Journal of Official Statistics, Vol. 14, No. 4, 1998, pp. 347–360

A Database System Prototype for Remote Access to Information Based on Confidential Data[1]

Sallie Keller-McNulty[2] and Elizabeth A. Unger[3]

U.S. federal statistical agencies[4] collect data under the protection of confidentiality. The public expects timely and equitable dissemination of information contained in this data. There is also a demand for access to the data from federal agencies for legitimate research purposes. Thus, there exists a tension between data access and data confidentiality. The computer science community in harmony with the statistics community needs to aggressively develop new data access procedures that maintain data confidentiality in ways that are consistent with federal policy. This article discusses the development of a prototype for a remote data access system that releases useful analytical results while protecting the confidentiality of the data. The prototype data access system combines computer science database technology with statistical data disclosure limiting techniques. These results provide evidence that the amalgamation of computer technology with statistical methodologies should be able to provide confidentiality protection of data while enhancing the quantity and quality of the released analytical results.

Key words: Data access; disclosure limitation; computer security; data confidentiality; database security.

1. Introduction

Statistical agencies collect data on individuals and organizations. These agencies are tasked with the dual assignments of collecting data under the protection of confidentiality and disseminating information contained in the data to the public. With our society's movement into the information age, people are becoming more aware of how much information is being collected and how little is being disseminated. There is also a significant demand for access to statistical agency data for legitimate research purposes (Duncan, Jabine, and de Wolf 1993; Duncan and Pearson 1991). This tension between data access and data confidentiality requires the agencies to aggressively seek out new data access procedures for the dissemination of information derived from sensitive data.

This article discusses the development of a prototype for a remote data access system designed to release useful analytical results while protecting the confidentiality of the data. The remote access system combines computer science data access controls with

[1] This research has been funded in part by a grant from the National Computer Security Center (R2-91-8012) and by a Joint Statistical Agreement with the U.S. Census Bureau (JFASA91-26).
[2] Statistical Sciences Group, Los Alamos National Laboratory, MSF600, Los Alamos, NM 87545, U.S.A.
[3] Department of Computer and Information Sciences, Kansas State University, Manhattan, KS 66506, U.S.A.
[4] These include, but are not limited to, U.S. agencies such as the Census Bureau, Bureau of Labor Statistics, National Center for Education Statistics, Internal Revenue Service, and National Center for Health Statistics.

statistical data disclosure limiting techniques. The successful design and implementation of the prototype provides evidence that the amalgamation of computer technology with statistical methodologies can provide confidentiality protection of data while enhancing the quantity and quality of the released analytical results. The importance of designing and implementing systems such as the one developed here goes well beyond providing a technological solution regarding the release of information derived from confidential data. Such systems could serve as laboratories to explore the tradeoffs between different approaches to disclosure limitation and to examine the role of the intruder who tries to exploit the system.

The following section provides some technical background and definitions that play an important role in the design of the data access system. The prototype system is described in Section 3. The testing of the implemented prototype is discussed in Section 4 and some conclusions are given in Section 5.

2. Background Definitions

This section briefly discusses five important concepts that require precise definition during the development of an operational data access system that maintains data confidentiality. These concepts are:

data – what data will be stored and in what format;
disclosure – what will be designated a disclosure;
access – who has access, how access will be granted, and what analytical uses are allowed;
attack – who, how, and what external information may be known by the attacker;
data usefulness versus risk – what level of data distortion results from maintaining an acceptable level of risk of disclosure.

A more complete discussion of these topics can be found in Fienberg 1997 and Keller-McNulty and Unger 1993.

2.1. Data

Data refers to some representation of information organized for analysis and decision making. One frequently thinks of data as numbers representing measurements, but data should also be thought of as the relationships between those measurements and between clusters of those measurements. A *database* system (or simply database) consists of *records*. The semantics or meaning and format or layout of the records is described by the *metadata*. The *fields* in the records represent *attributes*, i.e., variables. The value of a field on a given record is the *attribute value*.

Data or a database can be either *static* or *dynamic*. Static data remain constant over time, or at least over the time interval during which the security of the data is in question. Dynamic data change over time. Dynamic databases, e.g., databases created to store time series and longitudinal data, may experience changes in attribute values, in the relationships among attributes, and in the number of records as the database is modified through updates, insertions, and deletions.

2.2. Disclosure

The data contained in a database system can be thought of as describing some group, e.g., population of entities (individuals and/or organizations). A *disclosure* results if one or both of the following two conditions occur as a consequence of releasing the data:

i) a specific population entity is linked to one or more data records;
ii) confidential or sensitive information about this entity is learned.

