based on the query-restriction approach (see Figure 1a) provide protection through one of the following measures: restricting the query set size, controlling the overlap among successive queries by keeping an audit trail of all answered (or "could be deduced") queries for each user, making cells of "small size" unavailable to users of SDBs that are in tabular form, or partitioning the SDB. Section 3 includes a discussion of these methods.

Data perturbation introduces noise in the data. The original SDB is typically transformed into a modified (perturbed) SDB, which is then made available to researchers (see Figure 1b). Section 4 examines each of the data-perturbation-based methods.

The output-perturbation approach perturbs the answer to user queries while leaving the data in the SDB unchanged (see Figure 1c). A discussion of the output-perturbation-based methods is given in Section 5.

Types of Statistical Databases and Computer Systems

The nature of the database and the characteristics of the computer system strongly affect the complexity of the SDB security problem and the proposed solution approach. The following classification [Turn and Shapiro 1978] reflects the variety of environments.

Offline-Online

In an online SDB, there is direct real-time interaction of a user with the data through a terminal. In an offline SDB, the user neither is in control of data processing nor knows when his or her data request is processed. In this mode, protection methods that keep track of user profiles become more cumbersome. Compromise methods that require a large number of queries (e.g., regression-based-compromise method, see Section 7) also become more difficult when working offline.

Static-Dynamic

A static database is one that never changes after it has been created. Most census databases are static. Whenever a new version of the database is created, that new version is considered to be another static database. In contrast, dynamic databases can change continuously. This feature can complicate the security problem considerably, because frequent releases of new versions may enable snoopers to make use of the differences among the versions in ways that are difficult to foresee. Data-perturbation methods may not be suitable for dynamic SDBs since the efforts of transforming the original SDB to the perturbed one may become prohibitive.

Centralized-Decentralized

In a centralized SDB there is one database. In a decentralized (distributed) SDB, overlapping subsets of the database are stored at different sites that are connected by a communication network. A distributed database may be fully replicated, partially replicated, or partitioned. The security problem of a distributed SDB is more complex than that of a centralized one due to the need to duplicate, at each site, the security-control overhead, as well as the difficulty of integrating user profiles.

Dedicated-Shared Computer System

In a dedicated SDB, the computer system is used exclusively to serve SDB applications. In a shared system, the SDB applications run on the same hardware system with other applications (possibly using different databases). The shared environment is more difficult to protect, since other applications may be able to interfere with the protected data directly through the operating system, bypassing the SDB security mechanism.

The above discussion indicates that the degree of difficulty of securing an SDB against a snooper's attempt to infer some previously unknown, confidential data about an individual entity depends upon whether the SDB is online or offline, static or dynamic, centralized or decentralized, and running on a dedicated or shared computer system.

1. EVALUATION CRITERIA

This section discusses the criteria for evaluating the security-control methods investigated in Sections 2 through 5.

(1) Security is the level of protection provided by the control method against complete (or exact) and partial disclosure. In perturbation-based methods, partial disclosure is referred to as statistical disclosure. Beck [1980] suggests that a statistical

disclosure is said to occur if, using information from a series of queries, it is possible to obtain a better estimate for confidential information than was possible using only one query. The larger the number of queries required to compromise an SDB partially (or statistically), the more secure is the SDB.

In this paper, we adopted the following definition of compromise. Consider a confidential attribute A_i for an individual entity i represented in the database

$$A_i = \begin{cases} 1 & \text{the entity possesses a given property (e.g., disease type AIDS)} \\ 0 & \text{otherwise} \end{cases}$$

or

A_i is a numerical attribute (e.g., income)

An exact compromise is said to take place if by issuing one or more queries, a user is able to determine that $A_i = 1$ or its exact value (if it is a numerical attribute). A partial compromise occurs if by issuing one or more queries, a user is able to determine that $A_i = 0$ or to obtain, for the case of numerical attribute, an estimator $\hat{A}_i$ whose variance satisfies the following:

$$\mathrm{Var}(\hat{A}_i) < c_1^2,$$

where c_1 is a parameter that is set by the DBA.

In this paper, we consider a security-control method to be acceptable if it prevents exact disclosure and results in statistical-disclosure control. The term *statistical-disclosure control*, introduced in Dalenius [1977], refers to the ability of a system to provide users with a point estimate of the desired statistic and to require a "large" number of independent samples for obtaining a "small" variance of the estimator. What constitutes a large number of queries and a small variance depends on the sensitivity of the data involved. The important point here is that the DBA is able to set the system parameters according to what he or she considers a large number of queries and a small variance.

(2) Robustness of a given method is concerned with such assumptions as regarding the supplementary knowledge of a snooper. "A user's supplementary knowledge is a set of all the information about the database which a user knows from a source other than the system" [Haq 1977]. In dynamic SDBs especially, user knowledge with respect to inserting and deleting entities as well as updating attributes has to be considered.

(3) Suitability to numerical and/or categorical attributes needs to be considered. It is desirable to have a method that can be applied to control the security of confidential numerical as well as categorical attributes.

(4) Suitability to more than one attribute is necessary since a typical real-world application involves several attributes. Those methods that have been designed to deal with only one confidential attribute (numerical or categorical) are clearly restrictive.

(5) Suitability to dynamic SDBs is also necessary. In the types of environments we are discussing it is assumed that only the DBA and possibly a few other users are authorized to update the SDB; for them, the SDB is just a regular database. The rest of the users (researchers) must be provided with statistics that reflect the dynamics of the real world. Hence, for a security-control method to be suitable for an online dynamic SDB, it ensures that any changes to the SDB are reflected in the statistics provided to users as soon as the changes have taken place in the real world.

(6) Richness of information revealed to users is determined by the amount of nonconfidential information that is unnecessarily eliminated as well as, in case of perturbation-based methods, the statistical quality of the information provided to users. An ideal security-control method should provide users with all relevant nonconfidential information and at the same time protect all confidential information. For tabular SDBs, a measure of information loss is as follows [Özsoyoğlu and Chung 1986]:

$$\frac{100 * |\mathrm{SC}|}{\mathrm{TC}},$$

where | SC | is the number of suppressed cells and TC is the total number of cells. This formula may overestimate the amount of nonconfidential information eliminated. Notice that it is natural in the context of the lattice model, where there is no data manipulation language and all the information about the model can be characterized by the cells of the lattice.

Bias, precision, and consistency are three components of the statistical quality of the information revealed to users. Bias represents the difference between the unperturbed statistic and the expected value of its perturbed estimate [Denning and Schlörer 1983]. In general, the property of being unbiased is one of the more desired ones in point estimation. We will, therefore, assume that being unbiased is a desirable property of any security-control method.

Precision refers to the variance of the estimators obtained by users. On the one hand, we would like to provide users with as precise information as possible, that is, with an estimator with as low a variance as possible. On the other hand, we would like to ensure that an estimator obtained by a snooper would have as high a variance as possible. Therefore, an effective security-control method is one that enables the DBA to adjust the precision to an appropriate value by setting the method's parameters accordingly. It is also desirable to provide users with a confidence interval of the estimated statistic. As will be shown in the next sections, however, this requirement gives a snooper additional information that may enable him or her to find an easy way to compromise the database.

Consistency represents the lack of contradictions and paradoxes [Denning and Schlörer 1983]. Contradiction arises when, for example, different responses are obtained to repetitions of the same query or the average statistic differs from the computed average using the sum and count statistics. (A difference in answers to repetitions of the same queries that is due to changes in the real world is not, however, considered an inconsistency.) A negative response to a count query is an example of paradox. Consistency is a desirable feature of any security-control method.

(7) Cost is made up of three components. The first is the implementation cost, which represents the effort required by the DBA to implement the security-control method and to determine the required parameters of that method. The second component, processing overhead per query, measures the CPU time and storage requirements of the method during query processing. In an online-dynamic-SDB environment, the processing component of the cost is a significant factor. The third component is the amount of education required to enable users to understand the security-control method so that they can make effective use of the SDB.

2. CONCEPTUAL APPROACH

2.1 The Conceptual Model

In Chin and Özsoyoğlu [1981] and Özsoyoğlu and Chin [1982] a framework is described for dealing with the security problem from the development of the conceptual schema to implementation. Background on this framework can be found in Chin's earlier work [1978]. In the conceptual-modeling approach, the population (a population is a collection of entities that have common attributes) and its statistics are all that users can access. There is no data manipulation language (such as relational algebra) allowed to merge and intersect populations.

The framework, which has been analyzed in Özsoyoğlu and Chung [1986] and Özsoyoğlu and Su [1985], deals with several issues that have not been considered jointly elsewhere in the literature. The most important of these issues are as follows:

(1) An SDB is more than just one file that has data about one population. An SDB contains information about different types of populations (e.g., patients, treatments, medicines, doctors). Moreover, the security control components of the SDB management system should be made aware of the relevant subcategories of each population and the security constraints with respect to these subcategories that must be enforced.

(2) The dynamics of the database (including the dynamics of its structure and security constraints) should be taken into account. In particular, disclosure due to these dynamics should be prevented.

(3) Users' supplementary knowledge should be maintained and kept up to date. The security-control method could take such information into consideration when responding to a user query.

(4) Possible inferences that may lead to disclosure, with or without users' supplementary knowledge, about confidential information should be analyzed.

The framework relies on the distinction between the conceptual-data-modeling level and the internal level. The framework applies mainly to the conceptual level. The framework, however, contains a certified main kernel, which deals with all access to the confidential part of the physical database. Within this framework, the security is basically guaranteed by the introduction of the concept of "smallest nondecomposable subpopulations" (referred to as atomic populations or *A*-populations). These *A*-populations always contain either zero or at least two entities, thus preventing disclosure of information about one individual entity. The *A*-populations correspond to elementary cells in an SDB, which are represented in tabular form (see the discussion on the lattice model below). In the conceptual-modeling framework, however, the definition of the *A*-populations is controlled by the database designer (usually the DBA). Preventing *A*-populations from having only one entity due to insertions and deletions is controlled by either delayed update processing or by dummy entities. As Schlörer [1983] notes, adding dummy entities may introduce bias in reported statistics.

The framework describes an architecture, called the Statistical Security Management Facility, for realizing statistical security in SDB environments. The parts of this facility are discussed in Sections 2.1.1 to 2.1.5.

2.1.1 Population Definition Construct

Each subpopulation has a corresponding population definition construct that keeps track of information such as the allowed statistical query types for each attribute of the population, the security constraints enforced in updates, and the history of changes in the (sub)population.

2.1.2 User Knowledge Construct

User Knowledge Construct is a process that keeps track of the properties of each user group. These properties describe the group's knowledge from earlier queries as well as any supplementary knowledge it may have (e.g., knowledge about confidential attribute values of individual entities). It is worthwhile to note that several of the methods discussed in the next section, that is, query-overlap control and auditing, propose some kind of user knowledge construct that is not as elaborate as the one proposed here.

2.1.3 Constraint Enforcer and Checker

Constraint enforcer and checker is the process that enforces security constraints when queries are issued to the system. It provides the user knowledge construct with information on successfully answered queries (and, therefore, increased knowledge of a user group). The constraint enforcer and checker is invoked whenever the DBA needs to study changes in the security constraints. It is also invoked whenever users try to perform inserts, updates, or deletions of entities in the database. The DBA is informed by the constraint enforcer and checker about the security consequences of such transactions.

2.1.4 Conceptual Model Modification

Conceptual model modification is a process that supports all changes to the conceptual model. These changes are first checked in a test mode, where conceptual model modification communicates with the constraint enforcer and checker and reports security consequences of the proposed changes to the DBA.

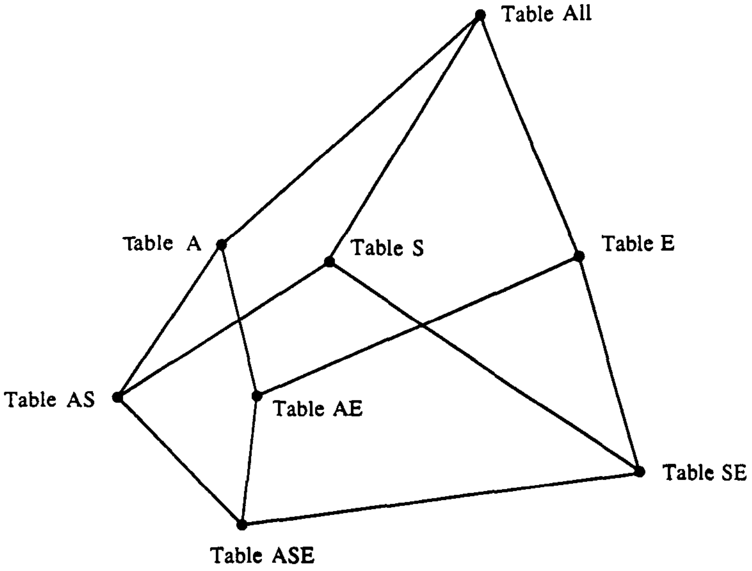

Figure 2. Lattice model.

2.1.5 Question and Answering System

The question and answering system is the primary tool by which the DBA communicates with all other processes. It is invoked whenever a process needs to bring some facts to the attention of the DBA (e.g., disclosure possibility, insertions, updates, deletions) or when the DBA initiates questions to the system.

The conceptual-modeling framework is an ambitious endeavor. To our knowledge, the framework has not been realized as a complete working software system. Also, note that considerable overhead may result from implementing such a system. Nevertheless, the framework has broadened the scope of research in security of statistical databases.

The conceptual-modeling framework has turned out to be a useful vehicle for further research. Update handling techniques within this framework have been studied in Özsoyoğlu and Özsoyoğlu [1981]. They conclude that output perturbation by means of rounding is convenient in reducing the probability of compromise in the case of single confidential attribute. Their investigation is concerned with a single population in the generalization hierarchy. Özsoyoğlu and Su [1985] extended that work in a study on the rounding method for a tree-organized generalization hierarchy, still within the conceptual-modeling approach.

