A Fast Procedure for Finding a Tracker in a Statistical Database

DOROTHY E. DENNING
Purdue University
and
JAN SCHLÖRER
Universität Ulm, W. Germany

To avoid trivial compromises, most on-line statistical databases refuse to answer queries for statistics about small subgroups. Previous research discovered a powerful snooping tool, the tracker, with which the answers to these unanswerable queries are easily calculated. However, the extent of this threat was not clear, for no one had shown that finding a tracker is guaranteed to be easy.

This paper gives a simple algorithm for finding a tracker when the maximum number of identical records is not too large. The number of queries required to find a tracker is at most $O(\log_2 S)$ queries, where S is the number of distinct records possible. Experimental results show that the procedure often finds a tracker with just a few queries. The threat posed by trackers is therefore considerable.

Key Words and Phrases: confidentiality, database security, data security, statistical database, tracker
CR Categories: 4.33

1. INTRODUCTION

The objective of a statistical database is to provide statistical summaries about a population without revealing confidential data about any individual. The problem is that it is frequently possible to compromise an individual's privacy by deducing information about him from the summaries. This problem is particularly difficult to control in modern database systems, especially the relational ones, which make it easy for on-line users to pose queries about arbitrary subgroups of individuals (see [5, 6, 13] for surveys).

In our earlier work we studied a simple, powerful snooping tool, the "tracker" [6, 11, 12, 14]. A tracker is a set of auxiliary characteristics which are added to the original characteristics in the formation of a query. The auxiliary characteristics pad the query set of the original characteristics to form answerable queries; the

This work was supported in part by the National Science Foundation under Grant MCS77-04835 at Purdue University.

Authors' addresses: D. E. Denning, Computer Science Department, Purdue University, West Lafayette, IN 47907; J. Schlörer, Klinische Dokumentation, Universität Ulm, Prittwitzstrasse 6, D 7900 Ulm, W. Germany.

questioner subtracts out the effect of the auxiliary characteristics to determine the answer to the query for the original characteristics. This circumvents the restriction that queries for small query sets (or their complements) cannot be answered.

In [11] we showed how to develop an "individual" tracker from the known characteristics of a particular individual in order to identify (if possible) the individual's record in the database and determine additional, unknown characteristics of the individual. The concept was extended in [6] and [14] to "general" and "double" trackers. A general tracker can be used to calculate the answer to any unanswerable query provided all query sets, including between one-fourth and three-fourths of the population, are answerable. A double tracker can be used in a similar way as long as all query sets, including between one-third and two-thirds of the population, are answerable. An $O(N^2)$ algorithm for constructing a general tracker was given in [6], where N is the number of records, but this algorithm unrealistically assumes that the entire database could be inspected.

Trackers were generalized further in [12] to "union" trackers, which also allow calculating the answer to any query. Such trackers may exist even when the only answerable queries are those involving approximately one-half the population, but then they are more difficult to find and apply than general trackers. The results also showed that large proportions of the possible queries in most databases are general trackers; thus a general tracker is apt to be discovered quickly simply by guessing. However, no bound on the number of queries required to find a tracker was found.

This paper continues our study of trackers by showing that a general tracker can usually be found quickly. The tracker finding procedure assumes that the user knows the value-sets for each category of values stored in the database; however no prior knowledge of the contents of particular records is needed. The algorithm will always find a tracker when the maximum number of identical records is not too large; empirical evidence shows that this condition is often satisfied. The number of queries required to find a tracker is at most $O(\log_2 S)$, where S is the number of distinct records possible. The results of an experiment performed on a statistical database show that the procedure often finds a tracker with just one or two queries.

These results show that existing designs for query systems do not adequately prevent disclosure of confidential data by combinatorial inference. This observation is not new. However, it has only been recently that we have begun to understand the myriad of inference techniques that may be used. We are continually finding that compromise is easier than once thought and that many controls are either too restrictive; too costly to implement, or easy to subvert. For example, controls which limit the overlap between query sets may preclude trackers. However, they are too restrictive for most applications, which require comparisons between the statistics for the subgroups of a population and the population as a whole (e.g., the average salary of all female employees and the average salary of all employees). Overlap controls are also extremely costly to implement if they require comparing each new query with all previous ones to determine if there would be too much overlap. Finally, overlap controls are often easily subverted even if the amount of overlap is restricted to just one record [3,

4, 7, 9, 10]. Better controls are needed. We hope that by furthering our understanding of the nature of the problem, we will be better able to evaluate the effectiveness of proposed solutions.

2. THE MODEL

A statistical database contains records for N individuals. Each *record* contains m fields, where the ith field ($i = 1, \ldots, m$) contains the value of the ith *attribute* (or *variable*) V_i. Each attribute V_i has n_i possible *values*, denoted $v_{i1}, \ldots, v_{in_i}$. (There may also be a unique identifier field which cannot be employed in a statistical query.) We assume that the database is free from updates and deletions during a period when compromise is attempted.

Table I shows a (sub) database containing confidential student records for a hypothetical university having 50 departments. Each record has $m = 5$ fields, whose possible values are shown in Table II. The attribute SAT specifies a student's average on the SAT (Scholastic Aptitude Test) and GP specifies a student's current grade point average.

A *query* for statistics is given in terms of a *characteristic formula* C, which, informally, is any logical formula over the attribute values, using the operators *and* ($\cdot$), *or* ($+$), and *not* ($^-$). An example of a formula is "M·(CS + EE)," which specifies all males in either the CS or EE department. The set of records whose values match C is called the *query set* X_C of C. Although a characteristic formula is not a set, we shall write $C1 \subseteq C2$ to denote $X_{C1} \subseteq X_{C2}$ (e.g., M·CS $\subseteq$ CS). We denote by D a "global" formula—one whose query set is the entire database; thus $C \subseteq D$ for any formula C.

