Are statistical data bases secure?*

by DOROTHY E. DENNING
Purdue University
West Lafayette, Indiana

INTRODUCTION

Statistical databases contain sensitive information about individuals. Their objective is providing access to summary statistics about groups of individuals, while denying access to the records pertaining to any particular individual. But this objective is difficult to meet, as seemingly innocuous summaries contain small vestiges of the original data. By correlating enough summaries, confidential data may be deduced and the privacy of some individual therefore compromised.

As more and more of these databases are put on-line, the problem of preventing their compromise is of growing concern. Several questions have been raised:

• How easy is it to compromise a statistical data base?
• Under what conditions are statistical data bases secure?
• Are these conditions usually met?
• Is it practical or even possible to impose these conditions on statistical data bases?

Recent studies reveal that the problem is even more difficult than at first believed. Methods once thought to significantly reduce the threat of compromise—e.g., refusing to issue summaries about small groups of individuals—are in fact easy to circumvent. This paper surveys some of these studies. We begin with a general model of a database.

DATABASE MODEL

Components

Consider a statistical database containing records of information about n individuals. Each record contains category and data fields. Categories are used to select subgroups of individuals having common characteristics; they need not be disjoint from the data values. There may also be a unique identifier field, which is neither category nor data. Statistical summaries are requested from the database with queries which apply to subgroups of individuals.

Table I shows a database of size n = 12 containing information on customer accounts for a bank. Each individual belongs to exactly one category in each of these sets:

<table>
<thead>
<tr>
<th>Category</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sex</td>
<td>M, F</td>
</tr>
<tr>
<td>Profession</td>
<td>LAWYER, DOCTOR, JOURNALIST, PRESIDENT, SENATOR, BUDGET DIRECTOR</td>
</tr>
<tr>
<td>Board Member</td>
<td>YES, NO</td>
</tr>
<tr>
<td>(Number of) Overdrafts</td>
<td>any integer ≥0</td>
</tr>
</tbody>
</table>

Each individual also has data values for:

<table>
<thead>
<tr>
<th>Data</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Number of) Overdrafts</td>
<td>any integer ≥0</td>
</tr>
<tr>
<td>Amount of Overdrafts</td>
<td>any integer ≥0</td>
</tr>
</tbody>
</table>

All examples will refer to this database.

Compromise

Compromise occurs whenever it is possible to deduce from the responses of one or more queries information not previously known about an individual. The compromise is positive if it reveals that the individual belongs to some category or has a particular data value. The compromise is negative if it reveals only that the individual does not belong to some category or have a particular data value. For example, learning that individual L had 50 overdrafts is a positive compromise; learning only that he had at least one overdraft is a negative compromise. Partial compromise occurs when information about at least one individual is deduced; complete compromise occurs when everything in the database is deduced. A database is strongly secure if both positive and negative compromise are impossible; it is weakly secure if only positive compromise is impossible.

Queries

Researchers have studied two basic forms of queries: characteristic-specified and key-specified. Characteristic-specified queries request statistics about all individuals in the data base satisfying a given characteristic; a characteristic is a logical formula using categories as operands and Boolean operators: and (·), or (+), and not (−). For a char-
characteristic C, the set of records satisfying C is called the query set $X_C$ of C. For example, the characteristic $C = M \cdot (LAWYER + DOCTOR)$, specifying all male lawyers and doctors, has query set $X_C$ consisting of the records for individuals A, D, E, and H. We shall use relations in the specification of characteristics, e.g., OVERDRAFTS = 2, since these are simply abbreviations for the "or" of several values, e.g., 0-OVERDRAFTS + 1-OVERDRAFTS + 2-OVERDRAFTS.

Studies of compromise of statistical data bases have considered characteristic-specified queries $q(C)$ having one of three forms:

- $COUNT(C) = |X_C|$, where $|X_C|$ is the size of $X_C$;
- $SUM(C;j) = \sum_{i \in X_C} v_{ij}$, where $v_{ij}$ is data field j of record i;
- $select(C;j) = select v_{ij}$, where select is MEDIAN, SMALLEST, LARGEST, etc.