2.2 The Lattice Model

A second general framework that has successfully supported the research in the security of SDBs is the lattice model [Denning 1983; Denning and Schlörer 1983]. This model can be viewed as a generalization of the one described in Kam and Ullman [1977]. The lattice model describes SDB information in tabular form at different levels of aggregation. The interest in it stems from the fact that statistical information that is provided at different levels of aggregation may introduce redundant information. If confidential information is suppressed at the detailed level (e.g., by suppressing cells with single entries, as with the A-populations [Chin and Özsoyoğlu 1981]), such information might be disclosed due to more aggregate information.

To illustrate, consider a hospital database with categorical attributes Age, Sex, and Employer (A, S, E). The corresponding lattice model is shown in Figure 2. The most detailed way to represent this database in tabular form consists of a

		AGE			
TABLE *ASE*		0–20	21–45	46–65	>65
EMPLOYER					
UNEMPLOYED	M	24	2	9	49
	F	26	0	1	51
ABC-COMPANY	M	0	1	9	0
	F	0	16	0	0
XYZ-INC.	M	1	20	48	0
	F	1	0	52	0

		AGE			
TABLE *AS*		0–20	21–45	46–65	>65
SEX	M	25	23	66	49
	F	27	16	53	51

	AGE			
TABLE *AE*	0–20	21–45	46–65	>65
EMPLOYER				
UNEMPLOYED	50	2	10	100
ABC-COMPANY	0	17	9	0
XYZ-INC.	2	20	100	0

	SEX	
TABLE *SE*	M	F
EMPLOYER		
UNEMPLOYED	84	78
ABC-COMPANY	10	16
XYZ-INC.	69	53

TABLE *ALL*:

$\boxed{310}$

	AGE			
TABLE *A*	0–20	21–45	46–65	>65
	52	39	119	100

	SEX	
	M	F
TABLE *S*	163	147

	EMPLOYER		
	UNEMPLOYED	ABC-COMPANY	XYZ-INC.
TABLE *E*	162	26	122

Figure 3. Lattice model of Figure 2 (extension).

three-dimensional table *ASE* with dimensions *A*, *S*, and *E* (Figure 3). A particular elementary cell in this table might be, for example, the cell, where $A = 42$, $S = \text{M}$, and $E = \text{ABC}$ company. This tabular form can be aggregated into three two-dimensional tables:

(1) Table *AS*, where *ASE* is aggregated over the dimension *E*; an example of a cell in this case is the one in which $A = 42$ and $S = \text{M}$.

(2) Table *AE*, where *ASE* is aggregated over the dimension *S*; an example of a cell in this case is the one in which $A = 42$ and $E =$ ABC company.

(3) Table *SE*, where *ASE* is aggregated over the dimension *A*; an example of a cell in this case is the one in which $S =$ M and $E =$ ABC company.

The aggregation performed in obtaining these three new tables is called microaggregation. The process can be repeated in order to obtain three one-dimensional tables:

(1) Table *A*, where table *AS* is aggregated over the dimension *S* or table *AE* is aggregated over the dimension *E*.

(2) Table *S*, where table *AS* is aggregated over the dimension *A* or Table *SE* is aggregated over the dimension *E*.

(3) Table *E*, where table *AE* is aggregated over the dimension *A* or Table *SE* is aggregated over the dimension *S*.

The aggregation can be extended one step further, where a zero-dimensional table is obtained. This table contains only one cell, providing statistics for the database as a whole. Note that the set of all two-dimensional tables may sometimes disclose the elementary cell statistic of the three-dimensional table.

The relationship between tabular data and SDBs can be defined more formally as follows: Let the set of attributes of an SDB be $\{A_1, \ldots, A_M\}$. Suppose that for each attribute A_i, a finite domain of allowable values, a_{ij_i} is given ($j_i = 1, \ldots, |A_i|$). Schlörer [1983] defines an *m*-set as a query set that can be specified using *m* (but not fewer) attributes. An elementary *m*-set is characterized by a formula of the form

$$(A_1 = a_{ij_1}) \\ \& \ (A_2 = a_{2j_2}) \ \& \ \cdots \ \& \ (A_m = a_{mj_m}).$$

These elementary *m*-sets correspond to the cells in tabular data with *m* dimensions (an *m*-table).

Although in an actual dynamic SDB data will not always be structured according to such tables, elementary *m*-sets provide an interesting concept for systematically studying the approaches to query restrictions. For example, it is shown in Denning and Schlörer [1983] that all *m*-tables for a given statistic (e.g., COUNT or SUM) constitute a lattice when *m* ranges from 1 to *M*.

3. QUERY RESTRICTION APPROACH

Five general methods have been developed to restrict queries: query-set-size control, query-set-overlap control, auditing, cell suppression, and partitioning. Following is a discussion of these five methods, together with an evaluation of their performance with respect to the criteria discussed in Section 1. The results are summarized in Table 1. (See also Denning [1982, Chapter 6] for an excellent discussion of several of these methods and Denning and Schlörer [1983] for additional results.)

3.1 Query-Set-Size Control

The query-set-size control method permits a statistic to be released only if the size of the query set $|C|$ (i.e., the number of entities included in the response to the query) satisfies the condition [Fellegi 1972; Friedman and Hoffman, 1980; Hoffman and Miller 1970; Schlörer 1975]:

$$K \leq |C| \leq L - K,$$

where *L* is the size of the database (the number of entities represented in the database) and *K* is a parameter set by the DBA. *K* should satisfy the condition

$$0 \leq K \leq \frac{L}{2}.$$

It was shown that by using a snooping tool called "tracker" it is possible to compromise the database even for a value of *K* that is close to $L/2$ [Denning et al. 1979; Denning and Schlörer 1980; Jonge 1983; Schlörer 1980; Schwartz et al. 1979]. Notice that *K* cannot exceed $L/2$, otherwise no statistics would ever be released.

To illustrate the basic idea of a tracker, consider the following queries where the female set is used as a tracker:

Q3: $q(C) =$ COUNT (Sex = Female)

Q4: $q(C) =$ COUNT (Sex = Female + (Age = 42 & Sex = Male & Employer = ABC))

Q5: $q(C) = \text{COUNT}$ (Sex = Female + (Age = 42 & Sex = Male & Employer = ABC & Diagnosis Type = Schizophrenia))

Suppose that the response to Q3 and Q4 are, respectively, A and B, where $K \leq A \leq L - K$ and $K \leq B \leq L - K$. If $B = A + 1$, then the target is uniquely identified by the clause "Age = 42 & Sex = Male and Employer = ABC." In this case, the database is positively compromised if the response to Q5 is B and negatively compromised if the response to Q5 is A. It is usually easy to find a tracker for a characteristic formula C [Schlörer 1980].

A summary of the performance of the query-set-size-control method with respect to the evaluation criteria is given in Table 1. In general, there seems to be a consensus in the literature that subverting the query-set-size-control method is "straight forward and cheap" [Denning 1982; Traub et al. 1984].

3.2 Query-Set-Overlap Control

Notice that Q3 and Q4 have a large number of entities in common. Dobkin et al. [1979] noticed that many compromises use query sets that have a large number of overlapping entities. They studied the possibility of restricting the number of overlapping entities among successive queries of a given user. If K denotes the minimum query set size and r denotes the maximum number of overlapping entities allowed between pairs of queries, then according to Dobkin et al., the number of queries needed for a compromise has a lower bound of $1 + (K - 1)/r$. Unfortunately, however, for typical values of K/r, the lower bound of the number of queries needed to compromise the database is not a practical hindrance.

In practice, query-set-overlap-control method suffers from drawbacks such as [Dobkin et al. 1979]: (1) this control mechanism is ineffective for preventing the cooperation of several users to compromise the database, (2) statistics for both a set and its subset (e.g., all patients and all patients undergoing a given treatment) cannot be released, thus limiting the usefulness of the database, and (3) for each user, a user profile has to be kept up to date. The performance of the query-set-overlap method with respect to the evaluation criteria is summarized in Table 1. In regard to the cost criterion, the following comments are in order. The initial implementation effort consists of developing software that maintains user profiles and compares a new query set with all previous sets. We feel that such an effort is moderate. The processing overhead per query may, however, be very high due to the comparison algorithm. Every new query issued by a given user has to be compared with his or her previously issued ones. Each comparison takes $O(L)$ processing time, where L is the size of the SDB.

3.3 Auditing

Auditing of an SDB involves keeping up-to-date logs of all queries made by each user (not the data involved) and constantly checking for possible compromise whenever a new query is issued [Hoffman 1977; Schlörer 1976]. Auditing has advantages such as allowing the SDB to provide users with unperturbed response, provided that the response will not result in a compromise [Chin et al. 1984]. One of the major drawbacks of auditing, however, is its excessive CPU time and storage requirements to store and process the accumulated logs.

Chin and Özsoyoğlu [1982] developed a CPU time and storage-efficient method (called audit expert) that controls disclosure of a confidential attribute when using the SUM query. Consider a response, d, to a SUM query. Such a response provides the user with information in the form of a linear equation:

$$\sum_{i=1}^{L} a_i x_i = d,$$

where L is the number of entities represented in the SDB, a_i is one if the ith entity belongs to the query set and is zero otherwise, and x_i represents the value of a confidential numerical attribute for entity i.

The user's knowledge obtained by querying the SDB may, therefore, be described in the form of a set of linear equations obtained from linear combinations of equations in the set of answered queries. The audit expert maintains a binary matrix

Table 1. A Summary of the Performance

Security Control Method	Security		Robustness	Suitable for Numerical or Categ. Attribute	Suitable for One or More Attributes	Suitability to Online Dynamic SDB
	Exact Disclosure Possible?	Partial Disclosure Possible?				
Query Set Size Control	Yes	Yes	Low	Both	More than one	Moderate
Query Set Overlap Control	Yes (unless number of queries severely restricted)	Yes	Low	Both	More than one	Moderate
Auditing	No	Yes	Low	Both	One, otherwise processing overhead is very high	Low
Partitioning	No	Yes, but more protection results from larger min. size of A-population	Controlled by size of A-population	Both	More than one	Yes
Cell Suppression	No	Yes	Low	Both	More than one	No

whose columns represent specific linear combinations of database entities and whose rows represent the user queries that have already been answered. These rows are chosen in such a way that they describe exactly and efficiently the knowledge space of each user. When a new query is issued, the matrix is updated. A row with all zeros except for an ith column indicates that exact disclosure of the confidential attribute of the corresponding entity is possible. Thus, the answer to the new query should be denied. It was shown that it takes the audit expert no more than $O(L^2)$ time to process a new query [Chin and Özsoyoğlu 1982]. Hence, the method is suited for small SDBs.

The auditing method has been further investigated in Chin et al. [1984] for a special type of SUM query, the range SUM query, which is defined as

$$\sum_{i=1}^{L} a_i x_i = d,$$

where a_i is one if $LB \leq i \leq UB$ and is zero otherwise, and x_i, as before, represents the value of a confidential-numerical attribute for entity i. Chin et al. [1984] show that when using the proper data structure, the complexity of checking if a new range SUM query could be answered can be reduced to $O(L)$ time and space as long as the number of queries is less than L or to $O(t \log L)$ time and $O(L^2)$ space for the tth new range SUM query with $t \geq L$.

Table 1 includes a summary of the performance of the auditing method with respect to the evaluation criteria. We notice that the audit expert that has been developed only for SUM queries does not provide protection against partial disclosure. Although the DBA can provide the audit expert with information regarding additional user knowledge, the robustness of the method is considered very low since statistics on subpopulations with only a few entities will be made available to users. With respect to the cost criterion, the initial implementation effort is high because complex algorithms have to be implemented. For example, Chin and Özsoyoğlu [1982] show that maximizing the amount of nonconfidential information that is to be of the Query Restriction Based Methods

Richness of Information				Costs		
Amount of Nonconf. Info. Eliminated	Bias	Precision	Consistency	Initial Implemenation Efforts	Processing Overhead Per Query	User Education
High	NA	NA	NA	Low	Low	Very low
Very high	NA	NA	NA	Moderate	Very high for large SDBs	Very low
Moderate	NA	NA	NA	High	Very high for large SDBs	Low
Moderate (very high for sparse SDBs)	Yes, if dummy entities are added	NA	NA	Moderate for static SDB; very high for dynamic SDB	Very low for static SDB	Low
Moderate	NA	NA	NA	High	None	None

provided to users is an NP-complete problem (i.e. there exists no polynomial–time algorithm for solving this problem).

Two observations are worth noting. First, the security level provided for the SUM query by the audit expert may in actuality be better than assumed. This is due to the fact that for snoopers to be able to perform a linear system attack on a SUM query they must have enough supplementary knowledge about the database entities to enable them to identify, through the characteristic formulas of the successive queries, controlled groups of entities [Denning 1983]. To illustrate, consider the following four queries:

$$x_1 + x_2 = d_1,$$

$$x_3 + x_4 = d_2,$$

$$x_1 + x_3 + x_5 = d_3,$$

$$x_2 + x_4 + x_5 = d_4.$$

Given the query responses d_1, d_2, d_3, and d_4, snoopers can infer x_5 as follows:

$$x_5 = \frac{d_3 + d_4 - d_1 - d_2}{2}$$

As noted in Denning [1983], however, in order for snoopers to perform such an attack, they must have enough supplementary knowledge about the database entities so that they know exactly the coefficients of x_1 through x_5 in the query responses. In most practical applications, it is rare that a user would have such supplementary knowledge.