Two types of queries are considered in our examples. The first, denoted COUNT(C), computes the number of individuals satisfying formula C. For example, COUNT(M·(CS + EE)) = 6. The second, denoted SUM(C; V_i), returns the sum of values for attribute V_i for all individuals satisfying formula C. For example, SUM(F; GP) = 4.0 + 2.8 = 6.8. Note that an average value can be computed from SUM(C; V_i)/COUNT(C). We shall use $q(C)$ to denote either a counting or summing query about individuals described by formula C.

Given n_i values for each of m attributes V_i ($i = 1, \ldots, m$), there are $S = n_1 \cdot n_2 \cdot \cdots \cdot n_m$ possible distinguishable records described by formulas of the form "$v_{1k_1} \cdot \cdots \cdot v_{mk_m}$," where v_{ik_i} is some value of variable V_i. The query set corresponding to a formula of the form "$v_{1k_1} \cdot \cdots \cdot v_{mk_m}$" is called an *elementary set*. Note that the records in an elementary set (if any) are indistinguishable. Thus there are S elementary sets in the database, some of which may contain no records. We let g denote the maximum size of all elementary sets; thus g is the maximum number of individuals having identical records, i.e., the size of the largest indecomposable query set. If the number of records N satisfies $N \leq S$, then $g = 1$ is possible. For Table I, $S = 820{,}000$ and $g = 1$.

Compromise occurs when a questioner deduces, from the responses of one or more queries, confidential information of which he was previously unaware [2]. It is well known that compromise is easy when query sets can be small or large compared to the size of the database [1, 8, 11]. As an example, suppose that a questioner, who knows that Allen is a male CS major graduating in 1980, poses

Table I. Statistical Database with $N = 9$ Student Records

NAME	SEX	MAJOR	CLASS	SAT	GP
Allen	M	CS	1980	550	3.4
Brooks	M	EE	1981	510	2.5
Cook	M	EE	1978	630	3.5
Davis	F	CS	1980	800	4.0
Evans	M	BIO	1979	500	2.2
Frank	M	EE	1978	450	3.0
Good	M	CS	1978	700	3.8
Hall	F	PSY	1979	580	2.8
Iles	M	CS	1981	600	3.0

Table II

Attribute (V_i)	Values	Number of values (n_i)
SEX	M, F	2
MAJOR	CS, EE, BIO, PSY, ...	50
CLASS	1978, 1979, 1980, 1981	4
SAT	310, 320, 330, ..., 790, 800	50
GP	0.0, 0.1, 0.2, ..., 3.9, 4.0	41

the two queries:

$$\text{COUNT(M·CS·1980)} = 1$$
$$\text{SUM(M·CS·1980; GP)} = 3.4.$$

These responses reveal that Allen's grade point is 3.4.

This trivial method of compromise may be prevented if queries which involve small or large query sets are not answered. Letting k denote a lower bound on query set size, the answer to a query $q(C)$ is released if $k \leq \text{COUNT}(C) \leq N - k$, but withheld otherwise. We shall write "$q(C) = \#$" if a query is not answered.

3. GENERAL AND DOUBLE TRACKERS

For $k \leq N/4$, it is possible to calculate the answer to any unanswerable counting or summing query $q(C)$ with the aid of a general tracker [6, 14].

A *general tracker* is a formula T such that $2k \leq \text{COUNT}(T) \leq N - 2k$. Given a tracker T, the procedure for calculating the value $q(C)$ when $q(C)$ is unanswerable because either $\text{COUNT}(C) < k$ or $\text{COUNT}(C) > N - k$ is as follows.

First $X = q(T) + q(\bar{T})$ is calculated. Next, if $q(C + T)$ and $q(C + \bar{T})$ are answerable, it will be true that $\text{COUNT}(C) < k$ and that $q(C) = q(C + T) + q(C + \bar{T}) - X$. Otherwise it will be true that $\text{COUNT}(C) > N - k$ and that $q(C) = 2X - q(\bar{C} + T) - q(\bar{C} + \bar{T})$. The following example illustrates.

Example. Let $k = 2$ for the database of Table I. Suppose a questioner wishes to learn Davis's GP. The query SUM(F·CS; GP) is not directly answerable since $\text{COUNT(F·CS)} = 1 < k$. However, Davis's GP can be calculated using the general

tracker T = "CS" from the following:

$$X = \text{SUM(CS; GP)} + \text{SUM}(\overline{\text{CS}}; \text{GP}) = 28.2$$

$$\begin{aligned} \text{SUM(F·CS; GP)} &= \text{SUM(F·CS + CS; GP)} + \text{SUM(F·CS + } \overline{\text{CS}}\text{; GP)} - X \\ &= 14.2 + 18.0 - 28.2 \\ &= 4.0 \end{aligned}$$

Double trackers are applicable for an even narrower range of answerable queries, namely, when $k \leq N/3$ [6, 14]. A *double tracker* is a pair of characteristics (T, U) such that

$$T \subseteq U,$$
$$k \leq \text{COUNT}(T) \leq N - 2k,$$

and

$$2k \leq \text{COUNT}(U) \leq N - k.$$

The amount of computation required to determine the answer to an unanswerable query with a double tracker is about the same as that for a general tracker. If $\text{COUNT}(C) < k$, the formula $q(C) = q(U) + q(C + T) - q(T) - q(\overline{C \cdot T} \cdot U)$ is used; otherwise $\text{COUNT}(C) > N - K$, and the formula $q(C) = q(\bar{U}) - q(\bar{C} + T) + q(T) + q(\overline{\bar{C} \cdot T} \cdot U)$ is used.