The linkage in the first condition can be thought of as a positive or negative link. A positive link identifies a set of one or more data records as belonging to the entity. A negative link confirms a disassociation of an entity with a set of data records.

2.3. Access

Within a database environment, computer scientists describe data security as protecting data through access controls from unauthorized disclosure (secrecy), from unauthorized modifications (integrity), and from the denial of authorized access (availability). This is usually accomplished by controlling access to different parts of the database through the use of passwords and data classification schemes. These types of access controls are only capable of controlling direct access to the data but not the analytical procedures that may be performed with the data.

To meet the data confidentiality needs of federal statistical agencies, additional access controls are essential. The necessary control may be gained through the use of *statistical databases* and *audit trails*. A statistical database is defined in the computer database literature as a database that only releases predefined aggregate statistical analyses (Adam and Wortmann 1989; Denning 1980). An audit trail is a data stream that can be used to automatically monitor the data and analysis requests of a user to determine if the requests are within the user's activity profile (Lunt 1988).

2.4. Attack

Data is compromised through *attacks*. An *attacker* (or *intruder*) is someone who exploits the data or the database for reasons other than intended by the data custodian. Attacks can be both intentional and unintentional. The general focus in the literature has been on intentional attacks, with the idea that unintentional attacks can be eliminated through access controls. In practice the attacker is frequently a legitimate user with various degrees of authorized access to the data (Beck 1980; Denning, Denning, and Schwartz 1979).

Methods of attack can be broadly classified into two groups on the basis of the types of requests for data, i.e., queries, used to compromise the data. A user may query a database using either direct queries or aggregate function queries. A *direct query* returns microdata values while an *aggregate function query* returns some aggregate value such as a mean, count, total, regression analysis or other statistical analysis. A *direct attack* occurs when the attacker attempts to seek sensitive attribute values through direct queries. An *indirect attack* occurs when the attacker attempts to seek sensitive attribute values by inferring them from statistical results.

Direct attacks are assumed to be controlled by restricting access to data to trusted users.

Indirect attacks have thus far proved nearly impossible to completely eliminate without rendering the data useless for analytical purposes. Indirect attacks have been the focus of what is called the *inference problem* (Keller-McNulty and Unger 1993; Pfleeger 1989). Formally the inference problem is defined as the deduction of confidential data from nonsensitive data. To make this deduction the attacker combines external information with the data she/he has access to in the database. This is depicted in Figure 1.

Indirect attacks can further be classified into two broad categories, *set manipulation* and *statistical inference* attacks. Set manipulation attacks are the result of constructing unions, intersections, complements, and differences of sets of data such that the result is a disclosure. Attacks made using statistical inference are the result of statistically manipulating accessible (nonsensitive) data to improve the intruder's knowledge about the sensitive data.

Both forms of indirect attack require the attacker to have some prior knowledge about the specific entity(ies) sought (Adam and Wortmann 1989). To accomplish a disclosure using the set manipulation principle, the intruder would need to know which set of attributes uniquely identifies the population entity(ies). Using statistical inference techniques, an intruder may compromise the database by using some subjective prior knowledge about the entity of interest and, through sequential access to the database, empirically revise this information.

The set manipulation form of attack has been associated with attempts to compromise single attribute values for entities in the database. This method can be thought of as a direct data reduction scheme and has typically been applied to data released in tabular form or as sequential aggregate function queries to a database (Cox 1980 and 1981; Beck 1980). Set manipulation attacks are characterized mathematically as linear systems of equations.

The *tracker*, which is a tool for constructing a set manipulation attack, is perhaps the most studied method of attack (Beck 1980; Denning 1980 and 1982; Denning and Schlörer 1983; Duncan and Muhkerjee 1991; Keller-McNulty and Unger 1990; Noren and Keller-McNulty 1993; Schlörer 1976 and 1980). A tracker is a logical formula T such that the query $q(T)$ is answerable under the database system's data security controls. For example, $q(T)$ could be a request for the sum of some attribute, e.g., salary, over all the records in the database defined by the logical formula, T, e.g., $T=$ (sex $=$ male AND rank $=$ professor).

