Secure Statistical Databases with Random Sample Queries

DOROTHY E. DENNING
Purdue University

A new inference control, called random sample queries, is proposed for safeguarding confidential data in on-line statistical databases. The random sample queries control deals directly with the basic principle of compromise by making it impossible for a questioner to control precisely the formation of query sets. Queries for relative frequencies and averages are computed using random samples drawn from the query sets. The sampling strategy permits the release of accurate and timely statistics and can be implemented at very low cost. Analysis shows the relative error in the statistics decreases as the query set size increases; in contrast, the effort required to compromise increases with the query set size due to large absolute errors. Experiments performed on a simulated database support the analysis.

Key Words and Phrases: confidentiality, database security, disclosure controls, sampling, statistical database
CR Categories: 4.33

1. INTRODUCTION

Protecting confidential personal records in on-line, centralized databases from unauthorized disclosure or modification is a problem of wide interest. These systems may include access controls to protect records from unauthorized query or update, authentification schemes to certify the identities of users at terminals, information flow controls to restrict data to their allowed security levels, and encryption schemes to protect data while in transit through an insecure channel or while stored in an insecure medium [12].

None of these controls deals successfully with the *inference problem*—the deduction of confidential data by correlating the declassified statistical summaries and prior information. For example, comparing the mean salary of two groups differing only by a single record may reveal the salary of the individual whose record is in one group but not the other. The objective of inference controls is to make the cost of obtaining information in this way unacceptably high.

Census bureaus have dealt successfully with this problem for years. They remove from the database information that easily identifies an individual, e.g., social security numbers and exact geographical locations; they release statistics

This work was supported in part by the National Science Foundation under Grant MCS-77-04835.
Author's address: Department of Computer Science, Purdue University, West Lafayette, IN 47907.

drawn from only a small sample of the entire population [4, 20]. Unfortunately, these techniques do not work well in small or medium data management systems where records are added, deleted, or updated frequently. Modern relational database systems have powerful query languages which make it easy to request statistics about arbitrary subgroups of individuals. It has remained an open question whether inference can be controlled in such systems.

Most of the research in this area has studied efficient attacks rather than effective safeguards. With few exceptions, proposed inference controls are either easy to circumvent or impractical to implement (see [10, 11, 15, 33]). Despite its negative tone, this research is valuable because the nature of the threat must be understood before effective countermeasures can be built.

The common feature of all attacks is that the user can control which set of records is queried. This paper investigates a new class of queries, called random sample queries (RSQs), that deny the intruder precise control over the queried records. RSQs introduce enough uncertainty that users cannot isolate a confidential record but can get accurate statistics for groups of records.

We briefly review our model of statistical databases and methods of compromise in Sections 2 and 3 and then introduce random sample queries in Section 4. Section 5 discusses a possible implementation. Section 6 analyzes the errors in the statistics and compares them with the errors observed in experiments with a simulated database. Section 7 studies the ability of RSQs to withstand attack.

2. STATISTICAL DATABASE MODEL

A statistical database contains N confidential records. Each record contains M fields, where the jth field ($j = 1, \ldots, M$) contains a *data value* for the jth *attribute* (*variable, category*). An example of an attribute is SEX, whose two possible values are MALE and FEMALE. We assume the database is static; that is, records are not inserted, deleted, or updated.

Statistics are obtained through queries of the database. A *query* is given in terms of a *characteristic formula* C, which, informally, is any logical formula over the values using the operators *and* ($\cdot$), *or* ($+$), and *not* ($^-$). The set of records whose values match C is called the *query set* X_C of C. The simplest forms of raw statistics are counts and sums:

$$\text{COUNT}(C) = n_C,$$

where $n_C = |X_C|$ is the size of X_C, and

$$\text{SUM}(C, j) = \sum_{i \in X_C} v_{ij},$$

where v_{ij} is the value of field j in record i. Note that SUM queries apply only to numeric data (e.g., SALARY). The responses from COUNT and SUM queries are used to calculate relative frequencies and means:

$$\text{RFREQ}(C) = \frac{\text{COUNT}(C)}{N} = \frac{n_C}{N} \tag{1}$$

$$\text{AVG}(C, j) = \frac{\text{SUM}(C, j)}{\text{COUNT}(C)}.$$

More general forms can be defined; for example, the SUM query could be modified to add up terms like $(v_{ij})^k$, thereby providing the raw statistics for the kth moment. We will use $q(C)$ to denote any of these kinds of queries.

3. A REVIEW OF RESEARCH ON METHODS OF COMPROMISE

Compromise (or *disclosure*) occurs when a questioner deduces, from the responses of one or more queries, confidential information of which he was previously unaware [6]. Researchers have studied methods of controlling compromise but have found that each method succumbs to simple attack or is impractical to use.

Most of the attacks are based on isolating a single data element at the intersection of several query sets; the confidential value is obtained by solving a system of equations employing the responses of these queries. The defenses against these attacks are of four kinds: controls on the sizes of query sets; controls on the overlaps of query sets; distorting the data or the query responses; and sampling from the database. These controls will be reviewed briefly in the next sections.