If $k = 3$ in the sample database of Table I, general trackers are not applicable because $k > N/4$. However, double trackers are applicable, and the characteristics T = "1978" and U = "1978 + 1979 + F" form a double tracker.

4. CONSTRUCTING A TRACKER

We assume that the user's prior knowledge of the database is limited to the attributes and their values; i.e., for each V_i, the user knows $v_{i1}, \ldots, v_{in_i}$, but the user does not know the distribution of values among the records or the contents of any particular record. In a later section, we shall consider the implications when the values of some of the attributes are not known.

Figure 1 illustrates the strategy employed to construct a tracker. The procedure starts with an answerable query $\text{COUNT}(C)$ for which $k \leq \text{COUNT}(C) < 2k$. (If $2k \leq \text{COUNT}(C) \leq N - 2k$, C is a tracker and there is nothing to do. If $N - 2k < \text{COUNT}(C) \leq N - k$, we apply the algorithm with $\bar{C}$ instead of C.) Formulas $C1$ and $C2$ are initialized to C and the global formulas D, respectively. At each step, the algorithm extends $C1$ or restricts $C2$ until a tracker T is found such that $C1 \subseteq T \subseteq C2$ and/or until $C1$ and $C2$ differ by a single elementary set. If the size of the largest elementary set g does not exceed $N - 4k$, a tracker will always be found. To ensure fast convergence, binary search is employed to extend $C1$ and restrict $C2$.

A procedure for constructing a tracker is given in Figure 2. The attributes $C1$, $C2$, T, $E1$, and $E2$ all represent formulas. For a disjunctive formula $E = v_1 + \cdots + v_n$, the operation $(E1, E2) :=$ **bisect** (E) assigns to $E1$ the disjunction of the first $\lfloor n/2 \rfloor$ terms and to $E2$ the disjunction of the remaining terms.

Initially $C1$ is set to C (or $\bar{C}$) and $C2$ is set to D so that $C1 \subseteq C2$. The procedure then makes one pass for each attribute V_i. During the ith pass, $C1$ is extended or $C2$ restricted using the values of V_i. This is done by assigning to E all of the values of V_i and iteratively bisecting E into $E = E1 + E2$. After each bisection, a characteristic $T = C1 + C2 \cdot E1$ is formed such that $C1 \subseteq T \subseteq C2$, and the query

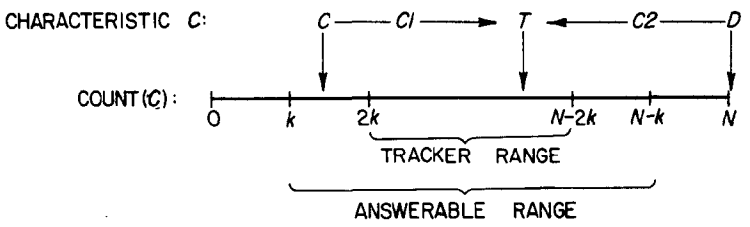

Fig. 1. Process of constructing a tracker.

Given: Characteristic C such that $k \leq \text{COUNT}(C) \leq N - k$, where $k \leq N/4$.

```
1.  if 2k ≤ COUNT(C) ≤ N − 2k then stop; "C is a general tracker"
2.  if COUNT(C) < 2k
3.      then C1 := C
4.      else C1 := C̄ "COUNT(C̄) < 2k";
5.  C2 := D;
    "construct general tracker T such that C1 ⊆ T ⊆ C2"
6.  for i := 1 to m do "one pass for each attribute V_i"
7.    begin "using bisection on the n_i values of V_i, extend C1 and restrict
          C2 until a single value v_{ik_i} is left"
8.      E := v_{i1} + ··· + v_{in_i};
9.      while |E| > 1 do "|E| is the number of values OR-ed in E"
10.       begin
11.       (E1, E2) := bisect (E);
12.       T := C1 + C2·E1; "construct T such that C1 ⊆ T ⊆ C2"
13.       a := COUNT(T); "query the database"
14.       if a = # then "if the query is unanswerable, swap E1 and E2"
15.         begin
16.         swap (E1, E2);
17.         T := C1 + C2·E1;
18.         a := COUNT(T) "this query will be answerable"
19.         end;
20.       if 2k ≤ a ≤ N − 2k then stop; "T is a general tracker"
21.       if a < 2k
22.         then begin "k ≤ COUNT(T) < 2k, so extend C1"
23.           C1 := T;
24.           E := E2
25.           end
26.         else begin "N − 2k < COUNT(T) ≤ N − k, so restrict C2"
27.           C2 := T;
28.           E := E1
29.           end "C1 and C2 will now form a double tracker"
30.     end of while
31.   end of for
    end of procedure
```

Fig. 2. Procedure for constructing a tracker.

COUNT(T) is posed ($E1$ and $E2$ may require swapping in order that COUNT(T) be answerable). If COUNT(T) is too small, $C1$ is extended to T and $E2$ is bisected next; conversely, if COUNT(T) is too large, $C2$ is restricted to T and $E1$ is bisected next. The iteration terminates when T is a general tracker, or when a single value v_{ik_i} is left, whichever is sooner. In the worst case, the procedure terminates after all attributes have been processed. In this case, $C1$ and $C2$ will differ by a single elementary set "$v_{1k_1} \cdot \cdots \cdot v_{mk_m}$."