The query $COUNT(C)$ simply returns the number of individuals satisfying characteristic C. The query $SUM(C;j)$ returns the sum of the values in data field j for all individuals satisfying C. Note that the mean of these values can be computed from

$$MEAN(C;j) = \frac{SUM(C;j)}{COUNT(C)}.$$

For select = MEDIAN, the query $MEDIAN(C;j)$ returns the median value in data field j for all individuals satisfying C. Examples of queries are:

<table>
<thead>
<tr>
<th>Formal Query</th>
<th>Answer</th>
<th>Informal Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>$COUNT(M \cdot LAWYER)$</td>
<td>3</td>
<td>number of male lawyers</td>
</tr>
<tr>
<td>$COUNT(M \cdot LAWYER \cdot (OVERDRAFTS &gt; 10))$</td>
<td>2</td>
<td>number of male lawyers having more than 10 overdrafts</td>
</tr>
<tr>
<td>$SUM(LAWYER + DOCTOR; OVERDRAFTS)$</td>
<td>58</td>
<td>total number of overdrafts of all lawyers and doctors</td>
</tr>
<tr>
<td>$SUM(BUDGET DIRECTOR; AMOUNT)$</td>
<td>$100,000$</td>
<td>total amount of budget directors</td>
</tr>
<tr>
<td>$MEDIAN(LAWYER + DOCTOR; OVERDRAFTS)$</td>
<td>1</td>
<td>median number of overdrafts of all lawyers and doctors</td>
</tr>
</tbody>
</table>

Key-specific queries request statistics for a set of k individuals identified by a list of keys I. The keys are typically the names of individuals. For a set key I, the set of records identified by the keys is the query set $X_I$ of I. The query set size, $m$, is fixed for all queries. Queries for sums and selection queries are expressed as for characteristic-specified queries, with a key set I substituted for a characteristic C; e.g., $SUM(I;j)$. Since the size of a query set is always $m$, queries for counts are not applicable.

Key-specified queries are of less practical interest than characteristic-specified queries, since statistical databases generally do not give out data about particular individuals. However, if each individual is identified by a unique set of categories, any key list can be expressed with characteristics. For example, the key list (A, B) can be expressed as the characteristic $(M \cdot LAWYER \cdot BOARD MEMBER + M \cdot JOURNALIST)$. Thus, results which show that a database can be compromised with keys apply also to characteristics. However, caution must be taken in interpreting these results. Even though keys can be formulated as characteristics, it is not possible to do so without knowledge of the characteristics identifying the individuals named. Without this information, it is more difficult to control the composition of the query sets. For this reason, achieving compromise with characteristics may be more difficult than with keys. On the other hand, achieving compromise with characteristics may be easier than with keys since counts can be obtained for variable-size query sets.

**METHODS OF COMPROMISE**

**Using small or large query sets**

In one of the first published papers describing the problem of securing statistical databases, Hoffman and Miller described a simple algorithm for compromising a data base responding to $COUNT(q)$ queries. Their algorithm is based on the principle of using queries which return small counts to isolate an individual. For example, consider these two queries and responses:

$$COUNT(M \cdot BUDGET DIRECTOR) = 1$$
$$COUNT(M \cdot BUDGET DIRECTOR \cdot (OVERDRAFTS > 0)) = 1$$

If it is known that L is a male budget director, then the second query reveals that he had at least one overdraft. In general, if it is known that an individual belongs to categories $c_1, \ldots, c_k$ and if $COUNT(c_1, c_2, \ldots, c_k) = 1$, then the query $COUNT(c_1, c_2, \ldots, c_k, c_{k+1})$ reveals whether or not the individual also belongs to category $c_{k+1}$ (according to whether or not the response is 1 or 0). Indeed, compromise may be possible even if the response to the first query is larger than 1, say 17, if the response to the second query is the same.

Hoffman and Miller's result is equally applicable to queries which return large counts of all but a few individuals in the data base. For example, L's overdrafts can also be determined from the queries:

$$COUNT(M \cdot BUDGET DIRECTOR) = 11$$
$$COUNT(M \cdot BUDGET DIRECTOR \cdot (OVERDRAFTS > 0)) = 11$$

These results have led researchers to consider databases which do not respond to queries $q(C)$ when the query set size $|X_C|$ falls outside the range $[k, n-k]$ for some $k > 0$, where $n$ is the size of the database. It was anticipated that $k \approx 2$ would significantly reduce the threat of compromise.
Are Statistical Data Bases Secure?