The second observation is related to the method proposed in McLeish [1983]. This method can be classified as a variant of auditing. It is based on the model used in Kam and Ullman [1977], which views an SDB as a function f from strings of k bits to the positive and negative integers with the keys being the domain of f. A query is always of length k bits; for example, for $k = 5$ a possible query could be 1**0*, with s 0's and 1's (in this case $s = 2$) and the * standing for "do not care." The result of a query Q that is of length k and has s 0's and 1's is given by

$$\sum_{\text{key } i \text{ matches } Q} f(i).$$

In the hospital database, for example, the key could consist of 17 bits xxxywwwwwwwzzzzzz as follows:

xxx is a code for the clinic,
y is a code for the patient's sex (0 = Male, 1 = Female),
wwwwwww is a code for the patient's age,
zzzzzz is a code for the type of disease.

Thus, the query ***0101010111111 would represent the sum of all male patients, independent of which clinic they are in, of age 42 who have a disease type 111111.

According to McLeish [1983], the amount of information gained by issuing a query Q is given by:

$$\begin{cases} \log\left(\dfrac{L}{\min(|C|,\ L-|C|)}\right) & \text{if } |C| \le L \\ & \text{or } |C| \le 0 \\ \log L & \text{if } |C| = L \\ 0 & \text{otherwise} \end{cases}$$

where L and $|C|$ are the database size and the query-set size, respectively.

McLeish [1983] argues that minimizing this information function corresponds to increasing the chance of compromising the database. Given a query of length k bits issued to an SDB of 2^k entities, it is shown that the expected value of the information gained by issuing such query is

$$k-(k-1)\left(\frac{1}{k}\right)^{1/k-1}$$

and is minimized when

$$p=\left(\frac{1}{k}\right)^{1/k-1} \quad \text{for } k>1 \qquad (1)$$

where p is the probability of an * occurring in any given bit position.

Based on the above results, the security-control method suggested in McLeish [1983] can be summarized as follows:

Keep audit trails of the sequence of queries in the following ways:

- Observe the actual value of p for that sequence of queries and determine if it is statistically significantly close to the minimum value given by (1). If so, there is a high likelihood that the user is attempting to compromise the database.
- Evaluate the information function for each query in the sequence and study statistically the deviation of this value from the minimum expected value given by (1). Based on these deviations, determine the likelihood that the user is attempting to compromise the database.

Since the study is preliminary in nature, no implementation details such as the computational time and storage requirements have been addressed.

In general, the applicability of the auditing method to real world situations is questionable since it is not feasible to account for disclosure "by collusion" that involves several users. Furthermore, unless we are concerned with a centralized and dedicated SDB with few users and one confidential attribute, the CPU time and storage requirements would render the method impractical. Despite these shortcomings, we believe it is too early to eliminate auditing completely from consideration.

3.4 Partitioning

The basic idea of partitioning is to cluster individual entities of the population in a number of mutually exclusive subsets, called atomic populations [Chin and Özsoyoğlu 1979, 1981; Schlörer 1983; Yu and Chin 1977]. The statistical properties of these atomic populations constitute the raw materials available to the database users. These authors pay special attention to the disclosure risk due to the dynamics of the SDB: If a snooper has additional knowledge of entity insertions, updates, and deletions, many new avenues of attack emerge under nearly all query-restricting methods. Partitioning could be an attractive technique to overcome this problem.

As long as atomic populations do not contain precisely one individual entity, a high level of security and precision can be attained. Partitioning is illustrated in Figure 4. As can be seen, the cells with size 1 have been eliminated by combining them with neighboring cells of different sex. In terms of the lattice model (Figure 2) we are, in this way, preserving the two-dimensional table AE (Age–Employer). The tables AS and SE shown in Figure 2 are no longer completely available. The choice to preserve the two-dimensional table AE (and consequently the one-dimensional tables E and A) is arbitrary. It is not always

TABLE *ASE*		AGE			
		0–20	21–45	46–65	>65
EMPLOYER					
UNEMPLOYED	M	24	2	**10**	49
	F	26	0		51
ABC-COMPANY	M	0	**17**	9	0
	F	0		0	0
XYZ-INC.	M	**2**	20	**48**	0
	F		0	**52**	0

TABLE *AS*		AGE			
		0–20	21–45	46–65	>65
SEX	M	X	X	X	49
	F	X	X	X	51

TABLE *AE*	AGE			
	0–20	21–45	46–65	>65
EMPLOYER				
UNEMPLOYED	50	2	10	100
ABC-COMPANY	0	17	9	0
XYZ-INC.	2	20	100	0

TABLE *SE*	SEX	
	M	F
EMPLOYER	X	X
UNEMPLOYED	X	X
ABC-COMPANY	X	X
XYZ-INC.		

TABLE *ALL*:

$\boxed{310}$

	AGE			
	0–20	21–45	46–65	>65
TABLE *A*	52	39	119	100

	SEX	
	M	F
TABLE *S*	X	X

	EMPLOYER		
	UNEMPLOYED	ABC	XYZ
TABLE *E*	162	26	122

Figure 4. Protection by partitioning on the model of Figure 3.

possible to preserve at least one ($m - 1$)-dimensional table if cells of m-dimensional tables are combined. For example, if the lady employed by XYZ, Inc., who is younger than 21 was not included in the database, the cell that contains the man employed by XYZ, Inc., who is younger than 21 would be combined with another cell, thus restricting some entry in table *AE*.

Schlörer [1983] has investigated a large number of practical databases and found that a considerable number of atomic populations with only one entity will emerge. Clustering such populations with larger ones leads to serious information loss [Schlörer 1983].

In order to cope with the problem of A-populations of size 1, it was proposed in Chin and Özsoyoğlu [1979, 1981], to add dummy entities to the database. Including dummy entities, however, introduces bias into statistics such as the Average [Schlörer 1983]. In the same study, Chin and Özsoyoğlu [1981] also proposed to postpone the processing of insert and delete transactions until there are two or more such transactions per atomic populations. Such a mode of operation is not well suited to a dynamic-SDB environment in which an update to the SDB should immediately be reflected in the information provided to users. Extensive study of these problems is still required before wide-scale application of partitioning is feasible [Schlörer 1983].

The performance of partitioning with respect to the evaluation criteria is summarized in Table 1. We add the following comments. In regard to partial disclosure, the original papers [Chin and Özsoyoğlu 1981; Özsoyoğlu and Özsoyoğlu 1981; Schlörer 1983] usually assume that the minimum size of the nonempty atomic populations equals 2. Atomic populations of small size make partial disclosure more likely. Small atomic-population sizes could lead to robustness problems too. The general idea of partitioning, however, is also applicable to atomic populations of some minimum size of 4, 5, . . . , and so on, in which case partial disclosure can be controlled to some extent.

The amount of nonconfidential information eliminated is moderate. For sparse databases, it could become considerable (information loss > 50%). See Özsoyoğlu and Chung [1986].

As far as cost of partitioning is concerned, there are several possibilities. If we are dealing with a static SDB that is already in tabular form, software modules have to be developed that (1) detect sensitive cells and create A-populations and (2) preprocess queries in order to detect violation of the partitioning rules. This seems to be a moderate initial effort. The processing overhead per query is very low. The situation becomes slightly more complicated for a static SDB in relational form. If feasible, a proper way to implement partititioning is to transform the SDB to a corresponding one of tabular form. It could be argued that such a transformation introduces some loss of information due to the fact that the relational model allows queries such as

$$\begin{aligned}\text{COUNT (Sex} &= \text{Female + (Sex}\\ &= \text{Male \& Age}\\ &= \text{42) \& Employer}\\ &= \text{ABC \& Diagnosis Type}\\ &= \text{Schizophrenia)}\end{aligned}$$

The additional richness of queries in the relational model, however, always stems from the fact that A-populations are divided, which according to partitioning should be disallowed.

For dynamic SDBs in relational form, A-populations have to be defined as separate entity types. Here, several notions from the conceptual-modeling approach described in Section 1 will be required. Therefore, the initial investment in software modules will be considerable. The processing overhead per query could still be minor if each new version of the dynamic SDB is brought into tabular form. Otherwise, depending upon the way in which the query language and the SDB are implemented, the processing overhead per query could become very high.

Finally, the cost of user education involves the publication of the set of A-populations to the user community; otherwise, users are unaware of queries that should be denied.

3.5 Cell Suppression

Cell suppression [Cox 1980; Sande 1983] is one of the techniques typically used by census bureaus for data published in tabular form. Cell suppression has been investigated for static SDBs. The basic idea is to suppress from the released table(s) all cells that might cause confidential information to be disclosed. Other cells of nonconfidential information that might lead to a disclosure of some confidential information also have to be suppressed (this is called complementary suppression). The

		AGE			
TABLE *ASE*		0–20	21–45	46–65	>65
EMPLOYER					
UNEMPLOYED	M	X	X	X	49
	F	X	X	X	51
ABC-COMPANY	M	0	X	X	0
	F	0	X	X	0
XYZ-INC.	M	X	X	48	0
	F	X	X	52	0

		AGE			
TABLE *AS*		0–20	21–45	46–65	>65
SEX	M	25	23	66	49
	F	27	16	53	51

	AGE			
TABLE *AE*	0–20	21–45	46–65	>65
EMPLOYER				
UNEMPLOYED	50	2	10	100
ABC-COMPANY	0	17	9	0
XYZ-INC.	2	20	100	0

	SEX	
TABLE *SE*	M	F
EMPLOYER	84	78
UNEMPLOYED	10	16
ABC-COMPANY	69	53
XYZ-INC.		

TABLE *ALL*:

$\boxed{310}$

	AGE			
	0–20	21–45	46–65	>65
TABLE *A*	52	39	119	100

	SEX	
	M	F
TABLE *S*	163	147

	EMPLOYER		
	UNEMPLOYED	ABC	XYZ
TABLE *E*	162	26	122

Figure 5. Protection by cell suppression in the model of Figure 3.

basic idea of cell suppression is illustrated in Figure 5 for the SDB shown in Figure 3.

A thorough study on cell suppression for the static SDB environment is presented in Denning et al. [1982]. They show that cell suppression becomes impractical if an arbitrary complex syntax for queries is allowed. (With such a syntax, suppression of complete tables from the lattice model might be necessary.) If, however, the syntax is restricted such that the query set is an elementary m-set (see Section 2),

$$(A_1 = a_{ij_1}) \\ \& (A_2 = a_{2j_2}) \& \cdots \& (A_m = a_{mj_m}),$$

cell suppression would remain of practical value. Fortunately, the summary tables used by the census bureaus usually take the form of such elementary m-sets.

The determination of complementary suppressed cells has been studied by Cox [1980]. He shows that the determination of a minimum set of complementary suppression involves a great deal of computational complexity. An insight into heuristics and software required to solve these problems in practice (the Canadian census) is included in Sande [1983]. It is interesting to note that Cox and Sande use an elaborate sensitivity criterion, called the $k\%$-dominance rule. According to this criterion, a cell is sensitive if the attribute values of two or three entities in the cell contribute more than $k\%$ of the corresponding SUM statistic.

Recently, Özsoyoğlu and Chung [1986] have published a study that is of interest for both the partitioning method and the cell-suppression method. It is concerned with preventing users from obtaining the information that a cell is of size 1. The problem is attacked by merging a cell of size 1 with a cell of size > 1. For this problem, an efficient heuristic is presented. The information loss may still become high (52%), however, thus limiting its application to real world SDBs.

In Table 1 we summarize the performance of the cell suppression method described in Cox [1980] with respect to the evaluation criteria.

4. DATA PERTURBATION

The methods based on the data-perturbation approach fall into two main categories, which we will call the probability-distribution category and the fixed-data-perturbation category. The probability-distribution category considers the SDB to be a sample from a given population that has a given probability distribution. In this case, the security-control method replaces the original SDB by another sample from the same distribution or by the distribution itself. In the fixed-data-perturbation category, the values of the attributes in the database, which are to be used for computing statistics, are perturbed once and for all. The fixed-data-perturbation methods discussed in the literature have been developed exclusively for either numerical data or categorical data. Accordingly, these methods will be discussed separately.

As shown in Figure 1b, the data-perturbation approach usually requires that a dedicated transformed database is created for statistical research. If the original database is also used for other purposes, the original database and the transformed SDB are both maintained by the system.

4.1 The Bias Problem

Before discussing the data-perturbation-based methods, it is important to note that this approach has a large risk of introducing bias to quantities such as the conditional means and frequencies. This point is stressed in Matloff [1986]. Following Matloff's notation and line of thought, the source of this bias can be explained as follows.

Let X denote the original value of an attribute to be perturbed, and let X' denote this perturbed value $X' = X + \alpha$. Now consider the set of entities that has a perturbed value w. Matloff shows that the expected value of X under the condition that $X' = w$; that is, $\mathrm{E}(X \mid X' = w)$ is not necessarily equal to w. More specifically, if X is a numerical-positive variable with a strictly decreasing density function (e.g., the exponential density function) to which a perturbation that is symmetrical around 0 has been added, Matloff [1986] shows that

$$\mathrm{E}(X \mid X' = w) < w.$$

In this case perturbation introduced bias in the response to the user query.