The following assertions collectively imply the correct operation of the algorithm:

(1) *Every assignment to* T (*lines* 12 *and* 17), $C1$ (*line* 23), *and* $C2$ (*line* 27) *preserves the relation* $C1 \subseteq T \subseteq C2$.

(2) *At all times,* $k \leq \text{COUNT}(C1) < 2k$ *and* $N - 2k < \text{COUNT}(C2) \leq N$.

(3) *Every time control reaches line* 20, $k \leq a \leq N - k$. A violation would imply that neither of the queries at lines 13 and 18 is answerable. If the query at line 13 is not answerable, then, since $k \leq \text{COUNT}(C1) < 2k$, it must be the case that $C2 = D$ and $\text{COUNT}(T) = \text{COUNT}(C1 + D \cdot E1) > N - k$; once $C2$ is restricted (line 27), $\text{COUNT}(C2) < N - k$, and the query at line 13 is always answerable. But $\text{COUNT}(C1 + D \cdot E1) > N - k$ implies $k > N - \text{COUNT}(C1 + D \cdot E1) = \text{COUNT}((D - C1) \cdot E2)$ ($E1$ and $E2$ are disjoint). Since $k \leq N/4$, $k \leq \text{COUNT}(C1 + D \cdot E2) = \text{COUNT}(C1) + \text{COUNT}((D - C1) \cdot E2) < 3k \leq N - k$, whence the query at line 18 is answerable after $E1$ and $E2$ are swapped.

Assertions (4) and (5) show that the difference between $C1$ and $C2$ converges to a single elementary set.

(4) *Let* A_i *denote the difference* $C2 - C1$ *at the start of the* i*th pass* [*line* 7]; *then* $A_{i+1} = A_i \cdot v_{ik_i}$ *for some value* v_{ik_i} *of* V_i ($i = 1, \ldots, m - 1$). The effect of lines 23 and 24 is to extend $C1$ to $C1 + A_i \cdot E1$ and set E to $E2$, whence the difference between $C2$ and $C1$ becomes $A_i \cdot E$. The effect of lines 27 and 28 is to reduce $C2$ to $C1 + A_i \cdot E1$ and set E to $E1$, whence the difference between $C2$ and $C1$ also becomes $A_i \cdot E$ (see Figure 3). Each iteration repeats this, leaving the difference $C2 - C1$ at $A_i \cdot E$ for a more restricted E. When the entire pass terminates, $|E| = 1$, implying $E = v_{ik_i}$ for some value v_{ik_i} of V_i. This terminal difference is the initial difference for the $(i + 1)$st pass (see Figure 4).

(5) *On completion of the* m*th pass* (*without finding a tracker*) $C1$ *and* $C2$ *differ by a single elementary set.* Assertion (4) applied repeatedly shows that the final difference $C2 - C1$ is $A_1 \cdot F = \bar{C} \cdot F$, where $F = v_{1k_1} \cdot v_{2k_2} \cdot \cdots \cdot v_{mk_m}$ is an elementary set. Furthermore, $F \subseteq \bar{C}$, since otherwise $C1 = C2$ and both $C1$ and $C2$ would be general trackers. Thus $C2 - C1 = F$.

(6) *The procedure always halts.* Each pass takes at most $1 + \lfloor \log_2 n_i \rfloor$ iterations. Thus the entire algorithm cannot run longer than $m + \sum_{i=1}^{m} \lfloor \log_2 n_i \rfloor \leq m + \lfloor \log_2 S \rfloor$ iterations.

THEOREM 1. *The algorithm always finds a general tracker with at most* $2(m + \lfloor \log_2 S \rfloor)$ *queries, provided* $k \leq \lfloor (N - g)/4 \rfloor$.

PROOF. Assume to the contrary that the procedure terminates without finding a tracker. By assertion (5),

$$\text{COUNT}(C2) - \text{COUNT}(C1) = \text{COUNT}(F) \leq g \leq N - 4k.$$

From assertion (2), $\text{COUNT}(C1) < 2k$ and $\text{COUNT}(C2) > N - 2k$; thus $(N - 2k) - (2k) < \text{COUNT}(F)$. This leads to the contradiction $N - 4k < N - 4k$. Each pass requires at most two queries per iteration. □

If $k > \lfloor (N - g)/4 \rfloor$, the procedure may not return a general tracker. However, as long as $k < \lfloor N/4 \rfloor$, the probability that a tracker will be found may remain

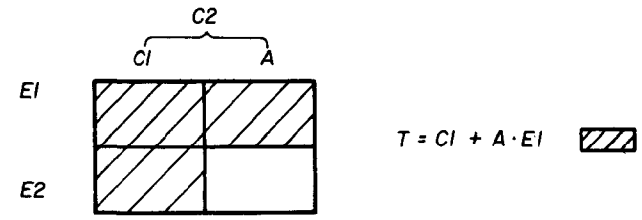

Fig. 3. Characteristic sets during single iteration of the inner while loop.

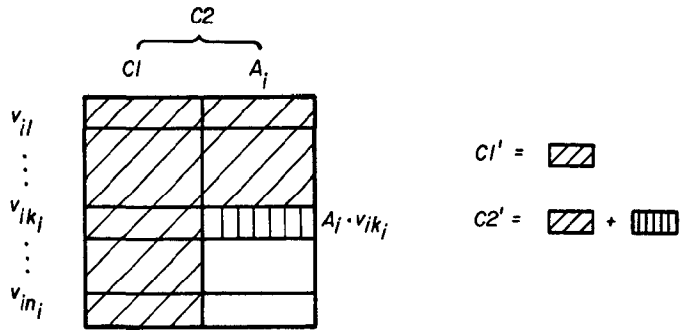

Fig. 4. Characteristic sets before ($C1$ and $C2$) and after ($C1'$ and $C2'$) the ith pass of the procedure.

high. This is illustrated by the experimental results described in Section 6 and shown in Table V.