TABLE I.—Database Containing Information on Customer Accounts for a Bank

<table>
<thead>
<tr>
<th>keys</th>
<th>categories</th>
<th>data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Sex</td>
<td>Profession</td>
</tr>
<tr>
<td>1</td>
<td>A</td>
<td>M</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>M</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>M</td>
</tr>
<tr>
<td>4</td>
<td>D</td>
<td>M</td>
</tr>
<tr>
<td>5</td>
<td>E</td>
<td>M</td>
</tr>
<tr>
<td>6</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>7</td>
<td>G</td>
<td>F</td>
</tr>
<tr>
<td>8</td>
<td>H</td>
<td>M</td>
</tr>
<tr>
<td>9</td>
<td>I</td>
<td>F</td>
</tr>
<tr>
<td>10</td>
<td>J</td>
<td>M</td>
</tr>
<tr>
<td>11</td>
<td>K</td>
<td>F</td>
</tr>
<tr>
<td>12</td>
<td>L</td>
<td>M</td>
</tr>
</tbody>
</table>

Trackers

Unfortunately, recent studies have shown this not to be the case. Schlorer showed that compromise may be possible even for large values of $k$ using a “tracker” technique. To illustrate his tracker, suppose $k=3$ for our database; i.e., no responses are given to queries which involve fewer than three or more than nine individuals. Consider these queries and responses, using as a tracker the characteristic $M \cdot$ BUDGET DIRECTOR:

\[
\begin{align*}
\text{COUNT}(M \cdot \text{BUDGET DIRECTOR}) &= 7 \\
\text{COUNT}(M \cdot \text{BUDGET DIRECTOR} + M \cdot (\text{OVERDRAFTS}=0)) &= 7
\end{align*}
\]

Because the responses to both queries are the same, it can be concluded that no male budget director had no overdrafts; therefore, $L$ must have had an overdraft. Palme suggested a similar technique for queries that compute means.

To construct a Schlorer tracker requires substantial knowledge of the characteristics identifying the individuals to be compromised. Denning, Denning, and Schwartz showed that general trackers may be found even without this knowledge. Such trackers may be used to determine the answer to any COUNT or SUM query not answerable because the query set size is too small or too large. For example, if $k=3$, the characteristic $T=\text{LAWYER} + \text{DOCTOR}$ is a general tracker. Let $C=M \cdot \text{BUDGET DIRECTOR}$. Whereas the number of $L$'s overdrafts cannot be determined directly from queries using only $C$ (since $|X_C|=1$), it can be determined from queries using characteristics combining $C$ with the tracker $T$ as follows. First, the sum of the queries $\text{COUNT}(T)$ and $\text{COUNT}(T)$ can be used to determine the size $n$ of the database; this gives $n=12$. Second, it can be deduced that $L$ is the only male budget director from the queries $\text{COUNT}(C+T)$ and $\text{COUNT}(C+T)$, since

\[
\begin{align*}
\text{COUNT}(C) &= \text{COUNT}(C+T) + \text{COUNT}(C+T) - n \\
&= 7 + 6 - 12 \\
&= 1
\end{align*}
\]

Third, the total number of overdrafts $X$ can be determined by adding the responses to the queries $\text{SUM}(T; \text{OVERDRAFTS})$ and $\text{SUM}(T; \text{OVERDRAFTS})$, getting $X=111$. Finally, since $L$ is the only budget director, the number of $L$'s overdrafts can be determined from the queries $\text{SUM}(C+T; \text{OVERDRAFTS})$ and $\text{SUM}(C+T; \text{OVERDRAFTS})$, as

\[
\begin{align*}
\text{SUM}(C; \text{OVERDRAFTS}) &= \text{SUM}(C+T; \text{OVERDRAFTS}) + \text{SUM}(C+T; \text{OVERDRAFTS}) - X \\
&= 108 + 53 - 111 \\
&= 50
\end{align*}
\]

We found further that for $k \leq n/4$ almost all data bases have general trackers (though finding them may not be easy). Trackers may also be applicable for larger values of $k$.