3.1 Controls on the Sizes of Query Sets

The *minimum query size* control aims to defend against attacks employing very large or very small query sets, e.g., with a formula C that identifies a single record [5, 22]. Let k denote a parameter giving the lower bound on allowable query set size. A query $q(C)$ is not answered unless $k \leq n_C \leq N - k$. Unfortunately, this control is often easily subverted (even for k near $N/2$) by a simple snooping tool called the "tracker" [13, 14, 29, 31, 35]. A tracker is a set of characteristic formulas whose query sets pad the query set of the original formula to form answerable queries; the questioner subtracts out the effect of the tracker to determine the answer to the query for the original formula. Trackers are generally easy to find and apply. One of the most powerful trackers is the *general tracker*: a formula T such that $2k \leq n_T \leq N - 2k$ [13, 35]. Given an unanswerable query $q(C)$ and a tracker T, only a few queries are required to compute the answer to $q(C)$ from answerable queries which pad C with T. For example, when $n_C < k$, relative frequencies and averages can be computed from

$$\begin{aligned}
\mathrm{RFREQ}(C) &= \mathrm{RFREQ}(C + T) + \mathrm{RFREQ}(C + \bar{T}) - 1 \\
\mathrm{AVG}(C, j) &= [\mathrm{AVG}(C + T, j)\mathrm{RFREQ}(C + T) \\
&\quad + \mathrm{AVG}(C + \bar{T}, j)\mathrm{RFREQ}(C + \bar{T}) \\
&\quad - \mathrm{AVG}(T, j)\mathrm{RFREQ}(T) \\
&\quad - \mathrm{AVG}(\bar{T}, j)\mathrm{RFREQ}(\bar{T})]/\mathrm{RFREQ}(C).
\end{aligned} \tag{2}$$

Similar equations are used when $n_C > N - k$ (see [13]).

3.2 Controls on the Overlap of Query Sets

The *minimum overlap* control inhibits the responses from queries that have more than a predetermined number of records in common with each prior query [16]. No efficient implementation of this control is known: before responding, the

query program could have to compare the current query group against every previous one. This control may also be subverted by queries that overlap by small amounts (e.g., by solving a system of equations) [8, 9, 16, 23, 28, 29, 34, 36].

An effective method of preventing a clever intruder from isolating a record by overlapping queries is *partitioning* the database [37]. Records are stored in groups, each containing at least some predetermined number of records. Queries may apply to any set of groups, but never to subsets of records within any group. It is therefore impossible to isolate a record. A variant is called *microaggregation*: individuals are grouped to create many synthetic "average individuals"; statistics are computed for these synthetic individuals rather than the real ones [17]. Partitioning has two severe practical limitations in dynamic databases. First, the free flow of useful statistical information can be severely inhibited by excessively large groups or by ill-considered groupings. Second, forming and reforming groups as records are inserted, updated, and deleted from the database can lead to costly bookkeeping.

3.3 Distorting the Data or the Query Responses

The minimum query size control and minimum overlap control give exact answers when they respond. Rounding aims to prevent inference by perturbing the responses. Under *direct rounding*, the answer to a query is rounded up or down by some small amount before it is released [19, 20, 25, 27]. Rounding by adding a zero-mean random value (noise) is insecure since the correct answer can be deduced by averaging a sufficient number of responses to the same query. Rounding by adding a pseudorandom value that depends on the data is preferable, because then a given query always returns the same response. The method can sometimes be subverted with trackers [30] by adding dummy records to the database [24] or simply comparing the response to several queries in order to narrow the range of values containing the confidential value [1, 21].

A method of indirect rounding is called *error inoculation*; this control aims to prevent inference by perturbing or replacing the values stored in records [2–4]. Like direct rounding, this control attempts to trade accuracy in the statistics for security. One approach is to modify the data when the record is created (losing the original data); the problem with this approach is that correctness of the raw data may be essential for other uses of the data, e.g., storage and retrieval of patients' medical records. A better approach stores a "perturbation factor" in the record along with the original data and applies this factor when the data are used in a query [2].

A variation of error inoculation which may not disturb the accuracy of the statistics is *multidimensional transformation* or *data swapping*: the values of fields of records are exchanged so that the record for any particular individual is likely to be incorrect, but so that all i-order statistics are preserved for $i = 0, \ldots, m$ and some m (an i-order statistic is one derived from a characteristic formula over the values of i attributes); higher order statistics are not necessarily correct [7, 32]. Data swapping reduces the risk of compromise since there is no way of knowing with which individual a disclosed value is actually associated. The problem with the approach is that no efficient method for finding groups of records whose values can be swapped or of determining whether a valid swap even exists is known.

3.4 Random Samples

All the controls listed above are subverted by a single basic principle of compromise; because the questioner can control the composition of each query set, he can isolate a single record or value by intersecting query sets. Rounding and error inoculation perturb the responses, but the "noise" can often be removed by averaging responses for carefully selected query sets.

The U.S. Census Bureau has for years used the principle of *random sampling* to prevent inference. The questioner may apply responses to a set of records no longer selected by him. This prevents inference by depriving him of the ability to isolate a known record. The 1960 U.S. Census, for example, was distributed on tape as a random sample of one record in 1000 [20]. The best snooper would have at best a 1/1000 chance of associating a given sample record with the right individual.

Commercial data management systems now permit the construction of small- to medium-scale dynamic databases. A small fixed subsample would not be statistically significant and would not represent the current status of the data. For this reason, random sampling has been ignored as a possible inference control in modern statistical database systems.

The remainder of this paper shows that random sampling using *large* samples may effectively reduce risk but maintain high accuracy.

4. RANDOM SAMPLE QUERIES

Our proposal for random sampling differs in two important ways from the traditional statistical sampling methods used by the Census Bureau:

(1) To insure accurate statistics, each sample contains a large proportion of the records in the query set. To assure timely statistics, the sample is formed at the time a query is made.
(2) Instead of a query being applied to a sample of the entire database, a sample is formed from each query set. This enables implementation of the control at a very low cost.

The *random sample queries* (*RSQ*) control is defined as follows: As the query system locates records satisfying a given characteristic formula C, it applies a selection function $f(C, i)$ to each record i satisfying C; f determines whether i is kept for the sample. This produces a sampled query set $X_C^* = \{i \in X_C \mid f(C, i) = 1\}$. The statistic returned to the user is calculated from X_C^*. A parameter p specifies the sampling probability that a record is selected.

The uncertainty introduced by this control is the same as the uncertainty in sampling the entire database, with a probability p of selecting a particular record for the sample. The expected size of a random sample over the entire database of size N is pN.