The characteristics $C1$ and $C2$ may satisfy the conditions for a double tracker before a general tracker is constructed. This happens on the first execution of line 29, if $k \leq N/4$. Thus the procedure can be modified to return either a general or double tracker, depending on which is found first.

The exact implementation of the procedure will depend on the user and the system. The bookkeeping for the formulas $C1$, $C2$, T, E, $E1$, and $E2$ could be performed either by the computer or off-line. If the values $v_{i1}, \ldots, v_{in_i}$ of attribute V_i (line 11 in Figure 2) are ordered by $v_{i1} < v_{i2} < \cdots < v_{in_i}$, then the formulas E, $E1$, and $E2$ are easily represented by three values of V_i: x, y, z, where $E =$ "$x \leq V_i \leq z$," $E1 =$ "$x \leq V_i \leq y$," and $E2 =$ "$y < V_i \leq z$." If the values are not ordered, they can be represented by three pointers into the list (two for the ends and one for the middle).

If the database has the facility to store record sets defined by characteristic formulas for later reuse, it will not be necessary to form successively more complicated formulas for $C1$, $C2$, and T. Instead, the record sets corresponding to these formulas can be developed step by step. This facility was available in the system on which our experiments were performed, which employed totally inverted lists and the option to invert quantitative continuous variables using arbitrary classifications [15].

5. AN EXAMPLE

The procedure may be used to find a tracker for the database of Table I when $k \leq [(N - g)/4] = [(9 - 1)/4] = 2$. We now show how this may be done starting with the characteristic $C =$ F.

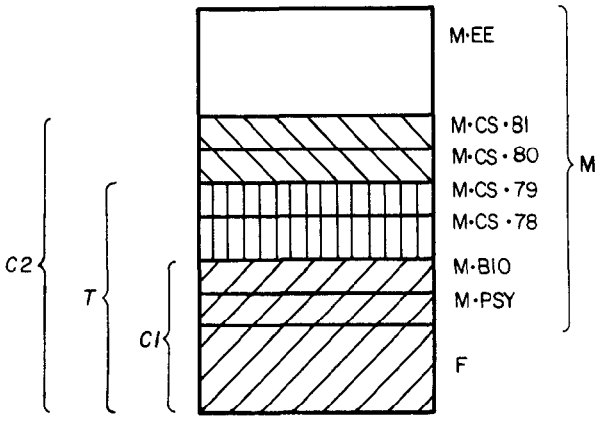

Fig. 5. Characteristics *C1*, *C2*, and *T* after a general tracker is found for the sample database of Table I.

Initial Conditions. $C1 = \mathrm{F}$, $C2 = D$, and $\bar{C} = \mathrm{M}$. (We do not attempt to bisect the values of SEX, since SEX is already bisected by the starting query.)

Pass 1: Bisect the Values of MAJOR. For simplicity, we assume that only the values CS, EE, BIO, and PSY are possible.

(a) *Bisection of* $E = \mathrm{CS} + \mathrm{EE} + \mathrm{BIO} + \mathrm{PSY}$. Try $E1 = \mathrm{CS} + \mathrm{EE}$, $E2 = \mathrm{BIO} + \mathrm{PSY}$, and $T = C1 + C2 \cdot E1 = \mathrm{F} + D \cdot (\mathrm{CS} + \mathrm{EE}) = \mathrm{F} + \mathrm{M} \cdot (\mathrm{CS} + \mathrm{EE})$. Since $\mathrm{COUNT}(T) = 8 > N - k$, the query on line 13 is not answerable; thus *E1* and *E2* are switched, giving $E1 = \mathrm{BIO} + \mathrm{PSY}$, $E2 = \mathrm{CS} + \mathrm{EE}$, and $T = \mathrm{F} + \mathrm{M} \cdot (\mathrm{BIO} + \mathrm{PSY})$. Since $\mathrm{COUNT}(T) = 3 < 2k$, *C1* is extended to *T* and *E2* is bisected on the next step.

(b) *Bisection of* $E = \mathrm{CS} + \mathrm{EE}$. Try $E1 = \mathrm{CS}$, $E2 = \mathrm{EE}$, and $T = C1 + C2 \cdot E1 = (\mathrm{F} + \mathrm{M} \cdot (\mathrm{BIO} + \mathrm{PSY})) + D \cdot \mathrm{CS} = (\mathrm{F} + \mathrm{M} \cdot (\mathrm{BIO} + \mathrm{PSY})) + \mathrm{M} \cdot \mathrm{CS} = \mathrm{F} + \mathrm{M} \cdot (\mathrm{BIO} + \mathrm{PSY} + \mathrm{CS})$. Since $\mathrm{COUNT}(T) = 6 > N - 2k$, *C2* is restricted to *T*. At this point, *C1* and *C2* are double trackers, but we shall continue with the procedure until a general tracker is found.

Since *E1* cannot be further bisected, pass 1 terminates with $C1 = \mathrm{F} + \mathrm{M} \cdot (\mathrm{BIO} + \mathrm{PSY})$ and $C2 = \mathrm{F} + \mathrm{M} \cdot (\mathrm{BIO} + \mathrm{PSY} + \mathrm{CS})$, and thus $C2 - C1 = \mathrm{M} \cdot \mathrm{CS}$.