This result has important consequences for fixed-data perturbation. Each query in which the selection of the query set is based on perturbed values runs the risk of being biased. Matloff also shows that this bias can be considerable. For example, if X and Y are correlated attributes with a bivariate Gaussian distribution whose expected value is 0 and X is perturbed by an independent noise variable α with mean value zero and variance $\mathrm{Var}(\alpha)$, then the following bias occurs: Let

$$\begin{aligned} m(w) &= \mathrm{E}(Y \mid X = w) \\ &= (\mathrm{E}(XY)/\mathrm{Var}(X))w \\ m'(w) &= \mathrm{E}(Y \mid X' = w) \\ &= (\mathrm{E}(X'Y)/\mathrm{Var}(X'))w. \end{aligned}$$

Since α is independent of Y and both Y and α have a mean of zero, we have

$$\begin{aligned} m'(w) &= \left(\frac{\mathrm{E}(XY)}{\mathrm{Var}(X')}\right)w \\ &= \left(\frac{\mathrm{E}(XY)\mathrm{Var}(X)}{\mathrm{Var}(X')\mathrm{Var}(X)}\right)w \\ &= \left(\frac{\mathrm{Var}(X)}{\mathrm{Var}(X')}\right)m(w) \end{aligned}$$

or

$$\begin{aligned} m'(w) &= \left(\frac{\mathrm{Var}(X)}{\mathrm{Var}(X+\alpha)}\right)m(w) \\ &= \left(\frac{\mathrm{Var}(X)}{\mathrm{Var}(X)+\mathrm{Var}(\alpha)}\right)m(w) \\ &= \left(\frac{1}{1+\mathrm{Var}(\alpha)}\right)m(w) \end{aligned}$$

then

$$\frac{m'(w)}{m(w)} = \frac{1}{1 + \mathrm{Var}(\alpha)/\mathrm{Var}(X)}$$

Therefore, if $\mathrm{Var}(\alpha) = \mathrm{Var}(X)$, which is not unreasonable for perturbing noise, a bias of 50% occurs. We assume that such a bias is unacceptable. This bias problem will be further addressed in the context of the methods discussed below.

4.2 Probability Distribution

Within the probability-distribution category, two methods can be identified. The basic idea of the first is to transform the original SDB with another sample that comes from the same (assumed) probability distribution. This method is described in Reiss [1980, 1984] for multicategorical attributes and is called "data swapping" or "multidimensional transformation" [Schlörer 1981]. The method has been followed for categorical or numerical attributes by Liew et al. [1985]. The second method, described in Lefons et al. [1983], calls for replacing the original SDB by its (assumed) probability distribution. A discussion of each of these methods is presented.

4.2.1 Data Swapping

Reiss [1984] suggested a method that deals with multicategorical attributes called "approximate data swapping." The method, which extends earlier work by Reiss [1980] and Schlörer [1981], is described for Boolean attributes (0–1). According to Reiss [1984], however, it is straightforward to extend the method to the case in which attributes take any value within the set $\{0, 1, 2, \ldots, r-1\}$ for any arbitrary value r. In this method, the original database is replaced with a randomly generated database having approximately the same t-order statistics as the original database. (A t-order statistic is some statistical quantity that can be computed from the values of exactly t attributes [Schlörer 1983], for example, the number of patients whose Sex = Male and Disease = AIDS is two-order frequency count.) Due to the computational requirement of the method, it is only feasible to consider it for static SDBs where offline mode of usage is used (see Figure 1a). Even if the computational requirement were reduced, the following issues have to be resolved before its application to an online-usage mode would be practical.

(1) Every time a new entity is added or a current entity is deleted, the relationship between this entity and the rest of the database has to be taken into consideration when computing a new perturbation. The required algorithm is not straightforward.

(2) There is a need for a one-to-one mapping between the original database and the

Table 2. A Summary of the Performance

Security Control Method	Security: Exact Disclosure Possible?	Security: Partial Disclosure Possible?	Robustness	Suitable for Numerical or Categ. Attribute	Suitable for One or More Attributes	Suitability to Online Dynamic SDB
Data Swapping	No	Yes (difficult)	High	Categorical	More than one	Not suitable
Probab. Distribution by Liew et al.	No	Yes (easy especially for large SDBs)	For exact discl.—Moderate; for partial discl.—Very Low	Both	One	Not suitable for real time
Analytical Method	Yes in extreme cases	Yes (especially for large SDBs)	Moderate	Numerical	More than one	Moderate
Fixed Data Perturb. by Traub et al.	No	Can be balanced against security	Moderate	Numerical	One	Moderate
Fixed Data Perturb. by Warner	No	Can be balanced against security	Moderate	Categorical	One	Moderate

perturbed database. Although several alternative methods discussed in Reiss [1984] look promising, further investigation is required.

(3) The precision resulting from this method may be considered unacceptable since, as shown in Reiss [1984, p. 33], the method may in some cases have an error of up to 50%. In a static-SDB environment such extreme cases may be analyzed and corrected by the DBA before the release of the perturbed SDB. This is infeasible, however, in the case of the dynamic SDB.

(4) The small query-set (size 0 or 1) problem needs to be resolved; otherwise the database is vulnerable to such compromise as the regression-based one [Palley 1986; Palley and Simonoff 1987] (see Section 7).

A summary of the performance of this method is presented in Table 2. In general, data swapping has not been developed enough to be seriously considered for static or dynamic SDBs.

4.2.2 The Probability-Distribution Method by Liew et al.

Liew et al. [1985] describe a method for protecting a single confidential attribute in an SDB. The method is applicable to both categorical and numerical attributes. The generalization to multiple-dependent-confidential attributes is not, however, straightforward. The method consists of three steps:

(1) Identify the underlying density function of the attribute values and estimate the parameters of this function.

(2) Generate a sample series of data from the estimated density function of the confidential attribute. The new sample should be the same size as that of the database.

(3) Substitute the generated data of the confidential attribute for the original data in the same rank order. That is, the smallest value of the new sample should replace the smallest value in the original data, and so on.

As a result of the third step, this method could equally as well be classified under the fixed-data-perturbation category. For ease of discussion, we describe it in this section.

This method is equivalent to a noise-addition method, hence it introduces sampling bias in query responses [Matloff 1986]. The bias results from sampling from a population that is not the true-target of the Data Perturbation Methods

Richness of Information				Costs		
Amount of Nonconf. Info. Eliminated	Bias	Precision	Consistency	Initial Implementation Efforts	Processing Overhead Per Query	User Education
None	Yes (could be serious)	Could be very low	High	Very high	None	Moderate
None	Yes (could be serious for small SDBs)	Inversely related to size of the SDB	NA	Moderate	None	Very low
None	Yes (serious)	Moderate	NA	Very high	None	Very low
None	Yes (serious)	Can be balanced against security	Moderate except for extreme	Low	Very low	Very low
None	No	Can be balanced against security	Moderate	Low	Very low	Very low

population [Tendick and Matloff 1987]. For an SDB of small size, the noise introduced by this method is larger; thus better security is achieved but biased-query responses are provided to users. As the size of the database increases, the bias becomes smaller but less security of confidential attributes is achieved.

The evaluation of the performance of this method with respect to the evaluation criteria is summarized in Table 2. Partial disclosure is easily possible since the noise added to the confidential attribute becomes rapidly small, even for databases of moderate size. Additional knowledge about other entities in the database may enhance the likelihood of partial disclosure.

4.2.3 The Analytical Method

Lefons et al. [1983] describe a method for protecting multinumerical-confidential attributes. The method consists of estimating the joint probability function of several numerical attributes. The key contribution of this work lies in the approximation of the data distribution by orthogonal polynomials. The coefficients used in the computation of this approximation are called canonical coefficients. These coefficients are well suited for usage in an online environment because they can be adopted easily in case of insertions and deletions of the database entities.

Although the method looks promising, its security aspect needs further investigation. In particular, if the new probability-distribution function is a very precise description of the original data, then there is hardly any protection against partial disclosure. On the other hand, if deviations between the distribution function and the original data are possible, then issues such as how to avoid bias and how could the DBA exercise control on the trade-off between precision and security need to be addressed.

The evaluation of the analytic approach with respect to the criteria of Section 1 is presented in Table 2. Exact disclosure is possible in extreme cases. For example, if the distribution shows that 1% of the population satisfies certain criteria and it is known that the size of the original database amounts to 100, exact disclosure occurs.

4.3 Fixed-Data Perturbation

This section discusses the fixed-data-perturbation method for numerical attributes and the fixed-data-perturbation methods for categorical attributes.

4.3.1 Fixed-Data Perturbation for Numerical Attributes

Traub at al. [1984] developed a method that applies to numerical attributes. Suppose, for example, that the true value of a given attribute (e.g., salary) of an entity k is Y_k. The response to the sum query, under this method, will be

$$T = \sum_{k=1}^{n} X_k,$$

where

$$X_k = Y_k + e_k,$$

e_k is a random-perturbation variable with

$$\mathrm{E}(e_k) = 0 \qquad \text{and} \qquad \mathrm{Var}(e_k) = \sigma_e^2,$$

and $\{e_k\}$ are independent for different k's.

Under this method, the perturbation e_k of an entity k is fixed. Thus, it is not possible for snoopers to improve their estimates of a given statistic by repeating queries.

The additive-perturbation method described above suffers in terms of scale. For example, perturbing a salary of $150,000 by 3000 would be considered a compromise while at the same time perturbing a salary of $15,000 by 3000 would preverve the confidentiality of the data. Several alternatives to this basic method have been suggested in Traub et al. [1984]. One alternative is to apply multiplicative rather than additive perturbation, thus overcoming the scale problem.

Note that this method is akin to the probability-distribution method [Liew et al. 1985]. The bias problem, which was discussed earlier in this section, applies also to the fixed-data-perturbation method [Traub et al. 1984]. The method could, however, be saved for practical usage only if a bias-compensation mechanism is also developed and implemented. In this respect, this method has an advantage over the probability-distribution method [Liew et al. 1985], because the way in which noise is added to the data is much clearer and therefore better suited for statistical analysis. We expect that the bias problem for single attribute (or independent attributes) is solvable. Recently, Tendick and Matloff [1987] have suggested a bias-correction mechanism for both the multivariate normal and nonparametric cases. The application of these bias-correction mechanisms to the fixed-data-perturbation method [Traub et al. 1984] needs further study.

A summary of the evaluation of the fixed-data-perturbation method [Traub et al. 1984] with respect to the criteria of Section 1 is given in Table 2. Unlike other data-perturbation methods, this method is likely to be appropriate for usage in an online SDB. The original and perturbed values can be maintained, with users accessing only the perturbed database. Updates, deletions, and insertions affecting the original attributes can immediately be reflected into the perturbed values.

4.3.2 Fixed-Data Perturbation for Categorical Attributes: Basic Method

Warner [1965] developed the "randomized response" method for the purpose of applying it to data collection through a survey. The method deals with a single confidential attribute that can take only the value 0 or 1 (e.g., drug addiction). In order to describe this method as applied to the COUNT query in an SDB, we extend our hospital database example to include a confidential-categorical attribute (0–1): drug addiction. Consider the following query:

Q6: COUNT (Age = 42 & Sex = Male & Employer = ABC & Drug Addiction = Yes)

Let

n = the query-set size (true answer to the above query).

n_0 = the number of entities that satisfy the characteristic formula, excluding the clause that pertains to the confidential attribute; that is, COUNT (Age = 42 & Sex = Male & Employer = ABC). This will be referred to as the nonconfidential-response set.

Y_i = the value of the confidential attribute after applying data perturbation to entity i. Specifically, if X_i is the original value of the confidential attribute, then

$$Y_i = \begin{cases} X_i & \text{with probability } p \\ |1 - X_i| & \text{with probability } 1 - p \end{cases}$$

where the fixed parameter p is set by the DBA. Typical values for p are 0.6–0.8. The roles of p and $1 - p$ can be interchanged such that p might take values 0.2–0.4.

n_1 = the number of Y_i's with a value of 1 in the confidential-response set (i.e., the query-set size).

N = the response to the user. It is the unbiased-maximum-likelihood estimator of n (thus, no bias problem) and is given by

$$N = \frac{n_0(p - 1) + 1}{2p - 1}$$

where

$$\mathrm{Var}(N) = n_0\left[\frac{1}{16(p - \frac{1}{2})^2} - \left(\frac{n}{n_0} - \frac{1}{2}\right)^2\right]$$

Note that a correct determination of n_0 is crucial for this method. This, in general, is not a trivial problem. For the query Q6, it is easy to determine n_0 because the query consists of a nonconfidential-Boolean expression connected by an AND operator with one simple confidential clause. Thus, we are able to distinguish, in a straightforward way, between the nonconfidential-response set (with cardinality n_0) and its subset for which $Y = 1$ (with cardinality n_0).

Consider, however, the following query:

Q7: COUNT (Sex = Male & Drug Addiction = Yes + Employer = ABC & Drug Addiction = No)

In this case, defining n_0 and n_1 is not a trivial task. We outline a general procedure for dealing with this problem that aims at transforming the characteristic formula C into a formula C' of the form

$$(Y = 1) \,\&\, A_1 + (Y = 0) \,\&\, A_2 + A_3,$$

where $A_i \cap A_j = \emptyset$, for $i \leq j$ and A_i is nonconfidential for $i = 1, 2, 3$.

Step 1. Transform C into its most expanded form. Each term of the new formula contains a clause $(Y = 1)$, a clause $(Y = 0)$, or no reference to Y at all.

Step 2. Collect all terms with $Y = 1$, all terms with $Y = 0$, and the remaining terms. This yields a formula of the form $(Y = 1)$ & $B_1 + (Y = 0)$ & $B_2 + B_3$.

Step 3. Transform this formula into $(Y = 1)$ & B_1 & $(\neg B_2)$ & $(\neg B_3) + (Y = 0)$ & $(\neg B_1)$ & B_2 & $(\neg B_3) + B_1 + B_3$, which gives the required result (the formula C').

Step 4. During the query processing, the algorithm should use separate counters $n_0^{(2)}$ and $n_1^{(1)}$ for the first clause, $n_0^{(2)}$ and $n_1^{(1)}$ for the second clause, and $n^{(3)}$ for the third clause.

Step 5. After processing, the response N can be estimated as the sum of the three variables $N^{(1)}$, $N^{(2)}$, and $N^{(3)}$, where

$$N^{(1)} = \frac{n_0^{(1)}(p-1) + n_1^{(1)}}{2p-1}$$

$$N^{(2)} = \frac{n_0^{(2)}(p-1) + n_1^{(2)}}{2p-1}$$

$$N^{(3)} = n^{(3)}$$

This procedure shows that this fixed-data-perturbation method is a feasible solution for an SDB application, although its generalization is not trivial.