5. IMPLEMENTATION

A simple case results when $p = 1 - \frac{1}{2}^k$ for some $k > 0$. Let $r(i)$ be a function which maps the ith record into a random sequence of $m \geq k$ bits. Let $g(C)$ be a function which maps formula C into a random sequence of length m over the alphabet $\{0, 1, *\}$; this string includes exactly k bits and $m - k$ asterisks (asterisks denote

"don't care"). The ith record is *excluded* from the sampled query set whenever $r(i)$ matches $g(C)$ (a "match" exists whenever each nonasterisk character of $g(C)$ is the same as the corresponding symbol of $r(i)$). The selection function $f(C, i)$ is thus given by

$$f(C, i) = \begin{cases} 1 & \text{if } r(i) \text{ does not match } g(C), \\ 0 & \text{if } r(i) \text{ matches } g(C). \end{cases}$$

The above method applies for $p > \frac{1}{2}$ (e.g., $p = 0.5$, 0.75, 0.875, and 0.9375). For $p < \frac{1}{2}$, use $p = \frac{1}{2}^k$; the ith record is *included* in the sample if and only if $r(i)$ matches $g(C)$.

Example. Suppose that $p = \frac{7}{8}$, that $m = 8$, and that $g(C) =$ "$*10*1***$". If $r(i)$ = "11011000" for some i, that record would match $g(C)$ and be excluded from X_C^*. If r generates unique random bit sequences, then the expected size of X_C^* is $\frac{7}{8}$ that of X_C.

Encryption algorithms, such as DES [26], are excellent candidates for the functions r and g, since they yield seemingly random bit sequences. If the database is encrypted for other security reasons, the function r could simply select m bits from some invariant part of the record (e.g., the identifier field); this would avoid the computation of $r(i)$ during query formation. With a good encryption algorithm, two formulas C and D having almost identical query sets will map to quite different $g(C)$ and $g(D)$, thereby ensuring that X_C^* and X_D^* differ by as much as they would if purely random sampling were being used.

Under RSQs, it is more natural to return relative frequencies and averages directly, as defined by eq. (1), since the statistics are not based on the entire database, and the users may not know what percentage of the records are included in the random samples. The sampled relative frequencies and means are

$$\text{RFREQ}^*(C) = \frac{n_C^*}{pN},$$

where $n_C^* = |X_C^*|$ is the sampled query set size, and

$$\text{AVG}^*(C, j) = \frac{1}{n_C^*} \sum_{i \in X_C^*} v_{ij}.$$

Note that the expected value of n_C^* is pn_C; therefore the expected value of the sampled frequency is n_C/N, the true frequency. Although the use of relative frequencies and averages in place of counts and sums is not required for security, security is enhanced due to the rounding errors introduced by division (provided not too many significant digits are provided). However, a user who knows p and N can compute approximations for both the sampled and unsampled counts and sums:

$$\begin{aligned} \text{COUNT}^*(C) &= \text{RFREQ}^*(C) \cdot pN \\ \text{SUM}^*(C, j) &= \text{AVG}^*(C, j) \cdot \text{COUNT}^*(C) \\ \text{COUNT}(C) &\simeq \text{RFREQ}^*(C) \cdot N \\ \text{SUM}(C, j) &\simeq \text{AVG}^*(C, j) \cdot \text{COUNT}(C). \end{aligned}$$

Indeed, it may be necessary for the database designers to publish the values for p and N so that users can judge the significance of the estimates returned.

A minimum query set size restriction may be necessary with RSQs if the sampling probability p is large. Otherwise, all the records of a small query set are included in a sample with high probability and compromise is possible (see Section 7). One alternative to this restriction is a variable p that decreases in proportion to the query set size. This could be implemented in at least three ways. The first method makes two passes over the data records: (1) to determine the query set size and select p, and (2) to calculate the response.

The second method calculates statistics for more than one value of p simultaneously, and selects one for the response after the query set size is known. The third method "guesses" an appropriate value for p by selecting p proportional to the reciprocal of the number of records scanned until the first record in the query set is found. The method best suited for a particular database would depend on the organization of the records in the database.

Ideally, the function g should use a normal form for formulas C, so that $g(C) = g(D)$ whenever formulas C and D are reducible to each other. This would prevent a questioner from determining the true answer to a query by repeatedly asking the same query, though expressed in different forms, and averaging the responses. Unfortunately, the problem of reducing a formula to a normal form is intractable; even if an efficient algorithm could be found, there are other methods for removing the sampling errors (see Section 7.3).

6. ANALYSIS OF ERRORS

RSQs control compromise by introducing small sampling errors into the statistics. The relative errors in frequencies are a function of the probability p of including a record in a sample and of the query set size. The relative errors in averages are a function of p, the query set size, and the distribution of values in the selected category field. Experimental results support the analysis.

6.1 Relative Frequencies

Let RFREQ*(C) be the response returned for a query RFREQ(C). The relative error between the sampled frequency and the true frequency is given by

$$f_C = \frac{\text{RFREQ}^*(C) - \text{RFREQ}(C)}{\text{RFREQ}(C)}.$$

Appendix A shows that the sampled relative frequency is an unbiased estimator of the true relative frequency; thus the expected relative error is zero. The root-mean-squared relative error is shown to be

$$\hat{R}(f_C) = \sqrt{\frac{1-p}{n_C p}} \tag{3}$$

for query set size n_C. Thus for fixed p, the expected error decreases as the square root of the query set size.