Adding dummy entries

Hoffman observed that data bases restricting the size of allowable query sets may also be subverted if records can be added to the database. If query $q(C)$ is not answerable because $|X_C|$ is too small, individuals with characteristic $C$ are added to the database; if $|X_C|$ is too large, individuals with the characteristic $C$ are added. This result, together with the tracker results, shows that limiting the size of allowable query sets is not an effective security measure.

Overlapping query sets

Another simple technique for compromising a data base is to construct query sets which have a high degree of overlap. For example, consider these queries and responses:

\[
\begin{align*}
\text{SUM(\text{LAWYER} + \text{F \cdot SENATOR}; \text{AMOUNT})} &= \$60,000 \\
\text{SUM(\text{LAWYER}; \text{AMOUNT})} &= \$60,010
\end{align*}
\]

Since $G$ is the only female senator, the second query set
includes all individuals in the first query set except G. Thus, the amount of her overdrafts can be determined by subtracting the response of the second query from the first. This observation has led researchers to consider the effectiveness of methods which restrict the amount of overlap among query sets.

**Linear systems**

Dobkin, Jones, and Lipton considered compromising with key-specified SUM queries using query sets of size m. They showed that, even if no two query sets can overlap by more than one record, compromise may be achievable in linear time (in m) without prior information, provided the data base is sufficiently large (roughly at least m^2 records). The method involves solving systems of equations for some particular data value. For example, suppose m=3 and let A_i denote the amount of overdrafts for the i'th individual. Consider these 5 queries:

- Q_1 = SUM(A,B,C; AMOUNT)
- Q_2 = SUM(D,E,F; AMOUNT)
- Q_3 = SUM(A,D,G; AMOUNT)
- Q_4 = SUM(B,E,G; AMOUNT)
- Q_5 = SUM(C,F,G; AMOUNT)

\[
\begin{align*}
A_1 + A_2 + A_3 &= \$10 \\
A_4 + A_5 + A_6 &= \$50,100 \\
A_1 + A_4 + A_7 &= \$160 \\
A_2 + A_5 + A_7 &= \$50,050 \\
A_3 + A_6 + A_7 &= \$50
\end{align*}
\]

G's overdraft amount can be determined from

\[
(Q_3 + Q_4 + Q_5 - Q_1 - Q_2)/3 = \$50
\]

Davida et al. and Kam and Ullman considered compromise under similar conditions. Schwartz, Denning, and Denning extended these results to weighted summing queries. We were surprised to learn that if the weights are unknown, complete compromise is possible (in linear time) provided one value in the data base is known. But compromise is impossible without initial information—even if overlap between queries is unrestricted.

Chin showed how achieving compromise with linear systems applies to characteristic-specified queries. His model assumes only that no two individuals belong to the same categories.

**Selection methods**

Dobkin, Jones, and Lipton, Davida et al., and Reiss have considered key-specified queries which select some element from the query set; e.g., the median, largest, or smallest data value. Their results show that using combinatorics, compromise is generally achievable in linear time (or better) provided that no two individuals have the same data value. For example, consider these two queries involving records with distinct overdraft values:

\[
\begin{align*}
\text{MEDIAN}(A,D,E; OVERDRAFTS) &= 2 \\
\text{MEDIAN}(B,D,G; OVERDRAFTS) &= 2
\end{align*}
\]

Since D is the only individual common to both queries, the returned median value 2 must be the number of D's overdrafts. DeMillo, Dobkin, and Lipton also showed that this method of compromise applies even if the database "lies about the median value, provided it returns with some value in the query set." For example, the previous example would result in a compromise of D's overdrafts even if 2 were not the true median of the query set values.

**PRIVACY SAFEGUARDS**

We have considered two measures—restricting the size of allowable query sets and restricting the amount of overlap between queries—which appear on the surface to reduce the threat of compromise. Yet on closer inspection, we observed that both of these measures fail to defend a system from even simple attacks. We shall now examine other privacy safeguards which will, hopefully, prove more effective.

**Completely secure data bases**

Several studies have reported conditions which, if imposed on the structure or contents of a database, guarantee its security. In most cases, these conditions depend on the users having little or no supplementary knowledge about either the structure or the contents of the records. This generally rules out the users themselves (or their friends) being represented in the database. Since these conditions are seldom met, they seem to have little value in practice.