Table 2 includes a summary of the performance of the fixed-data-perturbation method by Warner [1965] with respect to the evaluation criteria. Notice that in this case, similar to the fixed-data-perturbation method described in Traub et al. [1984], it is possible to balance precision and security. A more detailed discussion of this issue is given in Section 6.

4.3.3 Fixed-Data Perturbation for Categorical Attributes: Extensions

The fixed-data perturbation for categorical data is an extension of Warner's [1965] method (as applied to data collection through interviews) to the case in which the population can be divided into t categories as given in Abul-Ela et al. [1967]. The confidential information is obtained by a "Yes" or "No" answer from each interviewee to precisely one randomly drawn question out of a set of $t - 1$ questions. Each question takes the form, "Do you belong to group i"? (for $i = 1, 2, \ldots, t - 1$). As in Warner's method, the interviewer is not aware of the particular question to which the interviewee is responding.

Applying this method to an SDB environment results in a considerable information loss since the original t-possible values of the confidential attribute are reduced to two possible values (yes or no). More specifically, assuming equal probability of each of the t values, the available information is reduced from $\log_2(t)$ bits to 1 bit. This information loss is considered a serious hindrance in applying the method to real-world SDBs.

It was shown that the above two methods of randomized response can be formulated as special cases of a general class of linear regression models [Warner 1971]. It was further suggested that this general model might be applied to perturb categorical data in a statistical database [Traub et al. 1984]. The two randomized-response methods discussed above are the only methods that we are aware of as being specific models that fall under the general class of models described by Warner. (We consider the methods described in Greenberg et al. [1969a, 1969b] to be minor variations of these two methods.) Therefore, for Warner's general class of models to be applicable to SDBs in online mode (see Figure 1b), new methods have to be developed. Such methods should have better performance with respect to their computational requirements and information loss.

The randomized-response methods discussed above do not address the case of more than one categorical attribute. The application of these methods to SDBs with multicategorical attributes requires further study.

The discussion presented in this section indicates that data-perturbation-based methods suffer from a bias problem, and/or not being suitable for dynamic SDBs, and/or being limited to one confidential attribute. A study of how to remedy these drawbacks is needed.

5. OUTPUT-PERTURBATION APPROACH

Before discussing the output-perturbation-based methods, we would like to note that the bias problem pointed out in the previous section is less severe here. This is due to the fact that the query set selected under the output-perturbation approach is based on the original values, not the perturbed values. Thus, the value of Y' in Section 4.1 is generated after selecting the appropriate values of X.

5.1 Random-Sample Queries

Denning [1980] proposed a method that is comparable to ordinary random sampling where a sample is drawn from the query set itself. Given a characteristic formula C, a set of entities that satisfies C is determined. For each entity i in the query set, the system applies a Boolean formula $f(C, i)$ to determine whether this entity is to be included in the sampled query set. The function f is designed in such a way that there is a probability P that the entity is included in the sampled query set. The probability P can be set by the DBA. The required statistics are computed based on the sampled query set. The statistics computed from the sampled query set have to be divided by P in order to provide a corresponding unbiased estimator. For example, if the response to a count query based on the sampled query set is n^*, an estimator for the true count is n^*/P. The sampling mechanism in this method is designed in such a way that lexicographically the same queries produce the same sample from the query set. Logically equivalent but lexicographically different queries, however, do not necessarily produce identical sampled query sets. As a result, independent estimates of a given statistic can be obtained by issuing logically equivalent but not identical queries. The method thus suffers from the resulting inconsistency.

The probability of an entity being included in the query set P is either fixed (independent of n) or variable (varies with n).

When P is fixed there should be a query-set-size restriction if P is large. Otherwise, there is a high probability of including all the entities of a small query set, thus compromising the database.

When P is variable, it should approach 0 for a query-set size approaching 1 in order to avoid compromising the database. This, however, results in an estimator whose variance, Var(n), approaches infinity. Therefore, in discussing this method we limit our attention only to the case of fixed P with a minimum query-set-size restriction. The minimum query-set-size restriction should be applied to n^* instead of n, otherwise it is easily shown that compromise can be achieved by a tracker of size $K - 1$, where the query-set size is less than or equal to K. We shall ignore this implementation issue in the remainder of the paper.

A summary of the performance of this method with respect to the evaluation criteria is given in Table 3, and additional discussion is included in Section 6.

Finally we note that a variant of the random-sample queries [Denning 1980] has been proposed in Leiss [1982]. According to the method by Leiss [1982], the modified query set is extended by a small sample from those entities in the SDB that are not included in the genuine query set. Unlike the random-sample queries [Denning 1980], it is difficult to avoid the bias resulting from the method in Leiss [1982]. Therefore, no further discussion of Leiss's method is included in this paper.

5.2 Varying-Output Perturbation

Beck [1980] suggested a method for SUM, COUNT, and PERCENTILE queries. The method introduces a varying perturbation to the data that are used to compute the response to a (eventually repeated) version of a given query. We will use the COUNT query and follow Beck's notation to illustrate the basic idea of the method. Let

n = the query-set size determined by the characteristic formula C.

N = the perturbed value of n; it is the response to users and is given by

$$N = \sum_{k=1}^{n} Z_{3,k},$$

where

$$Z_{3,k} = \sum_{i=1}^{n} Z_{3,k}^{(i)},$$

where $Z_{3,k}^{(i)}$ are independent random variables with

$$\mathrm{E}(Z_{3,k}^{(i)}) = \frac{1}{j} \quad \text{and} \quad \mathrm{Var}(Z_{3,k}^{(i)}) = \frac{a_1^2}{n}.$$

Thus

$$\mathrm{E}(Z_{3,k}) = 1 \quad \text{and} \quad \mathrm{E}(Z_{3,k}) = j\frac{a_1^2}{n}$$

and

$$\mathrm{E}(N) = n \quad \text{and} \quad \mathrm{Var}(n) = ja_1^2.$$

The variable Z_3 is the sum of j random variables $Z_3^{(i)}$ that change their values at varying rates. Each new draw of Z_3 causes at least a new draw of $Z_3^{(i)}$. After $(a_1/c_1)^2$ drawings of $Z_3^{(i)}$, however, a snooper might have an accurate estimator of n (although this estimator may be biased). Therefore, after $(a_1/c_1)^2$ drawings of $Z_3^{(i)}$, a new draw of $Z_3^{(i-1)}$ is made, and so on. Let $d = (a_1/c_1)^2$, then it takes d^j queries for snoopers to find their target with sufficient accuracy. Due to the varying rates of change of the values of $Z_3^{(i)}$, subsequent draw of Z_3 are highly correlated. The purpose of introducing this correlation is to make the number of queries needed for a disclosure grow exponentially with j.

The varying-output-perturbation method [Beck 1980] for SUM and PERCENTILE queries proceeds along the same line of thought. Here we will discuss only the SUM query.

Suppose we have a query SUM(Y) over a query set R, with n entities in R. Suppose further that the mean value of Y for the query set is $\bar{Y}_r$ and for the entire database is $\bar{Y}$. The response to the query will be

$$T = \sum_{k}^{n} X_k,$$

where $X_k = Y_k + Z_1(Y_k - \bar{Y}_r) + Z_2$.

In this equation Z_1 and Z_2 are independent random variables with

$$\mathrm{E}(Z_1) = 0 \quad \text{and} \quad \mathrm{Var}(Z_1) = 2ja^2$$

$$\mathrm{E}(Z_2) = 0$$

and

$$\mathrm{Var}(Z_2) = j\frac{2a^2}{n}(\bar{Y}_r - \bar{Y})^2.$$

Table 3. A Summary of the Performance

Security Control Method	Security: Exact Disclosure Possible?	Security: Partial Disclosure Possible?	Robustness	Suitable for Numerical or Categ. Attribute	Suitable for One or More Attributes	Suitability to Online Dynamic SDB
Random Sample Queries by Denning	No	Yes (can be balanced against precision)	Moderate	Both	More than one	High
Varying Output Perturb. by Beck	No	Yes (can be balanced against precision)	Moderate	Numerical	One or several independent ones	High

It can be shown that

$$E(T) = \sum_{k=1}^{n} Y_k,$$

$$\mathrm{Var}(T) = 2jna + 2S_r^2 + 2ja^2(\bar{Y}_r - \bar{Y})^2,$$

where

$$S_r^2 = \frac{\sum_{k=1}^{n} (\bar{Y}_r - \bar{Y})^2}{n}$$

is the sample variance over the set R.

Given Beck's criterion of compromise, the value Y_k of some individual k by an estimator $\hat{Y}_k$ through repeated queries is

$$\mathrm{Var}(\hat{Y}_k) < c^2(Y_k - \bar{Y})^2$$

Beck showed that d^j queries (with $d = (a/c)^2$) are required for partial compromise.

Table 3 shows the evaluation of this method with respect to the criteria discussed in Section 1. Additional discussion is included in Section 6.

5.3 Rounding

Output perturbation may take some form of rounding, where the answer to the query is rounded up or down to the nearest multiple of a certain base b. Systematic rounding [Achugbue and Chin 1979], random rounding [Fellegi and Phillips 1974; Haq 1975, Haq 1977], and controlled rounding [Dalenius 1981] are three types of rounding that have been investigated. The basic idea of each of the rounding methods is summarized below.

Let $r = |C| (\mathrm{mod}\, b)$ and $\lfloor X \rfloor$ = the value of X rounded downward.

5.3.1 Systematic Rounding

The response to the user

$$= \begin{cases} |C| & \text{if } r = 0, \\ |C| - r & \text{if } r < \lfloor (b+1)/2 \rfloor \\ |C| + b - r & \text{if } r \geq \lfloor (b+1)/2 \rfloor \end{cases}$$

5.3.2 Random Rounding

The response to the user

$$= \begin{cases} |C| & \text{if } r = 0, \\ |C| - r & \text{with probability } \dfrac{1-r}{b} \\ |C| + b - r & \text{with probability } \dfrac{r}{b} \end{cases}$$

5.3.3 Controlled Rounding

Consider a database in a tabular form and suppose that, due to security reasons, it was decided to round the true statistic n_{ij} of cell i, j in the table. As shown above, systematic and random rounding calls for adding a quantity $+d_{ij}$ or $-d_{ij}$ to n_{ij}. According to controlled rounding, the same quantity $+d_{ij}$ (or $-d_{ij}$) is added to three other cells. These cells are chosen in such a way that the released row sum, r_i, and column sum, r_j, equal the true row sum, n_i, and the true column sum, n_j, respectively.

Random rounding suffers from two major drawbacks [Achugbue and Chin 1979]:

(1) It is possible (with small probability, however) for a randomly rounded row of an

of the Query Restriction Based Methods

Richness of Information				Costs		
Amount of Nonconf. Info. Eliminated	Bias	Precision	Consistency	Initial Implementation Efforts	Processing Overhead Per Query	User Education
Low	None	Can be balanced against security	Low	Low	Low	Low
Low	None	Can be balanced against security	Low	Moderate	Moderate	Very high

SDB in a tabular form to determine the exact original values of the row cells.

(2) It is possible to determine the true value, n_{ij} (or greatly narrow its range) by averaging the responses to the same query.

Systematic rounding introduces nonzero bias. Furthermore, systematic rounding can be circumvented by derounding or using a tracker [Denning and Schlörer 1983].

Generally, rounding is not considered an effective security-control method. But combining rounding with other security-control methods seems to be a promising avenue. For example, in Özsoyoğlu and Su [1985], rounding was used as an additional security-control method.

Before concluding this section, it should be pointed out that the output-perturbation approach does not add noise to the data and thus does not suffer from a serious bias problem. It suffers, however, from the possibility of having a null query set, thus providing valuable information to a snooper. Compromised methods such as regression-based ones (see Section 7) could easily take advantage of the null-query-set situation.

6. COMPARATIVE ANALYSIS OF THE SECURITY-CONTROL METHODS

The discussion presented in the previous sections shows that the random-sample-queries method [Denning 1980], the varying-output-perturbation method [Beck 1980], the fixed-data-perturbation method [Traub et al. 1984], and the fixed-data-perturbation for categorical data method [Warner 1965] are clearly among the most promising security-control methods for on-line, dynamic SDBs (the most difficult type of SDBs). This section presents a comparison of these methods based on the evaluation criteria discussed in Section 1 as applied to both the COUNT and SUM queries.

6.1 Security Criterion for the COUNT Query

Partial disclosure occurs if a snooper attempts (through issuing m queries) to obtain an estimator $\hat{n}$ for the true count value, n, whose perturbed value is N and the variance of that estimator satisfies the following:

$$\mathrm{Var}(\hat{n}) < c_1^2,$$

where c_1 is a parameter that is set by the DBA. We want to determine the value of m that will result in

$$\mathrm{Var}(\hat{n}) < c_1^2$$

under each of the methods.

Let $\hat{n}$ be an estimate of n obtained after m_R repeated queries when using the random-sample queries [Denning 1980]. We then have

$$\mathrm{Var}(\hat{n}) = \frac{\mathrm{Var}(N)}{m_R} \quad \text{since answers to repeated queries are independent}$$

or

$$m_R = \frac{\mathrm{Var}(N)}{\mathrm{Var}(\hat{n})} = \frac{\mathrm{Var}(N)}{c_1^2},$$

but

$$\mathrm{Var}(N) = \frac{n(1-P)}{P}.$$

Hence,

$$m_R = \frac{n(1-P)}{Pc_1^2}.$$

When the varying-output-perturbation method [Beck 1980] is used, the answers to repeated queries are not independent and the number of repeated queries, m_V, that will result in

$$\mathrm{Var}(\hat{n}) = c_1^2$$

is given by

$$m_V = \left(\frac{a_1^2}{c_1^2}\right).$$

Under equal precision to users it can be easily shown that $m_R \le m_V$, since for equal precision we have

$$\frac{n(1-P)}{P} = ja_1^2.$$

Dividing both sides by c_1^2 we get

$$\frac{n(1-P)}{Pc_1^2} = j\left(\frac{a_1^2}{c_1^2}\right)$$

or

$$m_R = j\left(\frac{a_1^2}{c_1^2}\right)$$

as compared to

$$m_V = \left(\frac{a_1^2}{c_1^2}\right)^j.$$

Under certain circumstances (e.g., small n) the method [Denning 1980] provides a precision that might not be attainable by Beck's [1980] method. In such cases, $m_R < m_V$ as shown below since

$$\frac{n(1-P)}{P} < ja_1^2,$$

or

$$m_R c_1^2 < ja_1^2,$$

or

$$m_R < j\left(\frac{a_1^2}{c_1^2}\right),$$

which is clearly less than m_V.