Pass 2: Bisect the Values of CLASS

(a) *Bisection of* $E = 1978 + 1979 + 1980 + 1981$. Try $E1 = 1978 + 1979$ and $E2 = 1980 + 1981$. Then $T = C1 + C2 \cdot E1 = (\mathrm{F} + \mathrm{M} \cdot (\mathrm{BIO} + \mathrm{PSY})) + (\mathrm{F} + \mathrm{M} \cdot (\mathrm{BIO} + \mathrm{PSY} + \mathrm{CS})) \cdot (1978 + 1979) = \mathrm{F} + \mathrm{M} \cdot (\mathrm{BIO} + \mathrm{PSY} + \mathrm{CS} \cdot (1978 + 1979))$. Since $\mathrm{COUNT}(T) = 4$, *T* is a general tracker and the procedure terminates. Figure 5 illustrates the final state of *C1*, *C2*, and *T*.

6. NUMBER OF QUERIES REQUIRED TO FIND A TRACKER

The effort required to find a tracker with the procedure depends on two factors: the number of queries that must be asked, and the effort required by the user and the system to formulate and respond to the queries. Since the latter is system dependent, we shall consider only the number of queries required to find a tracker. After examining theoretical bounds on the number of queries, we shall report the results of an actual experiment.

Table III. Maximum Number of Queries Required to Find a General Tracker

	m				
n_i	1	2	5	10	20
2–3	4	8	20	40	80
4–7	6	12	30	60	120
8–15	8	16	40	80	160
16–31	10	20	50	100	200
32–63	12	24	60	120	240
64–127	14	28	70	140	280
128–255	16	32	80	160	320

As noted in assertion (6) in Section 4, the total number of iterations (through the inner while loop) is bounded by

$$m + \sum_{i=1}^{m} \lfloor \log_2 n_i \rfloor \leq m + \lfloor \log_2 S \rfloor,$$

where m is the number of attributes and S is the total number of elementary sets. Since at most two queries are asked during each iteration, the total number of queries asked, Q, is bounded by $Q \leq 2\ (m + \lfloor \log_2 S \rfloor)$. Thus $Q = O(\log S)$.

Table III shows values for Q for different values of m and n_i, where the n_i are equal (under equal n_i, $Q \leq 2(m + m\lfloor \log_2 n_i \rfloor) = 2m(1 + \lfloor \log_2 n_i \rfloor)$). For example, a general tracker can be found for a database with 10 attributes each having 8 values, within 80 queries. Such a database has $S = 8^{10}$ elementary sets and thus could uniquely represent over 1 billion individuals. The database of Table I has $m = 5$, $S = 820{,}000$, and $\lfloor \log_2 S \rfloor = 19$, whence a tracker will be found within 48 queries.

It is tempting, but erroneous, to conclude that the larger the value of Q, the more resistant the database is to the threat of compromise; the flaw in the argument is that the larger the values of n_i and m, the smaller the sizes of the elementary sets. This improves the odds not only of finding a tracker, but also of isolating a particular individual in the database once a tracker is found.

In fact, a tracker may be found much sooner than the bound Q. The following theorem states that a general tracker will be found during the ith pass of the procedure provided that no more than $N - 4k$ individuals are identified by any i values of the first i attributes bisected.

THEOREM 2. *Let g_i be the maximum number of individuals identified by the conjuction of any i values, one from each of $V_1, \ldots, V_i$. If $g_i \leq N - 4k$, then a general tracker will be found within the first i passes of the procedure.*

PROOF. At the end of the ith pass, $C2 - C1 = \bar{C} \cdot v_{1k_1} \cdot \cdots \cdot v_{ik_i}$ [assertion (5)]. Thus, COUNT($C2$) − COUNT($C1$) ≤ COUNT($v_{1k_1} \cdot \cdots \cdot v_{ik_i}$) ≤ g_i. As in the proof of Theorem 1, $g_i \leq N - 4k$ implies COUNT($C1$) ≥ $2k$ or COUNT($C2$) ≤ N − 2k; that is, one of $C1$ or $C2$ is a tracker. □

For example, every individual in the database of Table I is uniquely identified by the attribute SAT. Thus with $k = 2$ (implying $g_i \leq 1$ is needed for termination within i passes), bisecting SAT first would yield a tracker on the first pass within

Table IV

Attribute	Number of values	Relative frequencies
City of health insurance	6	0.020–0.372
Sex	2	0.494–0.506
Age (5 year classes, 15–59 years)	9	0.074–0.135
Personal status	5	0.041–0.694
Number of children	4	0.088–0.519
Number of inhabitants at residence	4	0.183–0.373
Job qualification	3	0.048–0.484
Type of job	9	0.0002–0.523

$2(1 + \lfloor \log_2 50 \rfloor) = 12$ queries, since $g_i = 1$. With $k = 1$ (implying $g_i \leq 5$ is needed), a tracker would also be found in one pass starting with MAJOR since $g_i = 4$ (at most four individuals have a common MAJOR).

This result suggests that an intruder with prior knowledge of the distribution of values in a database may be able to find a general tracker with as few as one or two queries. For example, if it is known that a database contains roughly as many males as females, then the trivial general tracker "male" could be discovered at once. Moreover, a double tracker may be found even before a general tracker, as in the example of Section 5.

This unfortunate result suggests that any attempt by the database to detect the construction of a tracker is likely to fail. Were a large number of queries always required to construct a tracker, it might be possible to detect the convergence toward a tracker from the sequence of query sets formed (although an intruder could thwart this by injecting spurious queries into the sequence). However, if only one or two queries are required to construct a tracker, there is little hope for detection.