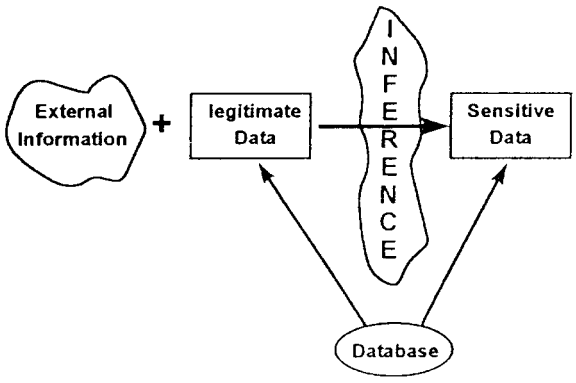

Fig. 1. Inference flow for external information

In one form of tracker attacks, the tracker is selected such that the queries $q(C \text{ or } T)$, $q(C \text{ or NOT } T)$, $q(\text{NOT } T)$ and $q(T)$ are answerable, while the query $q(C)$ is unanswerable. The query $q(C)$ is not answerable because the set C is too small or too large for the data security controls to allow the retrieval of the information pertaining to the entities characterized by C. The logical formula C is called the characterizing formula. Using the tracker T the unanswerable and sensitive query $q(C)$ is obtained through the following manipulation of four answerable queries:

$$q(C) = q(C \text{ or } T) + q(C \text{ or NOT } T) - [q(T) + q(\text{NOT } T)] \tag{1}$$

This specific tool is used to compromise the prototype system described in Sections 3 and 4.

2.5. Disclosure risk and data usefulness

A measure of how vulnerable the data are to disclosure is called *disclosure risk*. Considerable work has been done on estimating disclosure risk for tabular data (Bethlehem, Keller, and Pannekoek 1990; Blien, Wirth, and Müller 1992; Chen and Keller-McNulty 1998; Greenberg and Voshell 1990; Greenberg and Zayatz 1992; Keller and Bethlehem 1992; Mokken et al. 1992; de Waal and Willenborg, 1996; Zayatz 1991). This work measures how vulnerable the data are to disclosure by estimating, from a sample, the number of unique population entities in the data vis-à-vis a k-dimensional cross-classification of the population. It is assumed in these models that the intruder knows which set of attributes uniquely identifies an entity. This work is directly applicable to the measurement of disclosure risk in a sequential query access environment because each query defines a subpopulation or cell from a multidimensional table.

The models just discussed give estimates of the per cent of population at risk if a full, unconstrained set of data were released. A more general definition of disclosure risk is a measure of the effectiveness of a data confidentiality procedure against specific forms of attack (Duncan and Lambert 1989; Fienberg et al. 1997; Paass 1988). Data confidentiality procedures are characterized as providing *restricted data* and/or *restricted access* (Duncan, Jabine, and de Wolf 1993). Restricting data implies controlling the content of the released data through methods such as removing explicit identifiers and masking the sensitive data, e.g., grouping into categories. Restricting access imposes conditions on who has access to all or part of the data.

The effect of the data access and/or data restrictions on the usefulness of the data to the analyst has received little integrated attention in the literature. Many output perturbation techniques, including masking, imputation, sampling, controlled rounding, topcoding, and swapping, have been developed in an effort to maintain some of the data's original structure. Techniques such as grouping, blanking, cell suppression, and query set size restrictions have been developed to restrict access to sensitive data. The integration of these different types of techniques to provide both protection against various forms of attack and useful data has not been adequately studied.

3. A Mediator Model for Secure Data Access

The prototype research focused on the feasibility of storing and accessing data like that

collected by a federal statistical agency in a database system with both database security and statistical disclosure limiting controls. This integrated database system is more appropriately called an information system. It was assumed that data collected by federal statistical agencies would have many categorical attributes (variables) that could be used to form different levels of aggregation and that the attacks on the data would link entities (individuals or organizations) in the data through these attributes. It was also assumed that the sensitive information would be represented either as categorical or as continuous variables and only aggregate function queries would be allowed, i.e., statistical database access. With these assumptions in place it becomes obvious that this research could be applicable to many other information systems, including administrative databases and databases used by private business.

The model developed here places a software layer, called a *mediator*, between the user and the information system. The mediator interfaces the user to the database and mediates all access by the users to the statistical database. The mediator concept is illustrated in Figure 2. Each user has access to the database and metadata only through the mediator. The database and the corresponding metadata may be stored in a data dictionary, schema, or knowledge-based system. The mechanism for extracting the data from the database is an aggregate function query. A data acquisition request can pass iteratively from the *metadata/knowledge base* layer to the mediator, collecting additional data as needed for decision making. Decisions concerning the release of the data are made at the metadata/knowledge base layer. The mediator removes the user one step from the information system. This allows for queries to be parsed and reconstructed such that a useable form of the information can be passed to the information system and used to create a user response. Additional information about the query and/or the user such as the user's external knowledge related to the query or the user's access profile would be contained in the mediator and passed to the information system as needed.