Figure 1 shows a graph of the error $\hat{R}(f_C)$ as a function of n_C for several values of p. For $p > 0.5$, $n_C > 100$ gives less than a 10 percent error. For $p = 0.9375$,

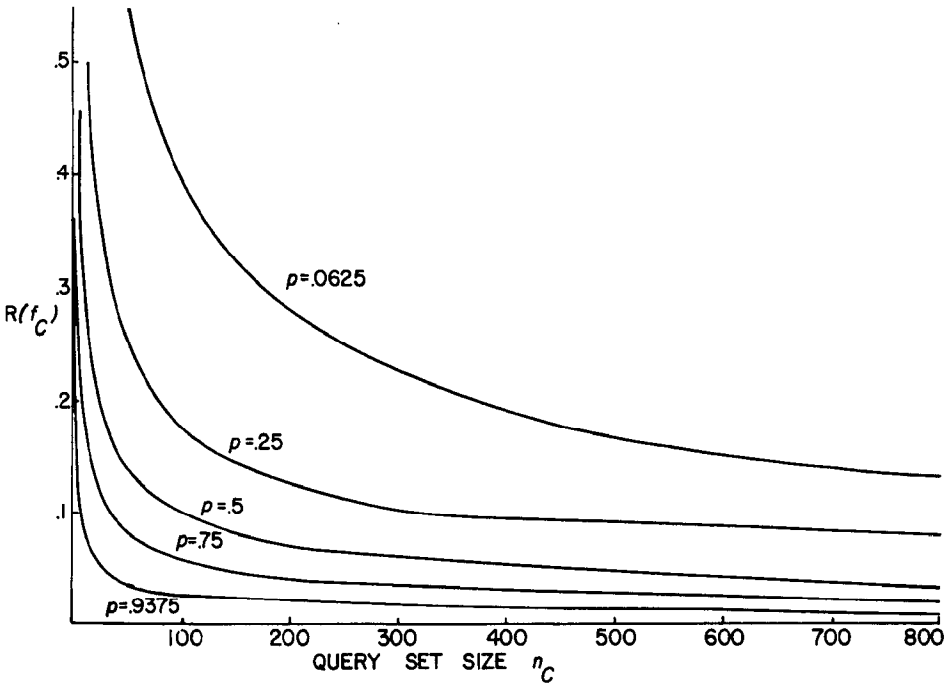

Fig. 1. Expected root-mean-squared relative error in frequency.

$n_C > 667$ gives < 1 percent error. Low relative errors are possible with high p even though query set sizes are relatively small.

However, for extremely small query sets, the relative errors may be unacceptably high. For example, for $p = 0.5$ and $n_C = 9$, $\hat{R}(f_C) = 0.33$. If a larger value of p is used for small query sets, then the relative errors decrease, but the risk of compromise increases (see Section 7). It may be preferable to impose a minimum query set size restriction than to release statistics with large errors.

Absolute errors for counts are greater than these for frequencies by a factor of N; however, their relative errors are comparable. The same is true for sums and averages.

6.2 Averages

Let AVG*(C, j) be the response returned for a query AVG(C, j). Let $E(x)$ and $Var(x)$ denote the mean and variance of the values of attribute j taken over the query set X_C; thus $E(x) = \text{AVG}(C, j)$. Appendix B shows that AVG*(C, j) is a biased estimator of the true average, where

$$E(\text{AVG}^*(C, j)) = E(x)[1 - (1 - p)^{n_C}].$$

For values of p of interest here ($p \geq 0.5$) and moderately large n_C ($n_C > 10$), the factor $[1 - (1 - p)^{n_C}]$ is negligible and can be ignored. Otherwise the response AVG*(C, j) can be divided by $[1 - (1 - p)^{n_C}]$ to yield an unbiased estimator.

The relative error between the sampled average and actual average is given by

$$a_{C,j} = \frac{\text{AVG}^*(C, j) - \text{AVG}(C, j)}{\text{AVG}(C, j)}.$$

Appendix B gives an exact formula for the root-mean-square relative error $\hat{R}(a_{C,j})$. For sufficiently large query set size n_C (the larger $|p - 0.5|$, the more asymmetric the distribution of the sampled query set size, and the higher the necessary n_C), $\hat{R}(a_{C,j})$ is approximately

$$\hat{R}(a_{C,j}) \simeq CV(x) \sqrt{\frac{1-p}{p(n_{C-1})}}$$
$$\simeq CV(x)\, \hat{R}(f_C)$$

where $CV(x) = (Var(x))^{1/2}/E(x)$ is the coefficient of variation for the distribution of data values.

As an example, suppose the data values for a category are uniformly distributed on $[1, s]$. The mean and variance for the query set are

$$E(x) = \frac{1}{s} \sum_{i=1}^{s} i = \frac{s+1}{2},$$

$$Var(x) = \frac{1}{s} \sum_{i=1}^{s} (i - E(x))^2 = \frac{s^2 - 1}{12}.$$

Thus

$$\hat{R}(a_{C,j}) \simeq D(s)\, \hat{R}(f_C) \tag{4}$$

where

$$D(s) = \frac{2}{s+1} \sqrt{\frac{s^2-1}{12}}.$$

The results discussed in the next section show that $\hat{R}(a_{C,j})$ closely approximates the actual errors observed in our experiments.

The function $D(s)$ rises rapidly and quickly approaches the limit:

$$\lim_{s \to \infty} D(s) = \sqrt{\tfrac{1}{3}}.$$

Thus for moderately large s ($s \geq 10$) and n_C,

$$\hat{R}(a_{C,j}) \simeq \sqrt{\tfrac{1}{3}}\, \hat{R}(f_C) \simeq 0.6\, \hat{R}(f_C).$$

When the data in a given category are uniformly distributed, the relative errors in averages behave the same as in frequencies but are 40 percent smaller.

6.3 Experimental Results

Random sample queries were tested on databases of size $N = 100$, $N = 500$, and $N = 1000$. The objective of the experiments was to measure the trade-off between the error in the statistics and the threat of compromise. Four values of p were used—0.5, 0.75, 0.875, and 0.9375, corresponding to specifications of between 1 and 4 bits, respectively, in the function $g(C)$. A pseudorandom number generator was used to create records for the database and to specify the functions r and g. Each record i had an 18-bit randomly generated ID field and several data fields; the ID field was used at the value of $r(i)$. The data fields were generated randomly over a uniform distribution.