The procedure was applied to a medical database stemming from a health care project [16], using an evaluation program developed by Selbmann [15]. The database contained $N = 31{,}465$ records and several dozen attributes, of which but $m = 8$ were used (see Table IV). The column labeled "Relative frequencies" gives the range of relative frequencies for each of the values of an attribute. For example, each of the six cities was associated with between $0.02N$ and $0.372N$ of the records.

There were $S = 233{,}280$ elementary sets in the subdatabase employed. Thus the maximum number of queries required to find a general tracker with the procedure is $2(m + \lceil \log_2 S \rceil) = 2(8 + 17) = 50$. No systematic attempt was made to determine g, the maximal cardinality of an elementary set. However, an elementary set containing 82 records was uncovered during the experiment.

Three variations of the experiment were performed as follows:

(1) *Completely Random.* The sequence of attributes selected for bisection, as well as the sequence of values employed in the bisections, was chosen at random (20 trials).
(2) *Partly Random.* The order in which attributes were selected for bisection was chosen at random. However, the bisection of each attribute took into account the frequency distribution of the values (20 trials).

Table V. Numbers of Queries and Attributes Needed to Find General Trackers

Assumed value of k:	3933 (= $\lfloor N/8 \rfloor$)	7734 (= $\lceil N/4 - (3/4)\sqrt{N} \rceil$)	7861 (= $\lfloor N/4 \rfloor - 5$)
In order to qualify as a general tracker COUNT(T) must be ϵ:	[7866, 23599]	[15468, 15997]	[15722, 15743]
Variation (1), *Completely Random* (20 *trials*)			
No. of queries			
Minimum	1	1	2
Maximum	4	16	22[a]
Mean	1.4	7.2	13.1[b]
No. of attributes needed (mean)	1.0	3.6	5.7[b]
Variation (2), *Partly Random* (20 *trials*)			
No. of queries			
Minimum	1	1	3
Maximum	1	6	14
Mean	1.0	3.9	8.0
No. of attributes needed (mean)	1.0	2.7	4.8
Variation (3), *Chosen* (8 *trials*)			
No. of queries			
Minimum	1	1	3
Maximum	1	6	9
Mean	1.0	3.4	6.1
No. of attributes needed (mean)	1.0	2.3	3.9

Note. Experiments were performed in three variations (1), (2), and (3). For details of design see the text. The largest elementary set encountered in the experiment contained 82 records,[a] so in the rightmost column $k = \lfloor N/4 \rfloor - 5 > \lfloor (N - 82)/4 \rfloor > \lfloor (N - g)/4 \rfloor$; nevertheless, all trials except one returned a general tracker.

[a] One trial did not reach the target interval, but stopped after 22 queries at two record subsets with cardinalities 15702 and 15784, differing by an elementary set of size 82. Maximum of the 19 successful trials was 17 queries.

[b] Of the 19 successful trials.

(3) *Chosen.* The first attribute bisected was predetermined; however, the order in which the remaining attributes were selected and then bisected took into account the frequency distributions (8 trials, one starting with each of the 8 variables).

The initial characteristic C was constructed throughout using values of but one of the attributes, the "basis attribute." In variation (1) the basis attribute was chosen at random (in general it was not the attribute of the first outer loop); in variations (2) and (3) the first attribute of the sequence was employed as the basis attribute. If three successive trials failed to yield C with answerable COUNT(C), the next attribute was tried (this happened only with variation (1) in 2 of the 20 cases). For the initial formula C, $k = \lfloor N/4 \rfloor = 7866$ was assumed throughout; thus $7866 \leq \text{COUNT}(C) \leq 23599$ was required to render C "answerable."

The results are presented in Table V for three different values of k. The value

```
Given: Characteristic C such that k ≤ COUNT(C) ≤ N − k.
 1. C1 := C;
 2. C2 := D;
 3. for i := 1 to m do "one pass for each variable V_i"
 4.   begin "use bisection as in Figure 2"
 5.     E := v_i1 + ··· + v_in_i;
 6.     while |E| > 1 do "|E| is the number of values OR-ed in E"
 7.       begin
 8.       (E1, E2) := bisect (E);
 9.       T := C1 + C2·E1; "construct T such that C1 ⊆ T ⊆ C2"
10.       a := COUNT(T); "query the database"
11.       if a = #
12.         then begin "query unanswerable, so restrict C2"
13.           C2 := T;
14.           E := E1
15.           end
16.         else begin "query answerable, so extend C1"
17.           C1 := T;
18.           E := E2
19.           end
20.       end of while
21.   end of for
end of procedure "COUNT(C1) ≤ N − k < COUNT(C2) ≤ COUNT(C1) + g"
```

Fig. 6. Procedure to approximate $N - k$.

of $k = \lceil N/4 - (3/4)\sqrt{N} \rceil$ in the middle column corresponds to the case for which 99.7 percent of all definable sets of records may be expected to be general trackers when $g = 1$ [12]. As mentioned, the database experimented upon had a value of $g \geq 82$. This lessens the prospects of finding general trackers, but as the results show, it still remains a straightforward matter.

The mean number of queries before a double tracker (assuming $k = \lfloor n/4 \rfloor = 7866$) found in the experiments was 2.95 queries in variation (1) and exactly two queries both in variations (2) and (3); two queries are the theoretical minimum, since a double tracker consists of two (noncomplementary) record sets.