The mediator concept was selected for three fundamental reasons. First, it was important to create a system that could be used for further experimentation with models based on a combination of assumed attack and corresponding deterrent methods. The mediator design allows easy replacement of modules that encode the information about forms of attack, including auditing user access and merging external data sources accessible to users, and modules that encode risk estimation methods, including agency

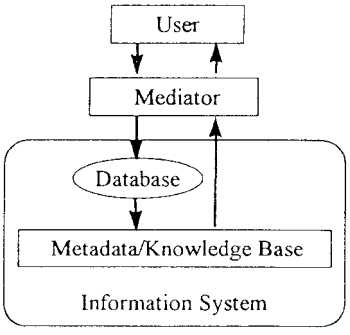

Fig. 2. Conceptual view of a mediator

specific data confidentiality policies and parameters. Second, it was important to determine if the concept of a non-invasive protection system could be developed and implemented for use with existing database management systems. This would allow the creation of a product that, once integrated with the user interface of the database management system, could provide inferential protection in existing database environments. Finally, the placement of the mediator outside the information system allows for the natural integration of a firewall between the external user and the agency protected data.

3.1. The prototype

This research set out to demonstrate that operational access and data restriction methods could be defined and implemented into a sequential access database system to provide data confidentiality protection. The research successfully designed, implemented and tested a prototype system based on the conceptual model in Figure 2. The implemented prototype data access system disseminates analytical results from sequential queries, where decisions about the access to the data are made at query time and on a query-by-query basis. The focus of the discussion in this section is on the design of that data access system model rather than on the computational details of the implementation. See Rogers and Unger (1993) for the computational details and the code.

The prototype was implemented using the database management system POSTGRES (Stonebraker and Rowe 1986). This relational database model uses a typical logic-based query language. POSTGRES was chosen because it is a public domain database management system available in source code.

The normal access for the user in a POSTGRES database system is through a set of codes, called the *monitor*, that interprets the user's query. The monitor interacts with

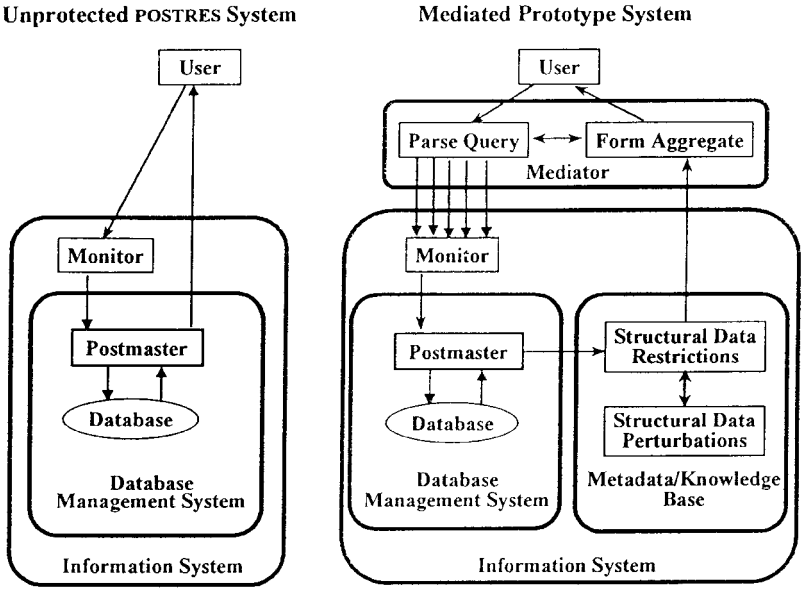

Fig. 3. Architectures of unprotected POSTGRES system and mediated prototype system

the database through a system module called the *postmaster*. The postmaster in POSTGRES is the mechanism that accesses the records in the physical database and performs the requested data manipulations. In a traditional POSTGRES environment, the postmaster retrieves as a set and aggregates all of the records that match the selection condition of the query.

The architecture of the original or ''unprotected'' POSTGRES database management system is given in the left panel of Figure 3. As a simple example, consider the request to form an average of a variable (attribute) X for those records in which attribute Y has a value of less than 30. Assume that both X and Y appear in the same microdata record. That is the query request can be satisfied from one file, rather than through joining several files. The monitor instructs the postmaster to retrieve and aggregate X over the entire set of microdata records (where Y has a value less than 30) and return the result to the user.