Three hundred random characteristic formulas were used to measure the error

in the statistics. For each formula C, the experimental relative error in RFREQ*(C) and AVG*(C, j) (for all data fields j) were calculated. Errors were classified according to ten equal intervals of $[0, N]$. For each interval, the experimental absolute values of the relative errors and the root-mean-squared relative errors were calculated for frequencies and averages. For comparison, the theoretical root-mean-squared errors $\hat{R}(f_C)$ and $\hat{R}(a_{C,j})$ were also computed for an interval of the form $[k(N/10) + 1, (k + 1)(N/10)]$ using $n_C = [(N/10)(k + \frac{1}{2})]$ in eqs. (3) and (4).

The results are shown in Table I for $N = 100$ and $N = 1000$, and for $p = 0.5$ and $p = 0.9375$. Each table gives the experimental mean relative error, the experimental root-mean-squared relative error, and the theoretical root-mean-squared relative error for frequencies and averages. Averages are shown for a variable uniformly distributed in the range [1, 64]; thus using eq. (4),

$$\hat{R}(a_{C,j}) \simeq \frac{2}{64+1}\sqrt{\frac{64^2-1}{12}}\,\hat{R}(f_C)$$
$$\simeq \sqrt{\frac{21}{65}}\,\hat{R}(f_C).$$

The theoretical root-mean-squared relative errors closely approximate the experimental root-mean-squared errors. The approximation is not as close in the first interval since most of the actual query sets turned out to be smaller than the midpoint of the interval and since eqs. (3) and (4) hold only for large query sets. The mean relative errors are about 20 percent smaller than the root-mean-squared relative errors.

7. COMPROMISE

RSQs control compromise by reducing a questioner's ability to interrogate the desired query sets precisely. We have studied the extent to which the control may be circumvented by three different methods of attack: small query sets (of size 0 or 1), general trackers, and error removal by averaging. Compromise may be possible with small query sets unless p is small or a minimum query set size restriction is imposed. Trackers, on the other hand, are no longer a useful tool for compromise. Attacks based on removing the sampling errors by averaging responses require a large number of "equivalent" queries.

7.1 Small Query Sets (of Size 0 or 1)

Suppose that a questioner knows an individual represented in the database satisfying formula C. If RFREQ(C) = $1/N$, then the questioner can deduce whether or not that individual also has an additional property a by posing the query RFREQ($C \cdot a$) [22], since

$$\text{RFREQ}(C \cdot a) = \begin{cases} \frac{1}{N} \Rightarrow \text{the individual has property } a \\ 0 \Rightarrow \text{the individual does not have property } a. \end{cases}$$

This technique can be used to compromise under RSQs only if the questioner can infer with high probability that a response RFREQ*(C) = $1/N$ (or 0) implies

Table I. Mean Relative Error, Root-Mean-Squared Relative Error, and the Theoretical Root-Mean-Squared Relative Error

		RFREQ*(C)			AVG*(C, j)		
Query set size range	Number queries	Mean relative error	rms relative error	$\hat{R}(f_C)$	Mean relative error	rms relative error	$\hat{R}(a_{C,j})$
A. Frequencies and Averages for $N = 100$ and $p = 0.5$							
1–10	50	0.518	0.646	0.447	0.385	0.534	0.254
10–20	21	0.115	0.150	0.258	0.104	0.132	0.147
20–30	31	0.127	0.156	0.200	0.102	0.127	0.114
30–40	15	0.160	0.201	0.169	0.059	0.077	0.096
40–50	27	0.106	0.131	0.149	0.066	0.077	0.085
50–60	71	0.090	0.107	0.135	0.063	0.076	0.077
60–70	27	0.094	0.109	0.124	0.040	0.052	0.070
70–80	28	0.079	0.111	0.115	0.053	0.065	0.065
80–90	20	0.094	0.106	0.108	0.047	0.056	0.061
90–100	6	0.104	0.112	0.103	0.045	0.052	0.059
B. Frequencies and Averages for $N = 1000$ and $p = 0.5$							
1–100	40	0.232	0.348	0.141	0.082	0.117	0.080
100–200	24	0.060	0.073	0.082	0.037	0.048	0.047
200–300	25	0.047	0.058	0.063	0.021	0.027	0.036
300–400	11	0.031	0.037	0.053	0.030	0.034	0.030
400–500	32	0.039	0.047	0.047	0.025	0.029	0.027
500–600	65	0.034	0.044	0.043	0.021	0.026	0.024
600–700	33	0.034	0.043	0.039	0.019	0.023	0.022
700–800	43	0.032	0.037	0.037	0.015	0.019	0.021
800–900	26	0.024	0.029	0.034	0.016	0.018	0.019
900–1000	1	0.029	0	0.032	0.000	0	0.018
C. Frequencies and Averages for $N = 100$ and $p = 0.9375$							
1–10	39	0.079	0.102	0.115	0.019	0.081	0.065
10–20	27	0.053	0.065	0.067	0.029	0.045	0.038
20–30	25	0.041	0.049	0.052	0.018	0.025	0.030
30–40	9	0.025	0.037	0.044	0.017	0.024	0.025
40–50	35	0.030	0.035	0.038	0.017	0.023	0.022
50–60	56	0.029	0.035	0.035	0.012	0.016	0.020
60–70	34	0.030	0.036	0.032	0.015	0.019	0.018
70–80	32	0.020	0.024	0.030	0.012	0.016	0.017
80–90	27	0.021	0.025	0.028	0.013	0.018	0.016
90–100	12	0.016	0.019	0.026	0.011	0.015	0.015
D. Frequencies and Averages for $N = 1000$ and $p = 0.9375$							
1–100	48	0.042	0.059	0.037	0.013	0.021	0.021
100–200	18	0.022	0.027	0.021	0.010	0.013	0.012
200–300	30	0.012	0.015	0.016	0.008	0.011	0.009
300–400	11	0.011	0.013	0.014	0.006	0.008	0.008
400–500	30	0.008	0.010	0.012	0.006	0.007	0.007
500–600	75	0.009	0.011	0.011	0.006	0.007	0.006
600–700	28	0.008	0.010	0.010	0.005	0.006	0.006
700–800	37	0.007	0.008	0.009	0.004	0.005	0.005
800–900	18	0.008	0.010	0.009	0.004	0.005	0.005
900–1000	5	0.005	0.004	0.008	0.005	0.005	0.005