It is tempting to assume that the database employed might be especially suitable for tracker finding, but this is unlikely: The average number of attributes needed for tracker finding, as well as the numbers of values per attribute, were small, while the prospects for tracker finding increase with increasing numbers of attributes and values per attribute. The frequency distributions of the attributes were not unusual, ranging from rather skewed to approximately uniform distributions.

7. APPROXIMATING k OR $N - k$

An intruder using the procedure of Figure 2 must know the values of k and $N - k$ in order to know when a tracker T has been found (line 20). A database manager might hope to make it more difficult to find a tracker by concealing these values. However, a slight variation of the procedure of Figure 2 can be used to approximate k or $N - k$. Figure 6 shows a procedure for estimating $N - k$. When the procedure terminates, COUNT($C1$) gives a lower bound for $N - k$ that is accurate to within g; i.e.,

$$0 \leq (N - k) - \text{COUNT}(C1) < g.$$

If N is known, k can be approximated from the estimate for $N - k$; otherwise k can be approximated with these modifications to the procedure of Figure 6:

```
1   C1 := 0
2   C2 := C;
    .
    .
    .
11  if a ≠ ≠ "if query answerable, then restrict C2"
    .
    .
    .
```

In this case, when the procedure terminates, COUNT($C2$) gives an upper bound for k that is accurate to within g; i.e.,

$$0 \leq \text{COUNT}(C2) - k < g.$$

8. CONCLUSION

Our earlier research showed that trackers, once found, were easily used to calculate the answers to unanswerable queries. In this paper we have demonstrated that trackers are also easy to find.

We have assumed that an intruder has complete knowledge of the attributes and their possible values. This assumption is not unrealistic, for this information must be available if the users are to know what are the valid queries in the database. If only some of the attributes or their values are known, then finding a tracker may or may not be possible. For example, with $k = 2$ for the database of Table I, a tracker cannot be found using only the attribute SEX; some knowledge of at least one of the other attributes is needed.

A tracker may be found with incomplete knowledge of the values of the attributes provided that enough information is known to reduce the difference COUNT($C2$) to within $N - 4k$. This may be possible using attributes V_i whose values are numerically ordered but unknown. An intruder can estimate the range of values for V_i and then iteratively bisect the range until he reaches the precision believed to represent the data (in the worst case, this would be the precision of the machine).

The existence of trackers in a database does not necessarily imply that a particular individual can be compromised. There must be enough information recorded about him so that it is possible to isolate his record in the database. Experimental studies have revealed that there are databases in which most individuals are uniquely identified [11]. For these databases, compromise with trackers is a very serious threat.

ACKNOWLEDGMENTS

The authors are grateful to Peter Denning for his comments and suggstions, and to I. Borchert and B. Nieswandt for their cooperation in the experiment.

REFERENCES

1. CHIN, F.Y. Security in statistical databases for queries with small counts. *ACM Trans. Database Syst. 3*, 1 (March 1978), 92–104.
2. DALENIUS, T. Towards a methodology for statistical disclosure control. *Statistisk Tidskrift 15* (1977), 429–444.

3. DAVIDA, G.I., ET AL. Database security. *IEEE Trans. Software Eng. SE-4*, 6 (Nov. 1978), 531–533.
4. DEMILLO, R.A., DOBKIN, D., AND LIPTON, R.J. Even data bases that lie can be compromised. *IEEE Trans. Software Eng. SE-4*, 1 (Jan. 1978), 73–75.
5. DENNING, D.E. Are statistical databases secure? Proc. AFIPS 1978 NCC, Vol. 47, AFIPS Press, Arlington, Va., pp. 525–530.
6. DENNING, D.E., DENNING, P.J., AND SCHWARTZ, M.D. The tracker: A threat to statistical database security. *ACM Trans. Database Syst. 4*, 1 (March 1979), 76–96.
7. DOBKIN, D., JONES, A.K., AND LIPTON, R.J. Secure databases: Protection against user inference. *ACM Trans. Database Syst. 4*, 1 (March 1979), 97–106.
8. HOFFMAN, L.J., AND MILLER, W.F. Getting a personal dossier from a statistical data bank. *Datamation 16*, 5 (May 1970), 74–75.
9. KAM, J.B., AND ULLMAN, J.D. A model of statistical databases and their security. *ACM Trans. Database Syst. 2*, 1 (March 1977), 1–10.
10. REISS, S.B. Medians and database security. In *Foundations of Secure Computation*, R.A. DeMillo, D. Dobkin, A.K. Jones, and R.J. Lipton, Eds., Academic Press, New York, 1978, pp. 57–91.
11. SCHLÖRER, J. Identification and retrieval of personal records from a statistical data bank. *Methods Inform. in Medicine 14*, 1 (Jan. 1975), 7–13.
12. SCHLÖRER, J. Disclosure from statistical databases: Quantitative aspects of trackers. To appear in *ACM Trans. Database Syst.*
13. SCHLÖRER, J. Statistical database security: Some recent results. Presented at Medical Informatics, Berlin, 1979.
14. SCHWARTZ, M.D. Inference from statistical data bases. Ph.D. Th., Comptr. Sci. Dept., Purdue U., W. Lafayette, Ind., Aug. 1977.
15. SELBMANN, H.K. Bitstring processing for statistical evaluation of large volumes of medical data. *Methods Inform. in Medicine 13*, 1 (Jan. 1974), 61–64.
16. VAN EIMEREN, W., SELBMANN, H.K., AND ÜBERLA, K. Modell einer allgemeinen Vorsorgeuntersuchung im Jahre 1969/70—Schleussbericht. W. E. Weinmann Druckerei, Bonlanden b. Stuttgart, 1972.

Received March 1979; revised August 1979