The prototype system uses the architectural design given in the right panel of Figure 3. The difference between the mediated prototype and the unprotected POSTGRES system is the addition of the mediator and the metadata/knowledge base. The mediated prototype system needed to be able to invoke data access and data restriction rules. Implementation of such rules requires access to the microdata record-by-record. Therefore, the system had to be modified to collect a set of microdata records, as opposed to direct computation of an aggregate. It was important that the required system modifications be minimally invasive to the database management system architecture, thus leaving the functionality of the monitor and the postmaster intact. This was accomplished by designing the mediator to first parse the aggregate query request into clauses of the logic based query and then issue a set of direct queries (i.e., requests for single microdata values) to the monitor. As in the unprotected system, the monitor instructs the postmaster to retrieve the data and form the aggregates, which are now aggregates of size one. The postmaster then sends the results to the metadata/knowledge base, rather than sending them directly back to the user. Once the set of direct queries has been sent to the monitor and data retrieved by the postmaster, the entire set of microdata records are sitting in the metadata/knowledge base in readiness for the execution of the data access and restriction methodologies.

It is in the metadata/knowledge base layer of the system that the access and data restriction rules are implemented. Here a disclosure risk assessment is done followed by the appropriate data suppression, perturbation, and manipulation. The ''modified'' data are sent back to the mediator for the calculation of the final result. The mediated protype system depicted in Figure 3 shows two levels of data security, *structural data restrictions* and *statistical data perturbation*, within the metadata/knowledge base layer. The structural data restriction level is where rules that restrict access to various entities would be implemented, e.g., blanking cells in a table, minimum query set size checks, etc. The statistical data perturbation level is where the microdata would be statistically altered or masked in various ways. It is important to note that these levels are designed to communicate with each other.

In principle, the final data modification may require iterative passes through the mediator to the database management system to the metadata/knowledge base, linking additional/external information to the query request as needed for the disclosure risk assessment. (Only a single pass through the system was implemented in the prototype.)

Collectively this could be viewed as simple output perturbation of the microdata prior to the compilation of the result. Here this output perturbation is invoked on a query-by-query basis and implemented such that the database maintains its integrity. This means, for static databases, that all database responses for the specific query sets must match, regardless of when, by whom, or how the requests are made. It is important to remember that the true data itself sits within the system.

3.2. Prototype data restrictions and data perturbation methods

The goal of the prototype system implementation was to demonstrate that such systems could be developed and made operational and to identify the decisions that need to be made in creating a mediated database system as shown in Figure 3. In addition to providing access to confidential data, an important use of these systems is for experimentation and evaluation of competing disclosure risk methodologies. The prototype described here has not yet been used to evaluate specific data restriction and data perturbation methods.

In a sequential access database system, query selection is typically based on categorical attributes which define some subset of the database, i.e., a cell or set of cells from a k-dimensional cross tabulation of the data. In the mediated prototype model (see Figure 3), the structural data restrictions level is designed to contain access restriction rules based on categorical attribute values. The access restriction rules could be based on disclosure risk measures that determine the number (or per cent) of unique database entities in the k-dimensional table defined by the attributes in the query. Minimum query set size restriction falls in this category in that it identifies whether the requested cell is of an acceptable size. A more informative statistical model that estimates, in real-time, the risk of an entire population due to the release of a k-dimensional decomposition of the database defined by a query could also be used (Chen and Keller-McNulty 1998; Skinner 1992; Skinner et al. 1994). Note that the database may be the entire population of some sample from the population.

Only the minimum query set size restriction has been implemented in the prototype. This was chosen as a first step because it is the standard computer security technique used to reduce the threat of tracker attacks, the form of attack studied with the prototype. Therefore, the first decision the system makes regarding the query response is whether or not to return a response. If the query set size is too small or too large (e.g., size one or population size -1), no response is returned. It is well documented that minimum query set size controls alone will not eliminate tracker attacks. This presents a perfect opportunity to couple some statistical disclosure limiting techniques (e.g., data perturbation methods) with a simple database security control in an attempt to further reduce the threat of tracker attacks. Therefore, if the query set size passes this check, then the prototype invokes some form of data perturbation.

The statistical data perturbation level in Figure 3 was designed to contain data restriction techniques that alter the attribute values in some way. The methods implemented in the prototype include simple random noise addition, random noise addition as a function of aggregate query statistics, and two variations of controlled rounding. For more information regarding the statistical and mathematical detail of the specific implemented methods, the reader is referred to Noren (1993) and Noren and Keller-McNulty (1993).