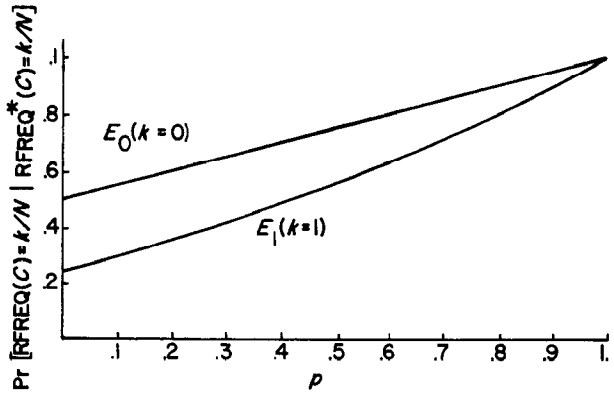

Fig. 2. Probabilities E_0 and E_1 that the sampling frequency is the true frequency as a function of p.

$\text{RFREQ}(C) = 1/N$ (or 0). In Appendix C we show that

$$E_1 = \Pr\left[\text{RFREQ}(C) = \frac{1}{N} \,\middle|\, \text{RFREQ}^*(C) = \frac{1}{N}\right] = \frac{a_1}{A'(1-p)}$$

$$E_0 = \Pr[\text{RFREQ}(C) = 0 \mid \text{RFREQ}^*(C) = 0] = \frac{a_0}{A(1-p)},$$

where $a_k (k = 0, \ldots, N) = \Pr[n_C = k]$ is the probability that C specifies a query set of size k,

$$A(z) = \sum_{k=0}^{N} a_k z^k$$

is the generating function for the distribution of $a_0, \ldots, a_N$, and $A'(z)$ is the derivative of $A(z)$.

As an example, suppose that the a_k are geometrically distributed with parameter λ, for $0 < \lambda < 1$. For large N, $a_k \simeq \lambda^k (1 - \lambda)$ (see Appendix C). The cumulative distribution function $A_k = \Pr[n_C \leq k]$ is given by

$$A_k = \sum_{j=0}^{k} a_j \simeq \sum_{j=0}^{k} \lambda^j(1-\lambda) = 1 - \lambda^{k+1}$$

Thus for $k \gg 1$, $A_k \simeq 1$; that is, most queries have small query sets. For $\lambda = 0.5$, the mean query set size is $\lambda/(1 - \lambda) = 1$. From Appendix C,

$$E_1 \simeq [1 - \lambda(1-p)]^2,$$

$$E_0 \simeq 1 - \lambda(1-p).$$

Figure 2 displays E_1 and E_0 for $\lambda = 0.5$ as a function of p. The odds are 50 percent that a response of zero is correct for all p and that a response of $1/N$ is correct for $p > 0.41$. For $p > 0.9$, the odds are 90 percent that a response of $1/N$ is correct and 95 percent that a response of zero is correct.

The conclusion is that inference of the true value of $\text{RFREQ}(C)$ is straightforward for large p; either a minimum query set size restriction or a p that diminishes with n_C must be used to prevent this.

Table II. Mean Absolute Relative Error in the Estimates for 50 Random Tracker Attacks Using $p = 0.9375$

N	Mean relative error for RFREQ(C)	Mean relative error for AVG(C, j)
100	2.22	4.42
500	4.48	5.89
1000	7.59	5.69

7.2 Trackers

Several random tracker compromises were attempted in the experimental databases of size $N = 100$, $N = 500$, and $N = 1000$. The target was a random individual uniquely identified by some formula C. A random tracker characterizing roughly half the database was constructed to estimate RFREQ(C) and $AVG(C, j)$ using eq. (2). Table II gives the mean relative error (not percentage) in the estimates for 50 random attacks using $p = 0.9375$ and the three values of N. The averages are given for a variable uniformly distributed over the range [1, 64]. For frequencies, the mean relative error in the estimates was over 700 percent for $N = 100$ and over 70 percent for $N = 1000$. Although the query errors decrease in N, the tracker errors actually increase in N since the absolute error using eq. (2) is magnified for larger N. The mean relative errors in averages were nearly 500 percent and seemed to be independent of N.

7.3 Error Removal

Since the same query always returns the same response, it is necessary to pose different but "equivalent" queries to remove the sampling errors. There are two methods for removing the error in the response to a query: (1) averaging the responses of several queries which specify the same query set, and (2) averaging estimates obtained from queries about disjoint subsets of a query set.

The first method averages the responses of m queries which specify the same query set but employ different random samples. Let $q(C)$ be a query for a frequency or average with response $q^*(C)$. The questioner poses queries of the form $q(C_i)$ $(i = 1, \ldots, m)$, where $X_{C_i} = X_C$ but $X^*_{C_i} \neq X^*_C$. An estimate $\hat{q}(C)$ for $q(C)$ is computed from

$$\hat{q}(C) = \frac{1}{m} \sum_{i=1}^{m} q^*(C_i).$$

Each query $q(C_i)$ could use a formula C_i which, though theoretically possible to reduce to C, is not reduced to C so that $g(C) \neq g(C_i)$. For example, if C = "MALE · (AGE $\geq$ 50 yrs)", C_1 might be "FEMALE · (AGE < 50 yrs)". Alternatively, C_i could be obtained by "ORing" into C terms which are known to specify empty query sets; that is, $C_i = C + D$, where $|X_D| = 0$. For example, if C is as before, C_2 might be "MALE · (AGE $\geq$ 50 yrs) + MALE · PREGNANT".