**Random subfiles**

Rather than making the entire Census available to statisticians, the Census Bureau distributes random subfiles of the data. The 1960 Census, for example, was distributed on tape as a 1/1000 sample of the full Census with names and addresses removed. Also deleted was geographic information below the level of broad city-size class within the nine geographical divisions of the country. As a result, the payoff from an attempted compromise of a particular individual's privacy from Census tapes is sufficiently small to not pose a serious threat. Unfortunately, this technique is applicable only to large databases; other methods are needed to prevent compromise of small databases.

**Rounding schemes**

Several studies have been made of rounding schemes for modifying the answers to queries. One such approach is pseudo-random rounding. Truly random rounding is not
secure since the correct answer to any query can always be determined by averaging a sufficient number of responses to the same query. With pseudo-random rounding, the same query always returns the same response. This method appears to be reasonably effective, although any kind of "stochastic error" added to responses is subject to removal by methods from communication theory.

A second approach is to always round the actual response down to the nearest multiple of some integer or to report a range. For example, the database could respond to the query COUNT(LAWYER; OVERDRAFTS) with 50 (or the range 50-60) rather than the true value 56. However, as noted by Karpinski, this method is easily circumvented if records can be added to the database. To determine the true value of the above query, an intruder could simply add records to the data base for fictitious lawyers having one overdraft each; after the 4th record is added, the response becomes 60 (or the range 60-70), implying an original value of 56. Even if records cannot be added to the database, it may be possible to modify the characteristic of the query to include records of individuals for whom the number of overdrafts is known.

**Partitioned databases**

Another approach to preventing compromise partitions the database into groups. In the Yu and Chin scheme, queries must be for characteristics involving entire groups, making it impossible to isolate a particular individual. For example, the sample database could be partitioned by profession into four groups:

<table>
<thead>
<tr>
<th>Group</th>
<th>Identifying Characteristic</th>
<th>Members</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LAWYER</td>
<td>A, E, F, H</td>
</tr>
<tr>
<td>2</td>
<td>DOCTOR</td>
<td>D, I</td>
</tr>
<tr>
<td>3</td>
<td>JOURNALIST+</td>
<td>B, K, L</td>
</tr>
<tr>
<td>4</td>
<td>PRESIDENT+SENATOR</td>
<td>C, G, J</td>
</tr>
</tbody>
</table>

Whereas queries involving complete groups (e.g., all lawyers and doctors) are allowed; queries involving only part of a group (e.g., all male lawyers) are not. Clever query sequences can at best disclose information about an entire group. Yu and Chin show that the technique can be made effective even if the database is dynamically undergoing insertions, deletions, and updates.

The drawback is that the partitions may destroy the usefulness of the database. If the groups are not properly formed, users may not be able to obtain needed information. Whether or not a useful database can be partitioned in this way is open.

**Threat monitoring**

Threat monitoring techniques detect the possible occurrence of compromise. Felligi showed that it is at least theoretically possible to determine whether the response to a query, when correlated with the responses to earlier queries, could result in compromise. However, the method is too cumbersome to apply in practice. Hoffman and Miller suggested that a log or audit trail of queries be kept and inspected for unusual bursts of activity or queries returning small counts. Although certain compromises may go undetected, the presence of a log may provide a deterrent.

Schlörr suggested that frequency counts of categories be used to determine whether or not a given query might lead to compromise because of small counts. Response is not made to any query involving categories c1, . . . , ck unless the product of the frequency counts COUNT(ci)/n (for i=1, . . . , k) is above some threshold.

**CONCLUSIONS**

The "obvious" techniques for reducing the threat of compromise—e.g., limiting the range of allowable responses, restricting the amount of overlap between query sets, and certain rounding schemes—are easily circumvented. Other techniques—e.g., partitioning—may be robust, but at the price of limiting the usefulness of the data base. The conclusion is that complete privacy cannot be enforced without severely restricting the free flow of information. The questions of interest then become:

- Can we measure the relative security of a data base?
- What is an acceptable level of security?

**ACKNOWLEDGMENTS**

It is a pleasure to thank Mayer Schwartz and Peter Denning whose ideas and perceptions strongly influenced this paper.

**REFERENCES**

12. DeMillo, R. A., D. Dobkin, and R. J. Lipton, "Even Data Bases that