One example of a data perturbation method implemented in the prototype is the following. For answerable queries, (i.e., queries with query set sizes exceeding some minimum value), each record, y_i, in a query set G is imputed as

$$y_i^* = y_i + X_i H_i \bar{y}_G \tag{2}$$

where $\bar{y}_G$ is the mean of the query set, X_i and H_i are independent random variables distributed as

$$X_i = \begin{cases} 1 & \text{with probability } p_1 \\ -1 & \text{with probability } p_2 \\ 0 & \text{with probability } 1 - p_1 - p_2 \end{cases} \quad \text{and } H_i \sim \text{Uniform}(U, L), \text{ for } 0 \le L \le U \le 1.$$

The aggregate function requested for the query set is denoted as $q(G) = q(y_i^* : y_i \in G)$. It is straightforward to find $E[q(G)]$ and $Var[q(G)]$ for various types of aggregate query functions $q()$. Suppose $q() =$ average and $N_G =$ the number of records in G. Then,

$$q(G) = \frac{1}{N_G} \sum_{y_i \in G} y_i^*$$

$$E(q(G)) = \bar{y}_G + N_G \bar{y}_G \left(\frac{U + L}{2} \right) (p_1 - p_2) \text{ and}$$

$$Var(q(G)) = \frac{\bar{y}_G^2}{12 N_G} [4(p_1 + p_2)(U^2 + UL + L^2) - 3(p_1 - p_2)^2 (U + L)^2]$$

This implies that the perturbation method can give biased responses where the degree of bias and uncertainty in the responses are functions of the perturbation parameters p_1, p_2, L, and U. Evaluation of $E[q(C)]$ and $Var[q(C)]$, where $q(C)$ is the sum of the four queries making up a tracker attack in Equation (1), is also straightforward. These expectations again depend on the values of p_1, p_2, L, and U. Since it takes four queries to obtain $q(C)$, the cumulative bias and variance associated with $q(C)$ is larger than for a single legitimate query.

This example illustrates how each of the perturbation techniques in the metadata/knowledge base requires the data custodian to set various parameters. These parameter settings result in different degrees of ''correctness'' of the legitimate aggregate query results and different degrees of ''distortion'' of results for an intruder issuing a tracker attack.

The prototype system was implemented with integrity constraints that forced the system to return the same response each time a specific query set was requested. Note that many different orderings of logical formulas based on the categorizing attributes that make up a query request can lead to the same query set (e.g., **A** AND **B** OR **D** versus (**D** OR **B**) AND (**D** OR **C**)). In the prototype, once the query set is identified, the mediator issues a set of direct queries to the monitor that is independent of the ordering of the logic operators or attributes in the original query request. Every time the same query set is requested, the set of records must be selected in the same order and the same perturbation added to each record. The record order is accomplished by forcing the database to spin to the top before the postmaster begins selecting records that match the set of direct query requests. Then, by seeding the random number generators with deterministic functions of the query set

size, the same sequence of data perturbations is created, resulting in the same set of imputed records each time. This methodology also eliminates an intruder's ability to issue the same query multiple times and average out the perturbation.

4. Test Case

The specific data used in the prototype was a 7,000 record administrative student file. The allowable aggregate functions consisted of basic descriptive statistics: means, totals, counts, and variances. Attribute combinations, or characterizing formulas, that lead to unique cells in this test data were identified for one- to seven-way tables. Since the database contained the entire population of interest, cells of size one corresponded to unique population entities. It is interesting to note that over one-third of the students could be uniquely identified in a five-way table of relatively common variables (sex, marital status, ethnic origin, class, college). The corresponding characterizing formulas for the unique cells were electronically stored so they could be used to attack the database once the prototype was implemented. This resulted in approximately 6,000 characterizing formulas C that uniquely identified a record. The attack on the data was limited to the tracker attack discussed in Section 3.4. Therefore, approximately $4 \times 6{,}000 = 24{,}000$ individual logical formulas, not all unique, were used in forming the aggregate query requests needed to attack the system.

The effectiveness of the implemented access and data restriction methods with regard to preventing exact disclosures via trackers was empirically and theoretically evaluated. The distortion of the results for single legitimate queries due to the data restriction methods was also evaluated. This was done by sending the set of stored characterizing formulas through the prototype and collecting the perturbed query results. For all the implemented methods, the empirical results were compared against their corresponding theoretical statistical expectations for the various parameter settings. For example, using the perturbation method given in Equation 2 with $p_1 = .05$, $p_2 = .1$, $L = 0.02$, and $H = 0.08$, legitimate queries for the mean grade point average (GPA) of male students were checked to see if they fell in the range $(2.53, 2.54) = E(q(G)) \pm 2\sqrt{Var(q(G)}$, and the corresponding tracker results were checked to see if they fell in the range $(-9.27, 14.23,) = E(q(C)) \pm 2\sqrt{Var(q(C))}$. The true average GPA for males was 2.471, showing that legitimate queries would have a slight upward bias with these parameter settings.