The second method averages m estimates for a query $q(C)$ using disjoint subsets of the query set X_C. The ith estimate, denoted $\hat{q}_i(C)$, is computed from

the responses to queries using formulas $C_{i1}, \ldots, C_{iz_i}$, where

$$X_C = \bigcup_{k=1}^{z_i} X_{C_{ik}}$$

and

$$X_{C_{ik}} \cap X_{C_{ik'}} \neq \varnothing$$

for $k \neq k'$.

The estimate $\hat{q}(C)$ for $q(C)$ is then obtained from the average:

$$\hat{q}(C) = \frac{1}{m} \sum_{i=1}^{m} \hat{q}_i(C).$$

For frequencies, the ith estimate is obtained by summing the responses:

$$\widehat{\text{RFREQ}}_i(C) = \sum_{k=1}^{z_i} \text{RFREQ}^*(C_{ik}).$$

For example, if C = "FEMALE", RFREQ(C) could be estimated from

$$\begin{aligned}
\widehat{\text{RFREQ}}_1(C) &= \text{RFREQ}^*(\text{FEMALE} \cdot \text{PREGNANT}) \\
&\quad + \text{RFREQ}^*(\text{FEMALE} \cdot \overline{\text{PREGNANT}}) \\
\widehat{\text{RFREQ}}_2(C) &= \text{RFREQ}^*(\text{FEMALE} \cdot (\text{AGE} < 20 \text{ yrs})) \\
&\quad + \text{RFREQ}^*(\text{FEMALE} \cdot (\text{AGE} \geq 20 \text{ yrs})) \\
&\vdots
\end{aligned}$$

Estimates for averages are similarly obtained by summing the products of responses for averages and frequencies.

Since the sampled query sets $X^*_{C_{ik}}$ used to obtain an estimate are independently selected from the disjoint query sets $X_{C_{ik}}$, and since the union of the $X^*_{C_{ik}}$ is a sample of X_C, the expected error in the estimate $\hat{q}_i(C)$ is the same as in a single response $q^*(C_j)$ for fixed p, where $X_{C_j} = X_C$. Therefore, the expected error in each estimate $\hat{q}_i(C)$ under the second method is the same as in a single response $q^*(C_j)$ under the first method, and the same number of estimates m must be averaged under the second method as responses under the first method to obtain the same level of confidence in the estimate $\hat{q}(C)$. However, the second method requires more queries since several queries are required to compute each estimate $\hat{q}_i(C)$. Furthermore, if p is inversely proportional to the query set size, then the second method requires still more queries since the expected errors are greater. Therefore, we shall analyze the number of queries required to compromise under the first method, as it provides a lower bound on m.

Let $F_1^*, \ldots, F_m^*$ be the responses for m independent queries which estimate RFREQ(C) for some C. Let $n_C = |X_C|$, and let

$$\hat{F} = \frac{1}{m} \sum_{i=1}^{m} F_i^*$$

be an approximation to the true value $F = \text{RFREQ}(C)$. From Appendix A, the mean and variance of $\hat{F}$ are

$$E(\hat{F}) = \frac{1}{m}\sum_{i=1}^{m} E(F_i^*) = \frac{1}{m}\left(m\frac{n_C}{N}\right) = \frac{n_C}{N},$$

$$Var(\hat{F}) = \frac{1}{m^2}\sum_{i=1}^{m} Var(F_i^*) = \frac{1}{m^2}\, m\frac{n_C(1-p)}{N^2 p} = \frac{n_C(1-p)}{m\,N^2 p}.$$

For large m ($m \geq 30$ should be sufficient when the distribution of possible responses for each F_i^* is symmetric), the distribution of $\hat{F}$ is approximately normal [18]. Letting $\sigma_{\hat{F}} = (Var(\hat{F}))^{1/2}$, the confidence intervals for the true frequency F given the estimate $\hat{F}$ are

$$\Pr[F \in [\hat{F} \pm 1.645\sigma_{\hat{F}}]] \simeq 0.90$$

$$\Pr[F \in [\hat{F} \pm 1.960\sigma_{\hat{F}}]] \simeq 0.95$$

$$\Pr[F \in [\hat{F} \pm 2.575\sigma_{\hat{F}}]] \simeq 0.99.$$

If we assume that an intruder requires a 95 percent confidence interval, the length of this interval is given by

$$I = 3.92\sigma_{\hat{F}} = \frac{3.92}{N}\sqrt{\frac{(1-p)n_C}{p\,m}}.$$

Now, $I \leq 1/N$ is required to estimate F to within one record (such accuracy is required, for example, to estimate relative frequencies for small query sets using trackers). The number of queries required to achieve this accuracy is

$$m \geq (3.92)^2\left(\frac{1-p}{p}\right)n_C > 15\left(\frac{1-p}{p}\right)n_C.$$

For fixed p, the function grows linearly in the query set size n_C. For $p = 0.5$, over 450 queries are required to estimate frequencies for query sets of size 30; over 1500 queries are required to estimate frequencies for query sets of size 100. For $p = 0.9375$, 100 queries are required to estimate frequencies for query sets of size 100.

According to the formula, only 10 queries are required to estimate frequencies for query sets of size 10. Although the formula is not accurate for query sets this small, it suggests that compromise may not be difficult for small query sets, especially if p is large. If a smaller value of p is used for small query sets, the risk of compromise is reduced, but the relative errors in the statistics are increased (see Section 6.1). The best approach may be a minimum query set size restriction.