Finally, we notice that under Beck's method [1980], unlike under Denning's [1980], Var(N) has an attractive property of being a constant. Thus, a user always experiences the same error.

Under fixed-data perturbation, we restrict our discussion to the method described by Warner [1965]. For this case, a snooper might easily acquire the value of the perturbed attribute. Therefore, the probability $1 - p$ of perturbing a specific confidential attribute should be sufficiently large to prohibit any disclosure occurring if the perturbed value of that attribute is revealed. Thus, $1 - p$ would typically be set to, say, 0.20 to 0.40 (or 0.60 to 0.80, which is virtually the same). As can be seen from the formula for N in Section 4.3.2, the parameter p must not be set equal to 0.50. The fixed-data-perturbation method for categorical data [Warner 1965], in the variation described here, cannot be compromised.

The above discussion leads to the conclusion that, from the point of view of security of the COUNT query, Warner's method [1965] is superior to both the varying-output-perturbation method [Beck 1980] and the random-sample-queries method [Denning 1980]. Furthermore, the varying-output-perturbation method is superior to the random-sample-queries method.

6.2 Precision Criterion for the COUNT Query

A user who is interested in obtaining an estimate for the true value of the count n would obtain a value N as a response to the issued query. All methods (considered in this section) ensure that N is an unbiased estimator of n, that is, $\mathrm{E}(N) = n$.

A summary of the following discussion is presented in Table 4.

(1) Under the random-sample-queries method [Denning 1980], the variance of N is given by $\mathrm{Var}(N) = n(1 - P)/P$. Two input parameters are required for this

Table 4. The Precision Criterion for the COUNT Query

Factors	Random Sample Queries	Varying Output Perturbation	Fixed Data Perturbation by Warner
1. Precision: $V(N)$	$V(N) = \frac{n(1-P)}{P}$	$V(N) = ja_1^2$	$V(N) = n_0\left(\frac{1}{16(p-\frac{1}{2})^2} - \left(\frac{n}{n_0} - \frac{1}{2}\right)^2\right)$
2. Required input parameters and their recommended values	$.5 \le p \le .90$ Minimum query set size could be as low as 9	$4 \le j \le 30$ and $.5 \le c_1 \le 1.0$ and $a_1^2 = c_1^2 d$ and $d = 3.0$	$.6 \le P \le .8$
3. Range of Precision			
Min value of $V(N)$:	$.11n$ (when $p = .9$ and $n > 9$)	3.0 (when $j = 4$ and $c_1 = .5$)	$\frac{4}{9}n_0\left(\text{when } \frac{n}{n_0} = 1 \text{ and } p = .8\right)$
Max value of $V(N)$:	n (when $p = .5$ and $n > 9$)	90.0 (when $j = 30$ and $c_1 = 1.0$)	$\frac{25}{9}n_0\left(\text{when } \frac{n}{n_0} = \frac{1}{2} \text{ and } p = 0.6\right)$
4. Need for a minimum query set size	Yes	No	No
5. Confidence interval for n	Cannot be published since $V(N)$ is dependent on n	Can be published since $V(N)$ is independent of n	Cannot be published since $V(N)$ is dependent on n (however, see text)

method: the probability P and the minimum query-set size. The probability P that an entity is included in the sampled query set typically ranges between 0.5 and 0.9. The minimum query-set size, on the other hand, could be as low as 9. These values of P and the minimum query-set size result in a precision varying from $\text{Var}(N) = 0.11n$ (for $P = 0.9$ and $n < 9$) to $\text{Var}(N) = n$ (for $P = 0.5$ and $n > 9$).

We note that the confidence interval of N is a function of n, P, and the confidence level $(1 - \alpha)$. If P and α are known, knowledge of the confidence interval would lead to disclosure on n.

(2) Under the varying-output-perturbation method [Beck 1980], the variance of N is given by $\text{Var}(N) = ja_1^2$, where j is a parameter set by the DBA that controls the minimum number of queries needed to compromise the database. Typical values of j are in the range of 4 to 30, and $a_1 = (d)^{1/2}$. The values of the parameters c_1 and d are also set by the DBA, with c_1 controlling the desired degree of protection and typically taking a value < 1.0. The parameter d, on the other hand, follows from balancing precision against security, and its optimal value is shown to be 3.0.

We observe that under the varying-output-perturbation method [Beck 1980] the variance $\text{Var}(N)$ can take as low of a value as 3.0 (when $j = 4$ and $c_1 = 0.5$) and as large of a value as 90 (when $j = 30$ and $c_1 = 1$). Under the random-sample-queries method [Denning 1980], however, $\text{Var}(n)$ can take as low of a value as 1.11 (when $n = 10$ and $P = 0.9$) and can take as large of a value as n (when $P = 0.5$). In general, the random-sample-queries method may result in somewhat better precision for small query-set size, and the varying-output-perturbation method may result in a better precision for large query-set sizes.

Under the varying-output-perturbation method (unlike under the random-sample-queries method), the variance $\text{Var}(N)$ is constant and independent of the query-set size n. Therefore, publishing confidence intervals of N would not provide additional information on the value of n, and consequently no disclosure would be possible.

It is worth pointing out that, according to the varying-output-perturbation method, the response to a count query with an empty set is always 0. This may lead to a disclosure. One possible solution to this problem is to follow the suggestion in Beck [1980, p. 330] for dealing with a similar problem with the SUM query. Specifically, it is possible to enforce a lower bound (ja_1^2) for the variance of any COUNT query. This modification does not, however, introduce any autocorrelation among the responses to different but logically equivalent queries; thus, compromise is easier than without this modification.

(3) Under the fixed-data-perturbation method for one categorical confidential attribute [Warner 1965], the variance of N is given by

$$\text{Var}(N) = n_0\left(\frac{1}{16(p-\frac{1}{2})^2}\right) - \left(\frac{n}{n_0} - \frac{1}{2}\right)^2.$$

The parameter p takes a value between say 0.6 and 0.8. These values result in a precision varying from $\text{Var}(N) = \frac{4}{9}n_0$ (for $p = 0.8$ and $n/n_0 = 1$ or 0) to $\text{Var}(N) = \frac{25}{4}n_0$ (for $p = 0.6$ and $n/n_0 = 0.5$). In general, $\text{Var}(N) \leq kn_0$, where k is a function of p.

A confidence interval cannot be given because $\text{Var}(N)$ depends on n. A confidence interval that is based on the upper bound of $\text{Var}(N)$ will not lead to a disclosure, however, since n_0 is not considered confidential.

6.3 Security Criterion for the SUM Query

When comparing the security of the random-sample-queries and the varying-output-perturbation methods, a reasoning similar to the discussion on the COUNT query can be applied. Such a reasoning leads, again, to the conclusion that as far as security is concerned, the varying-output-perturbation method is superior to the random-sample-queries method. The fixed-data-perturbation method for one categorical-confidential attribute [Warner 1965] cannot be compromised and is thus considered superior to the other methods.

6.4 Precision Criterion for the SUM Query

A user who is interested in obtaining the sum of a given attribute Y in each entity of the response set would obtain a value of T as a response to the issued query. Similar to the COUNT query, all methods (considered in this section) ensure that T is an unbiased estimator of the true value. Table 5 summarizes the following discussion.

(1) Under the random-sample-queries method, the variance of T is approximately given by

$$\text{Var}(T) = n \text{ Var}(Y_r) \frac{(1 - P)}{P},$$

where $\text{Var}(Y_r)$ is the sample variance over the response set. As with the COUNT query, there are two input parameters: P and the query-set size.

The precision $\text{Var}(T)$ has an intuitively attractive property: $\text{Var}(T)$ is proportional to the sample variance over the response set. The precision ranges from $\text{Var}(T) = 0.11n \text{ Var}(Y_r)$ to $\text{Var}(T) = n \text{ Var}(Y_r)$, analogous to the COUNT query under the random-sample-queries method.

Publishing a confidence interval for T does not lead to immediate disclosure [because it is based on the product of n and $\text{Var}(Y_r)$], but it may facilitate compromise for a snooper who has additional knowledge of either n or $\text{Var}(Y_r)$.

(2) Under the varying-output-perturbation method [Beck 1980], the variance of T is given by

$$\text{Var}(T) = 2ja^2 n \text{ Var}(Y_r) + 2ja^2(\bar{Y} - \bar{Y}_r)^2,$$

where $\bar{Y}_r$ and $\bar{Y}$ are, respectively, the mean value of Y in the response set and the database. Similar to the COUNT query, j and a^2 are the input parameters. The value of j ranges from 4 to 30; the value of a^2 follows from $a^2 = dc^2$, where d is again recommended to equal to 3. The value of c would most likely range from 0.1 to 0.5. This leads to the following approximate minimum value of the variance $\text{Var}(T)$,

$$0.24n \text{ Var}(Y_r) + 0.24(\bar{Y} - \bar{Y}_r)^2,$$

and to a maximum value of

$$25n \text{ Var}(Y_r) + 45(\bar{Y} - \bar{Y}_r)^2.$$

As was previously mentioned, the response to any SUM query is required to have at least a variance equal to $ja\bar{Y}^2$, which is equivalent to a numerical value between $0.48\bar{Y}^2$ and $90\bar{Y}^2$. For large n, the value of $\text{Var}(T)$ is approximately proportional to $\text{Var}(Y_r)$, similar to the random-sample-queries method. The attractive property of the COUNT query under the varying-output-perturbation method viz. a precision that is independent of n is, however, not retained in the SUM query. Publishing a confidence interval for T does not lead to immediate disclosure, but might facilitate compromise for a snooper who has additional knowledge of n, $\text{Var}(Y_r)$, $\bar{Y}$, or $\bar{Y}_r$.

(2) Under the fixed-data-perturbation method [Traub et al. 1984], the precision is based on applying Chebyshev's inequality, which holds for an arbitrary distribution of the fixed perturbation e. The variance of T is given by

$$\text{Var}(T) = j\sigma_e^2.$$

The only parameter to be specified is σ_e. This parameter may virtually take any value. Consequently, any precision can be achieved, subject to the condition that $\text{Var}(T)$ is proportional to n. Publishing a confidence interval may simplify the task of the snooper in disclosing the value of n.

Traub et al. [1984] describe a variation of their method in which the standard deviation of the error e_k which is proportional to σ_e might be too small for large values of Y_k or too large for small values of Y_k. This results in a method in which

$$X_k = eY_k + Y'_k,$$

where $\text{E}(e) = 0$ and $\text{Var}(e) = \sigma_t^2 \cdot$ This means that

$$\text{Var}(T) = n\sigma_t^2 Y_r^2,$$

given that we adhere to the security criterion given by

$$\sigma(Y_k) < c_2 Y'_k,$$

Table 5. *The Precision Criterion for the SUM Query*

Factors	Random Sample Queries	Varying Output Perturbation	Fixed Data Perturbation by Traub et al.
1. Precision: $V(T)$	$V(T) = \frac{nV(V) \cdot (1-p)}{P}$	$V(T) = 2ja^2nV(Y)$ $+ 2ja^2(\bar{Y} - \bar{Y}_r)^2$	$V(T) = n\sigma_e^2$; alternatively, $V(T) = n\sigma_t^2 V'(Y)$
2. Required input parameters and their recommended values	$.5 \leq p \leq .90$ Minimum query set size, could be as low as 9	$4 \leq j \leq 30$ and $.1 \leq c_1 \leq .5$ and $a_1^2 = c_1^2 d$ and $d = 3.0$	σ_e no specific guidelines or restrictions; alternatively, $.2 \leq \sigma_t < 1.0$
3. Range of precision			
Min value of $V(T)$:	$.11nV(Y)$ (when $p = .9$ and $n > 9$)	$.24nV(Y) + .24(\bar{Y} - \bar{Y}_t)^2$ (when $j = 4$ and $C = .2$)	$.01nV'(Y)$ (when $J = .1$)
Max value of $V(T)$:	$nV(Y)$ (when $p = .5$ and $n > 9$)	$45nV(Y) + 45(\bar{Y} - \bar{Y}_r)^2$ (when $j = 30$ and $C = 1.0$)	$.25nV'(Y)$ (when $\sigma_e = .5$)
4. Confidence interval for T:	May be published but increases risk of disclosure	May be published but increases risk of disclosure	May be published but not in cases where confidential data exist in addition to numerical data

where c_2 is comparable to the parameter c in the varying-output-perturbation method [Beck 1980]. Therefore, reasonable values for c_2 range from 0.1 to 0.5.

6.5 Consistency Criterion

By its very nature, fixed-data-perturbation-based methods yield consistent responses to all queries. Contrary to this are both the varying-output-perturbation method [Beck 1980] and the random-sample-queries method [Denning 1980]. These methods normally provide users with inconsistent results. The consistency of the fixed-data-perturbation-based methods does not, however, prevent paradoxical values from occurring. For example, a negative salary is possible under the fixed-data-perturbation method [Traub et al. 1984]. In addition, under the fixed-data-perturbation method for one categorical-confidential attribute [Warner 1965] the estimated number of entities that have a given property, out of a nonconfidential-response set of size n_0, may occasionally be less than 0 or larger than n_0. Such "strange" results can also be obtained from both the random-sample-queries and the varying-output-perturbation methods. We feel that this is a general drawback of perturbation approaches as compared to other approaches such as query restriction.