Recall that the system was designed to generate the same perturbation sequence and give the same response for the same query set, regardless of when, how, or how often it is requested. Therefore, to specifically test the system's accuracy for legitimate queries, the entire system was restarted to obtain new perturbation sequences for the set of test queries. After several repetitions of this process it became clear that the prototype results were not within acceptable ranges. The problem was with our initial method of reseeding the random number generator for each new query request. The random number generator used was the C function **errand( )**, which was found to cycle in a non-random manner about every ten seed values. We were issuing thousands of queries, equating to thousands of reseedings of **errand( )**. To resolve this problem, a large table of random numbers was generated based on a single seeding of **errand( )** and then perturbation assignments were completed using a table look-up method where the query set size was used as the starting pointer to the table.

Once the random number generation problem was resolved, the query results from the prototype fell well within the ranges expected on the basis of the theoretical expectations and variances. This implied that the prototype did not have any serious implementation flaws and we had obtained proof of concept from statistical disclosure and database management system architectural points of view. Users could sequentially query the system using a logic based query language. Answerable queries returned aggregate functions falling within the statistically acceptable levels, yet perturbed enough such that an attacker could no longer isolate single attribute values.

The solution architecture of the prototype inserted procedures between the user and the database management system, (i.e., the mediator and the metadata/knowledge base), to change the data values retrieved by the system. This was done in a non-invasive way. The mediator presented not one, but many queries to the monitor for each statistical database request. During the initial design stage of the system a question arose as to whether this could be accomplished without changing the architecture of the database management system. The success of the prototype proved the answer was yes for a typical relational database system. The implication of such a finding is that security for statistical database systems that involve both computer security and data disclosure methodologies can be implemented relatively easily. In addition, the mediator architecture allows for replacement or addition of modules as security needs change.

4. Conclusions

This work has demonstrated that a mediator in conjunction with an information system can be used to protect the confidentiality of data accessed via a statistical database. The *data* used were a 7,000 record student data file stored in a POSTGRES database. The *disclosure* definition was an exact-positive disclosure of a single attribute value. The *access* was restricted to sequential requests for aggregate statistics such as a count, total, mean, or standard deviation. The *attack* was assumed to be a general tracker attack. The *disclosure risk* was a measure of query set size. The *data restriction methods* were a combination of minimum query set size control and output perturbation controls. The measure of effectiveness of the data restriction and access restriction methods against a tracker attack was computed. The *measure of usefulness* of the released data was limited to evaluation of the effects of the data perturbation on the computation of means and totals for legitimate queries.

The prototype functioned well in that it communicated in a non-invasive way with the database, knowledge base, and database management system. It was able to mediate access while performing the required statistical analyses and returning these results to the user. The modules that invoked the data restrictions and access restrictions ranged in computational complexity up to $\log(n)$. Therefore, the prototype ran noticeably slower than an unprotected POSTGRES database. This computational complexity would exist in any computational environment that used these methods. The mediator effect on computational complexity was constant. Thus, the mediator concept seems an efficient, reasonable and flexible approach to providing data confidentiality protection.

For all the perturbation methods implemented in the prototype, experimentation on the system was not necessary for evaluating the degree of distortion caused by the access and perturbation rules for various parameter settings. Those evaluations could be easily done

through deriving the statistical properties of the aggregate functions based on the imputed data, as was demonstrated with the example in Equation 2. The experimentation here was used to check the implementation. For more complex data perturbation schemes, more sophisticated aggregate functions, and non-trivial data restriction rules, the theoretical derivations of the statistical properties for the results may not be feasible and system experimentation will be critical.

5. References

Adam, N.R. and Wortmann, J.C. (1989). Security-Control Methods for Statistical Databases: A Comparative Study. ACM Computing Surveys, 21, 515–556.

Beck, L.L. (1980). A Security Mechanism for Statistical Databases. ACM Transactions on Database Systems, 5, 316–338.

Bethlehem, J.G., Keller, W.J., and Pannekoek, J. (1990). Disclosure Control of Microdata. Journal of the American Statistical Association, 895, 38–45.

Blien, U., Wirth, H., and Müller, M. (1992). Disclosure Risk for Microdata Stemming from Official Statistics. Statistica Neerlandica, 46, 69–82.

Chen, G. and Keller-McNulty, S. (1998). Estimation of Identification Disclosure Risk in Microdata. Journal of Official Statistics, 14, 79–95.

Cox, L.H. (1980). Suppression Methodology and Statistical Disclosure Control. Journal of Statistical Planning and Inference, 82, 152–164.

Cox, L.H. (1981). Linear Sensitivity Measures in Statistical Disclosure Control. Journal of Statistical Planning and Inference, 5, 152–164.

de Waal, A.G. and Willenborg, L.C.R.J. (1996). A View of Statistical Disclosure Control for Microdata. Survey Methodology, 22, 95–103.

Dalenius, T. (1977). Towards a Methodology for Statistical Disclosure Control. Statistisk tidskrift, 5, 429–444.

Denning, D.E. (1980). Secure Statistical Databases with Random Sample Queries. ACM Transactions on Database Systems, 5, 291–315.

Denning, D.E. (1982). Cryptography and Data Security. Addison-Wesley, Reading, Massachusetts.

Denning, D.E., Denning, P.J., and Schwartz, M.D. (1979). The Tracker: A Threat to Statistical Database Security. ACM Transactions on Database Systems, 4, 76–96.

Denning, D.E. and Schlörer, J. (1983). Inference Controls for Statistical Databases. Computer, 16, 69–82.

Duncan, G.T., Jabine, T.B., and de Wolf, V.A. (1993). Private Lives and Public Policies: Confidentiality and Accessibility of Government Statistics. National Academy Press, Washington, DC.

Duncan, G. and Lambert, D. (1989). The Risk of Disclosure for Microdata. Journal of Business and Economic Statistics, 7, 207–217.

Duncan, G. and Mukherjee, S. (1991). Microdata Disclosure Limitation in Statistical Databases: Query Size and Random Sample Query Control. Proceedings of the 1991 IEEE Symposium on Security and Privacy, 278–287.

Duncan, G. and Pearson, R. (1991). Enhancing Access to Data While Protecting Confidentiality. Statistical Science, 6, 217–239.

Fienberg, S.E. (1997). Confidentiality and Disclosure Limitation Methodology:

Challenges for National Statistics and Statistical Research. Paper commissioned by the Committee on National Statistics.

Fienberg, S.E., Makov, U.E., and Sanil, A.P. (1997). A Bayesian Approach to Data Disclosure. Optimal Intruder Behavior for Continuous Data. Journal of Official Statistics, 13, 75–89.

Greenberg, B.G. and Voshell, L.V. (1990). Relating Risk of Disclosure for Microdata and Geographic Area Size. Proceedings of the Section on Survey Research Methods, American Statistical Association, 450–455.

Greenberg, B.G. and Zayatz, L.V. (1992). Measuring Risk in Public Use Microdata Files. Statistica Neerlandica, 46, 33–48.

Keller, W.J. and Bethlehem, J.G. (1992). Disclosure Protection of Microdata: Problems and Solutions. Statistica Neerlandica, 46, 5–19.

Keller-McNulty, S. and Unger, E.A. (1990). The Deterrent Value of Natural Change in a Statistical Database. Proceedings of the Section on Statistical Computing, American Statistical Association, 15–23.

Keller-McNulty, S. and Unger, E.A. (1993). Database Systems: Inferential Security. Journal of Official Statistics, 9, 475–500.

Lunt, T. F. (1988). Automated Audit Trail Analysis and Intrusion Detection: A Survey. Proceedings of the 11th National Computer Security Conference, Baltimore, MD, U.S.A.

Mokken, R.J., Kooiman, P., Pannekoek, J., and Willenborg, L.C.R.J. (1992). Disclosure Risks for Microdata. Statistica Neerlandica, 46, 49–68.

Noren, E. (1993). The Protection of Confidential Data Stored in a Sequential Access Statistical Database. Master's Report, Department of Statistics, Kansas State University.

Noren, E. and Keller-McNulty, S. (1993). The Protection of Confidential Data in a Sequential Access Statistical Database. Proceedings of the Section on Survey Research Methods, American Statistical Association, 268–273.

Paass, G. (1988). Disclosure Risk and Disclosure Avoidance for Microdata. Journal of Business and Economic Statistics, 6, 487–500.

Pfleeger, C. (1989). Security in Computing. Englewood Cliffs, NJ: Prentice Hall, Inc.

Rogers, D.S. and Unger, E.A. (1993). Implementation of Security Mediators in POSTGRES Database Management System. Report TR-CS-93-10. Department of Computer and Information Sciences, Kansas State University.

Schlörer, J. (1976). Confidentiality of Statistical Records: A Threat Monitoring Scheme of On-Line Dialogue. Methods of Information in Medicine, 15, 36–42.

Schlörer, J. (1980). Disclosure from Statistical Databases: Quantitative Aspects of Trackers. ACM Transactions on Database Systems, 5, 467–492.

Skinner, C.J. (1992). On Identification Disclosure and Prediction Disclosure for Microdata. Statistica Neerlandica, 46, 21–32.

Skinner, C.J., Marsh, C., Openshaw, S., and Wymer, C. (1994). Disclosure Control for Census Microdata. Journal of Official Statistics, 10, 31–51.

Stonbraker, M. and Rowe, L. (1986). The Design of POSTGRES. Proceedings of the ACM-SIGMOD Conference, Washington D.C.

Received August 1997
Revised July 1998