6.6 Robustness Criterion

In general, a perturbation method is robust with respect to increased snooper's knowledge if the perturbation added to a specific value does not depend on other values in the query set.

Y_i represent a nonperturbed attribute-value of entity i,

X_i represent the perturbed attribute-value, and

e_i denote the perturbation under query $q(C)$.

Then

$$e_i = \begin{cases} X_i - Y_i & \text{for numerical data} \\ |X_i - Y_i| & \text{for categorical data} \end{cases}$$

The method is fully robust if e_i is dependent on Y_j only for $i = j$ for all $q(C)$.

Note that this definition of robustness encompasses attacks to dynamic databases by knowledge of insertions, deletions, or updates of records. Thus, the fixed-data-perturbation methods [Traub et al. 1984; Warner 1965] are robust. The random-sample-queries method [Denning 1980] is also fairly robust, although increased snooper's knowledge might be used to circumvent the effect of the minimum query-set-size restriction. The varying-output-perturbation method [Beck 1980] is much less robust because e_i is dependent on all Y_i in the query set.

6.7 Cost Criterion

In any SDB an initial effort is required to develop a "statistical filter" that would process all queries before handing them to the "normal query processor." The statistical filter ensures that

- a user can only access aggregate data (e.g., COUNT and SUM), and
- a user cannot access any directly identifying attribute (e.g., name or Social Security number).

Furthermore, as discussed in Section 1, the implementation of a given method involves three aspects: setting the input parameters, educating users, and making additional programming efforts to implement the specific method.

Each of the methods considered in this section requires the setting of few input parameters (see Tables 4 and 5). In all methods, once the basic policy decision of what is considered a disclosure has been made, setting the input parameters is straightforward.

With respect to user education, effective usage of the SDB requires understanding of the basic idea of the method in use. In relation to each other, the methods can be categorized as follows: The random-sample-queries method [Denning 1980] and the fixed-data-perturbation method [Traub et al. 1984] are relatively easy to explain, whereas the varying-output-perturbation method [Beck 1980] and the fixed-data-perturbation method for categorical data [Warner 1965] are more difficult to explain.

All methods can be implemented in such a way that the perturbation is applied during the query processing. Fixed-data-perturbation-based methods can, however, also be implemented such that data are perturbed upon data entry. The methods can be ranked in increasing order of implementation effort as follows: the fixed-data-perturbation method, the random-sample-queries method, the varying-output-perturbation method, and the fixed-data-perturbation method for categorical data.

With respect to the processing overhead per query, the random-sample-queries method is more efficient as compared to the varying-output-perturbation method. If the fixed-data-perturbation method and the fixed-data-perturbation method for categorical data are implemented once, at the time data are entered into the system, the CPU-time requirement would clearly be less than that of the random-sample-queries method. In this case, the online-storage requirement would be considerably more (assuming that under the fixed-data-perturbation methods the original SDB is also kept online). On the other hand, if the perturbed attribute values are generated every time they are accessed, additional storage requirement is avoided at the expense of an increased CPU-time requirement. As a result, the CPU time requirement of the fixed-data-perturbation method and the fixed-data-perturbation method for categorical data would be comparable to that of the random-sample-queries method. In practice, the bottleneck for the user is typically the input/output time requirement, which is determined by the database management system and is hardly affected by the security-control method.

6.8 Combination of the Traub et al. Method with Other Methods

Before concluding this section, it should be pointed out that the security of the categorical data may be violated indirectly by means of statistics on numerical data. In general, if we have a confidential-categorical attribute, the associated security method should be applied whenever this attribute is accessed in a given query, independent of the nature of the query (e.g., COUNT or SUM). To illustrate, consider the following example in which the fixed-data-perturbation method [Traub et al. 1984] is used for protecting numerical attributes and some other method is used for protecting categorical attributes:

SUM (Salary)
where (Age = 42 & Sex = Male & Employer = ABC)
SUM (Salary)
where (Age = 42 & Sex = Male & Employer = ABC & Diagnosis Type = Schizophrenia)

If the protection method for categorical attributes is not applied to the SUM query, a compromise occurs when the responses to both queries are equal (assuming that the target is a 42-year-old male working for the ABC organization).

The fixed-data-perturbation method can be combined with the fixed-data-perturbation method for categorical data, which is suitable for protecting a single Boolean-confidential attribute. For the general case in which more confidential-categorical attributes are involved, a combination of the fixed-data-perturbation, the random-sample-queries, and the varying-output-perturbation methods is an alternative that is worth investigating.

The combination of the fixed-data-perturbation and the random-sample-queries methods is easily implemented because the methods are nearly complementary. As far as precision of the SUM query is concerned, note that when combining both methods we get

$$T = \left(\frac{n}{n^*}\right) \sum_{k=1}^{n^*} X_k = \left(\frac{n}{n^*}\right) \sum_{k=1}^{n^*} (Y_k + e_k).$$

Thus,

$$\mathrm{Var}(T) = n\,\mathrm{Var}\left(\left(\frac{1}{n^*}\right) \sum_{k=1}^{n^*} Y_k\right) + n\,\mathrm{Var}\left(\left(\frac{1}{n^*}\right) \sum_{k=1}^{n^*} e_k\right)$$

Using the results in Cox [1980], we then have

$$\mathrm{Var}(T) = \frac{n(1-P)}{P}\mathrm{Var}(Y_r) + \sigma_e^2.$$

In other words, the variance under the combined method equals the sum of the variances of each method separately. The security and precision, with respect to the categorical attribute, are the same with the combined method as with the random-sample-queries method.

The combination of the fixed-data-perturbation method and the random-sample-queries method would result in the following response to the SUM query:

$$T = \sum_{k=1}^{n} Z_{3,k}X_k = \sum_{k=1}^{n} Z_{3,k}(Y_k + e_k).$$

Therefore,

$$\begin{aligned}\mathrm{Var}(T) &= \sum_{k=1}^{n} \mathrm{Var}(Z_{3,k}Y_k) + n\,\mathrm{Var}(Z_3 e)\\ &= \mathrm{Var}(Z_3)\sum_{k=1}^{n} Y_k^2 + n(\mathrm{E}(Z_3))^2\sigma_e^2\\ &= \left(\frac{1}{n}\right) j a_1^2\,(n\bar{Y}_r^2 + n\,\mathrm{Var}(Y_r)) + n\sigma_e\\ &= j a_1^2 \bar{Y}_r^2 + j a_1^2 \mathrm{Var}(Y_r) + n\sigma_e.\end{aligned}$$

In this case, the resulting variance is not *equal* to the sum of the variances of both methods separately, but is less for large n. This is due to the fact that, as was previously discussed, the precision of the COUNT query under the varying-output-perturbation method is essentially different from the SUM query. Again, the security and precision of the categorical data are the same under the combined method.

7. NEW TYPES OF THREATS

Several types of threats are just starting to be explored in the literature. These threats are quite different in nature from the ones examined by researchers concerned with the security of SDBs. It is important, therefore, that researchers concerned with the security of SDBs be aware of and give attention to these new types of threats. A discussion of these threats follows.

(1) Logical inference. Morgenstern [1987] addressed the threat to nonstatistical databases that arise from logical inference and the semantics of the application. Some approaches designed to overcome the logical-inference problem in nonstatistical-database environments have already been suggested in the literature [Denning 1984, 1985; Su and Özsoyoğlu 1987]. The extension of such a threat to SDBs is yet to be explored.

(2) Diophantine inference. Rowe [1984] examined situations in which a domain-dependent structure exists for a confidential attribute such that it can be characterized by very few independent variables. For example, in a university database we would have

$$n_1 * \mathrm{salary}_1 + n_2 * \mathrm{salary}_2 + n_3 * \mathrm{salary}_3 = \mathrm{total\_salary},$$

where n_1, salary_1, n_2, salary_2, and n_3, salary_3 are, respectively, the number and average salary of assistant, associate, and full professors.

The above linear equation is referred to as Diophantine (integer-solution) equation. In these situations, Rowe [1984] discussed ways of obtaining a finite set of possible values for each of the unknown variables in the Diophantine equation and for pruning this set using additional equality constraints on the possible values of the unknown variables. He stated that Diophantine inferences are very sensitive to changes in the coefficients involved, and it is very difficult to analyze expected and worst-case time complexities of solution methods for different problems. He concluded by pointing out that data or output perturbation seem to be the only real possible methods for protecting the database.

Although the practicality of this compromise method has not been studied, researchers concerned with security-control methods for SDBs need to take such a threat into consideration.

(3) Regression methodology. Palley [1986] and Palley and Simonoff [1987]

discussed the use of regression analysis to compromise SDBs. Their method calls for developing a synthetic database from the "original" database:

(i) Decide on a set of nonconfidential attributes $X = \{x_1, x_2, \ldots, x_k\}$ that would be useful for describing and predicting the confidential attribute (y). Develop a histogram for each x_i by asking queries of the form, "COUNT WHERE x_i = value".

(ii) For each of the k attributes, draw a random sample (sample$_i$, for $i = 1, 2, \ldots, k$) from the corresponding histogram. Issue the following queries: "COUNT y WHERE x_1 = sample$_1$, x_2 = sample$_2$, ..., x_k = sample$_k$", "MEAN y WHERE x_1 = sample$_1$, x_2 = sample$_2$, ..., x_k = sample$_k$", "STANDARD DEV. y WHERE x_1 = sample$_1$, x_2 = sample$_2$, ..., x_k = sample$_k$". Repeat for, say, 300 samples for each of the k attributes.

(iii) Record the results of the above step (all samples of the k attributes and corresponding value of the confidential attribute) in a synthetic database.

Once the synthetic database has been created, it is then used to estimate the β coefficients in the regression equation $y = X\beta + e$. A snooper can then use this regression equation to obtain an *estimate of the average value of y* for a given value of X attributes.

We agree with the authors that it is generally undesirable to have users of an SDB develop regression models that represent the functional relationship between a confidential attribute and a set of nonconfidential attributes. In general, however, the effectiveness of a regression model is limited by the existence of a regression relationship between confidential and nonconfidential attributes. Furthermore, the following two fundamental questions need to be addressed:

(1) Are users able, by using regression, to obtain a value of a confidential attribute that cannot be obtained directly from the security-control method?

(2) Are users able, by using regression, to obtain a better estimate of a confidential attribute than the one provided by the security-control method?

The answer to these questions should be considered in the context of the security-control method that is in effect. Specifically, consider the following two cases:

(a) The database is secured using one of the methods that withholds statistics that could lead to compromise. In this case, regression could be used to obtain an *estimate* (*not the actual value*) of the withheld statistic of the confidential attribute given a specific value of each of the nonconfidential attributes. The question, How good is this estimate? has not been fully investigated. Palley and Simonoff [1987] have mainly focused on the "R-squared" as the key measure for the performance of the regression-compromise method and give little attention to other measures such as the correlation between the withheld statistic and its regression estimate and the average error (difference between the withheld statistic and its regression estimate).

(b) The database is secured using one of the methods that does not withhold query responses (e.g., output or data perturbation-based method). In this case, the user can, legitimately, obtain directly from the security-control method, as well as using regression, an estimate of the desired aggregate statistic of the confidential attribute. The quality of the estimate obtained from the security-control method in comparison to the one obtained from the regression-compromise method depends on the security-control method that is in effect. In general, the security-control method makes use of the information contained in the whole database and the whole query set, whereas the regression-compromise method makes use of only a subset of the database and a subset of the query set. Therefore, it stands to reason, however, that the estimate obtained from the security-control method is better than the one obtained from the regression-compromise method.

8. CONCLUSIONS

Our conclusions are summarized in the following points.

(1) No single security-control method satisfies the conflicting objectives—high level of security, high level of richness of information provided to users, high level of robustness, low level of initial and processing costs—and is applicable to all types of SDB environments. An effective solution approach is to combine several methods into one that would be well suited for a general class of SDB environment, such as dynamic or static SDBs.

(2) The majority of the security-control methods developed to date have viewed the SDB as a file. Security-control methods should be based on the fact that an SDB is a database in which interrelated data about various types of populations are included. In this respect, the conceptual approach provides a useful framework. Extensive study of this approach is still required before wide-scale application is feasible.

(3) For dynamic online SDB environments, perturbation-based methods are well suited. There is a need to develop bias-correction mechanisms that would help overcome the bias problem that was pointed out in Matloff [1986]. The fixed-data-perturbation method [Warner 1965], which results in zero bias, is suitable for SDBs where there is only one confidential categorical attribute. Despite the fact that the random-sample-queries method [Denning 1980] has a low level of consistency, it is a viable alternative for SDBs where several dependent attributes are involved. The major problem with this method, however, is the small query-set size that could result in a partial disclosure or even exact disclosure in the case of a query set of size 1.

Auditing and partitioning are two security-control methods whose practical application to dynamic online SDB environments needs further study. The combination of partitioning and perturbation methods has been briefly investigated in Chin and Özsoyoğlu [1979].

(4) For a static online SDB environment, data swapping is potentially a suitable security-control method; however, the issues raised in Section 4.2.1 have to be resolved first. Both the probability-distribution method [Liew et al. 1985] and the fixed-data-perturbation method [Traub et al. 1984] could be applied to such an environment once the bias problem has been resolved.

(5) Although cell suppression has been widely used for static offline SDB environment, it suffers from attacks such as the regression-compromise-based method. The random-sample-queries method [Denning 1980] could be applied to such an environment.

(6) The new type of threat discussed in Section 6 raises several issues that no one has yet resolved. We hope that research work in the area of securing SDBs will address these new types of threats.

(7) Finally, to date there is no single security-control method that prevents both exact and partial disclosures. There are, however, few methods (fixed-data perturbation [Traub et al. 1984], fixed-data perturbation for categorical data [Warner 1965], random sample queries [Denning 1980], and varying-output perturbation [Beck 1980]) that prevent exact disclosure and enable the DBA to exercise "statistical-disclosure control." Some of these methods suffer from the bias problem discussed in Section 4.1 and/or the 0 or 1 query-set-size problem (i.e., partial disclosure is possible in case of null query set or a query set of size 1). Dalenius [1977] says (and we agree) that researchers should discard the notion of elimination of both exact and partial disclosures and focus their research efforts on the notion of statistical-disclosure control for the following reasons: "(i) It would be unrealistic to aim at elimination: such a goal is not operationally feasible; (ii) it would place unreasonable restrictions on the kind of statistics that can be released, it may be argued that elimination of disclosure is possible only by elimination of statistics." Thus, we recommend directing future research efforts toward developing new methods that prevent exact disclosure and provide statistical-disclosure control and at the same time do not suffer from the

bias problem and the 0, 1 query-set-size problem. Furthermore, efforts directed toward developing a bias-correction mechanism and solving the general problem of small query-set size would help salvage few of the current perturbation based methods.

ACKNOWLEDGMENTS

John C. Wortmann was a visiting professor at the Graduate School of Management, Rutgers University, when performing the research (January–June 1985).

We would like to thank Professors N. Matloff and M. Palley and three anonymous referees for their invaluable comments and suggestions on earlier drafts of this paper. They helped greatly in improving the quality of the paper.

REFERENCES

ABUL-ELA, A.-L., GREENBERG, B. G., AND HORVITZ, D. G. 1967. A multi-proportions randomized response model. *J. Am. Stat. Assoc. 62*, 319 (Sept.), 990–1008.

ACHUGBUE, J. O., AND CHIN, F. Y. 1979. The effectiveness of output modification by rounding for protection of statistical databases. *INFOR 17*, 3 (Aug.), 209–218.

BECK, L. L. 1980. A security mechanism for statistical databases. *ACM Trans. Database Syst. 5*, 3 (Sept.), 316–338.

CHIN, F. Y. 1978. Security in statistical databases for queries with small counts. *ACM Trans. Database Syst. 3*, 1, 92–104.

CHIN, F. Y., KOSSOWSKI, P., AND LOH, S. C. 1984. Efficient inference control for range sum queries. *Theor. Comput. Sci. 32*, 77–86.

CHIN, F. Y., AND ÖZSOYOĞLU, G. 1982. Auditing and inference control in statistical databases. *IEEE Trans. Softw. Eng. SE-8*, 6 (Apr.), 574–582.

CHIN, F. Y., AND ÖZSOYOĞLU, G. 1981. Statistical database design. *ACM Trans. Database Syst. 6*, 1 (Mar.), 113–139.

CHIN, F. Y., AND ÖZSOYOĞLU, G. 1979. Security in partitioned dynamic statistical databases. In *Proceedings of the IEEE COMPSAC*, pp. 594–601.

COX, L. H. 1980. Suppression methodology and statistical disclosure control. *J. Am. Stat. Assoc. 75*, 370 (June), 377–385.

DALENIUS, T. 1981. A simple procedure for controlled rounding. *Statistik Tidskrift 3*, 202–208.

DALENIUS, T. 1977. Towards a methodology for statistical disclosure control. *Statistik Tidskrift 15*, 429–444.

DALENIUS, T. 1974. The invasion of privacy problem and statistics production. An overview. *Statistik Tidskrift 12*, 213–225.

DENNING, D. E. 1985. Commutative filters for reducing inference threats in multilevel database systems. In *Proceedings of the 1985 Symposium on Security and Privacy, IEEE Computer Society*, pp. 134–146.

DENNING, D. E. 1984. Cryptographic check-sums for multilevel database security. In *Proceedings of the 1984 Symposium on Security and Privacy, IEEE Computer Society*, pp. 52–61.

DENNING, D. E. 1983. A security model for the statistical database problem. In *Proceedings of the 2nd International Workshop on Management*, pp. 1–16.

DENNING, D. E. 1982. *Cryptography and Data Security.* Addison-Wesley, Reading, Mass.

DENNING, D. E. 1981. Restricting queries that might lead to compromise. In *Proceedings of IEEE Symposium on Security and Privacy* (Apr.), pp. 33–40.

DENNING, D. E. 1980. Secure statistical databases with random sample queries. *ACM Trans. Database Syst. 5*, 3 (Sept.), 291–315.

DENNING, D. E., AND SCHLÖRER, J. 1983. Inference control for statistical databases. *Computer 16*, 7 (July), 69–82.

DENNING, D. E., AND SCHLÖRER, J. 1980. A fast procedure for finding a tracker in a statistical database. *ACM Trans. Database Syst. 5*, 1 (Mar.), 88–102.

DENNING, D. E., SCHLÖRER, J., AND WEHRLE, E. 1982. Memoryless inference controls for statistical databases. Computer Science Dept., Purdue Univ.

DENNING, D. E., DENNING, P. J., AND SCHWARTZ, M. D. 1979. The tracker: A threat to statistical database security. *ACM Trans. Database Syst. 4*, 1 (Mar.), 76–96.

DOBKIN, D., JONES, A. K., AND LIPTON, R. J. 1979. Secure databases: Protection against user influence. *ACM Trans. Database Syst. 4*, 1 (Mar.), 97–106.

FELLEGI, I. P. 1972. On the question of statistical confidentiality. *J. Am. Stat. Assoc. 67*, 337 (Mar.), 7–18.

FELLEGI, I. P., AND PHILLIPS, J. L. 1974. Statistical confidentiality: Some theory and applications to data dissemination. *Ann. Ec. Soc. Meas. 3*, 2 (Apr.), 399–409.

FRIEDMAN, A. D., AND HOFFMAN, L. J. 1980. Towards a fail-safe approach to secure databases. In *Proceedings of IEEE Symposium on Security and Privacy* (Apr.).

GHOSH, S. P. 1986. Statistical relational tables for statistical database management. *IEEE Trans. Softw. Eng. SE-12*, 12, 1106–1116.

GHOSH, S. P. 1985. An application of statistical databases in manufacturing testing. *IEEE Trans. Softw. Eng. SE-11*, 7, 591–596.

GHOSH, S. P. 1984. An application of statistical databases in manufacturing testing. In *Proceedings of IEEE COMPDEC Conference.*

GREENBERG, B. G., ABERNATHY, J. R., AND HORVITZ, D. G. 1969a. Application of randomized response technique in obtaining quantitative data. In *Proceedings of Social Statistics Section, America, Statistical Association*, (Aug.), 40–43.

GREENBERG, B. G., ABUL-ELA, A.-L., SIMMONS, W. R., AND HORVITZ, D. G. 1969b. The unrelated question randomized response model: Theoretical framework. *J. Am. Stat. Assoc. 64*, 326 (June), 520–539.

HAQ, M. I. UL. 1977. On safeguarding statistical disclosure by giving approximate answers to queries. In *Proceedings of International Computer Symposium* (North-Holland), pp. 491–495.

HAQ, M. I. UL. 1975. Insuring individual's privacy from statistical database users. In *Proceedings of National Computer Conference* (Montvale, N.J.), vol. 44. AFIPS Press, Arlington, Va., pp. 941–946.

HOFFMAN, L. J. 1977. *Modern Methods for Computer Security and Privacy.* Prentice-Hall, Englewood Cliffs, N.J.

HOFFMAN, L. J., AND MILLER, W. F. 1970. Getting a personal dossier from a statistical data bank. *Datamation 16*, 5 (May), 74–75.

JONGE, W. DE 1983. Compromising statistical databases: Responding to queries about means. *ACM Trans. Database Syst. 8*, 1 (Mar.), 60–80.

KAM, J. B., AND ULLMAN, J. D. 1977. A model of statistical databases and their security. *ACM Trans. Database Syst. 2*, 1, 1–10.

LEFONS, D., SILVESTRI, A., AND TANGORRA, F. 1983. An analytic approach to statistical databases. In *Proceedings of 9th Conference on Very Large Databases* (Florence, Italy), pp. 260–273.

LEISS, E. 1982. Randomizing a practical method for protecting statistical databases against compromise. In *Proceedings of 8th Conference on Very Large Databases*, pp. 189–196.

LIEW, C. K., CHOI, W. J., AND LIEW, C. J. 1985. A data distortion by probability distribution. *ACM Trans. Database Syst. 10*, 3, 395–411.

MATLOFF, N. E. 1986. Another look at the use of noise addition for database security. In *Proceedings of IEEE Symposium on Security and Privacy*, pp. 173–180.

MCLEISH, M. 1983. An information theoretic approach to statistical databases and their security: A preliminary report. In *Proceedings of the 2nd International Workshop on Statistical Database Management*, pp. 355–359.

MILLER, A. R. 1971. *The Assault on Privacy-Computers, Data Banks and Dossiers.* University of Michigan Press, Ann Arbor, Mich.

MORGENSTERN, M. 1987. Security and Inference in Multi-level Database and Knowledge-Bare Systems. In *Proceedings of ACM Special Interest Group on Management of Data*, pp. 357–373.

ÖZSOYOĞLU, G., AND CHIN, F. Y. 1982. Enhancing the security of statistical databases with a question-answering system and a kernel design. *IEEE Trans. Softw. Eng. SE-8*, 3, 223–234.

ÖZSOYOĞLU, G., AND CHUNG, J. 1986. Information loss in the lattice model of summary tables due to cell suppression. In *Proceedings of IEEE Symposium on Security and Privacy*, pp. 75–83.

ÖZSOYOĞLU, G., AND ÖZSOYOĞLU, M. 1981. Update handling techniques in statistical databases. In *Proceedings of the 1st LBL Workshop on Statistical Database Management* (Berkeley, Calif., Dec.), pp. 249–284.

ÖZSOYOĞLU, G., AND SU, T. A. 1985. Rounding and inference control in conceptual models for statistical databases. In *Proceedings of IEEE Symposium on Security and Privacy*, pp. 160–173.

PALLEY, M. A. 1986. Security of statistical databases compromise through attribute correlational modeling. In *Proceedings of IEEE Conference on Data Engineering*, pp. 67–74.

PALLEY, M. A., AND SIMONOFF, J. S. 1987. The use of regression methodology for compromise of confidential information in statistical databases. *ACM Trans. Database Syst. 12*, 4 (Dec.), 593–608.

REISS, J. P. 1980. Practical data-swapping: The first steps. In *Proceedings of IEEE Symposium on Security and Privacy*, pp. 36–44.

REISS, S. P. 1984. Practical data swapping: The first steps. *ACM Trans. Database Syst. 9*, 1 (Mar.), 20–37.

ROWE, N. 1984. Diophantine inference from statistical aggregates on few-valued attributes. In *Proceedings of IEEE Conference on Data Engineering*, pp. 107–110.

SANDE, G. 1983. Automated cell suppression to reserve confidentiality of business statistics. In *Proceedings of the 2nd International Workshop on Statistical Database Management*, pp. 346–353.

SCHLÖRER, J. 1983. Information loss in partitioned statistical databases. *Comput. J. 26*, 3, 218–223.

SCHLÖRER, J. 1981. Security of statistical databases: multidimensional transformation. *ACM Trans. Database Syst. 6*, 1 (Mar.), 95–112.

SCHLÖRER, J. 1980. Disclosure from statistical databases: Quantitative aspects of trackers. *ACM Trans. Database Syst. 5*, 4 (Dec.), 467–492.

SCHLÖRER, J. 1976. Confidentiality of statistical records: A threat monitoring scheme of on-line dialogue. *Methods Inform. Med. 15*, 1, 36–42.

SCHLÖRER, J. 1975. Identification and retrieval of personal records from a statistical data bank. *Methods Info. Med. 14*, 1, 7–13.

SCHWARTZ, M. D., DENNING, D. E., AND DENNING, P. J. 1979. Linear queries in statistical databases. *ACM Trans. Database Syst. 4*, 2, 156–167.

SU, T., AND ÖZSOYOĞLU, G. 1987. Data dependencies and inference control in multilevel relational database systems. In *Proceedings of the 1987 Symposium on Security and Privacy, IEEE Computer Society*, pp. 202–211.

TENDICK, P., AND MATLOFF, N. S. 1987. Recent results on the noise addition method for database security. Presented at the Joint ASA/IMS Statistical Meetings, San Francisco.

TRAUB, J. F., YEMINI, Y., AND WOZNIAKOWSKI, H. 1984. The statistical security of a statistical database. *ACM Trans. Database Syst. 9*, 4 (Dec.), 672–679.

TRUEBLOOD, R. P. 1984. Security issues in knowledge systems. In *Proceedings of 1st International Workshop on Expert Database Systems*, vol. 2, pp. 834–840.

TURN, R., AND SHAPIRO, N. Z. 1978. Privacy and security in databank systems: Measure of effectiveness, costs, and protector-intruder interactions. *Computers and Security*, C. T. Dinardo, Ed. AFIPS Press, Arlington, Va., pp. 49–57.

WARNER, S. L. 1971. The linear randomized response model. *J. Am. Stat. Assoc. 66*, 336 (Dec.), 884–888.

WARNER, S. L. 1965. Randomized response: A survey technique for eliminating evasive answer bias. *J. Am. Stat. Assoc. 60*, 309 (Mar.), 63–69.

YU, C. T., AND CHIN, F. Y. 1977. A study on the protection of statistical databases. In *Proceedings of ACM SIGMOD International Conference on Management of Data* (Aug.), pp. 169–181.

Received January 1987; final revision accepted November 1988.