Next, let $A_1^*, \ldots, A_m^*$ be the responses for m independent queries which estimate $\text{AVG}(C, j)$. Let $A = \text{AVG}(C, j)$, and let $E(x)$ and $Var(x)$ denote the mean and variance of the data values in category j for the records in the query set X_C (i.e., $E(x) = A$). Let

$$\hat{A} = \frac{1}{m}\sum_{i=1}^{m} A_i^*$$

be an estimate of the true average $\hat{A}$. From Appendix B, the mean of $\hat{A}$ is

$$E(\hat{A}) = \frac{1}{m}\sum_{i=1}^{m} E(A_i^*) = \frac{1}{m}(m\,E(x)) = E(x),$$

and the variance of $\hat{A}$ can be approximated with

$$Var(\hat{A}) = \frac{1}{m^2}\sum_{i=1}^{m} Var(A_i^*) \simeq \frac{1}{m^2}(m\,Var(x))\frac{1-p}{p(n_C-1)} = \frac{Var(x)(1-p)}{m\,p(n_C-1)}.$$

For large m and n_C, the distribution of $\hat{A}$ is approximately normal. Letting $\sigma_{\hat{A}} = (\mathrm{Var}(\hat{A}))^{1/2}$, the 95 percent confidence interval is defined by

$$\Pr[A \in [\hat{A} \pm 1.960\sigma_{\hat{A}}]] \simeq 0.95.$$

The length of this interval is given by

$$I = 3.92\sigma_{\hat{A}} \geq 3.92\ Var(x)\sqrt{\frac{1-p}{mp\,n_C}}.$$

Now $I \leq 2H\,E(x)$ is sufficient to estimate A with a relative error of at most H for $0 < H \leq 1$. Solving the above equation for m,

$$m > (1.96)^2 \frac{Var(x)}{E^2(x)}\frac{1-p}{H^2 p\,n_C} \tag{5}$$

queries must be made to obtain an estimate with relative error at most H.

To determine a bound on the relative error H that can be tolerated to achieve compromise, suppose that estimates for averages are used in the simplest form of attack: the tracker. Let D be a characteristic uniquely identifying an individual, and consider an estimate for $\mathrm{AVG}(D, j)$ for some category j using eq. (2). (We assume that a minimum query set size restriction is in effect so that the query $\mathrm{AVG}(D, j)$ is not directly answerable.) Rewriting eq. (2) we have

$$\begin{aligned}\mathrm{AVG}(D,j) = {} & \mathrm{AVG}(D+T,j)n_{D+T} \\ & + \mathrm{AVG}(D+\bar{T},j)n_{D+\bar{T}} - \mathrm{AVG}(T,j)n_T - \mathrm{AVG}(\bar{T},j)n_{\bar{T}}.\end{aligned}$$

Since we are interested in determining the number of estimates required for a single AVG query, suppose that all of the terms on the right-hand side of the above equation are known exactly except for one AVG. (This will also give a worst-case analysis of the threat.) Let $A_C = \mathrm{AVG}(C, j)$ represent the unknown AVG and let $A_D = \mathrm{AVG}(D, j)$. The relative error in the estimate $\hat{A}_D$ is given by

$$\frac{\hat{A}_D - A_D}{A_D} = \frac{(\hat{A}_C - A_C)n_C}{A_D}.$$

The estimate A_D will have a relative error $\leq h$, for $0 < h \leq 1$ if

$$\frac{|\hat{A}_C - A_C|\,n_C}{|A_D|} \leq h$$

or

$$\frac{|\hat{A}_C - A_C|}{|A_C|} \leq \frac{h|A_D|}{n_C|A_C|}.$$

Therefore, a relative error of at most

$$H = \left(\frac{h}{n_C}\right)\left|\frac{A_D}{A_C}\right|$$

in the estimate $\hat{A}_C$ is necessary to obtain an estimate $\hat{A}_D$ with relative error at most h. Substituting for H in eq. (5) gives

$$m > (1.96)^2 \frac{Var(x)}{E^2(x)}\left(\frac{1-p}{p}\right)\left(\frac{n_C}{h^2}\right)\left|\frac{A_C}{A_D}\right|^2.$$

As an example, consider the special case where the data values are uniformly distributed over an interval $[1, s]$. The coefficient of variation squared is (see Section 6.2)

$$D^2(s) = \frac{s^2-1}{12}\left(\frac{2}{s+1}\right)^2.$$

In Section 6.2 we showed that $D^2(s)$ is approximately $\frac{1}{3}$ for moderately large s (e.g., $s \geq 10$); thus

$$m > 1.28\left(\frac{1-p}{p}\right)\left(\frac{n_C}{h^2}\right)\left|\frac{A_C}{A_D}\right|^2$$

estimates are needed. For $h = 0.1$ and A_D near the average, this is

$$m > 128\left(\frac{1-p}{p}\right)n_C.$$

For fixed p, m grows linearly in the query set size n_C. For $n_C = 100$, over 853 estimates are required for $p = 0.9375$ and over 12,800 for $p = 0.5$. In a database of size 20,000 if a tracker is used which characterizes roughly half of the population, over 85,300 estimates of the averages are required for $p = 0.9375$ and over 1,280,000 for $p = 0.5$. For $h = 0.01$, the number of estimates needed is increased by a factor of 100. If A_D is much smaller than the average A_C, even more queries are required to obtain a good estimate; however, if A_D is larger than A_C, fewer queries are required. Whereas the relative errors in averages (for uniform distributions) are lower than in frequencies, more queries are required to obtain estimates accurate enough to compromise with averages than with frequencies.

For large query sets, the number of queries required to obtain reliable estimates of confidential data under RSQs is sufficiently large to protect against manual attack using trackers. A computer might be able to subvert the control by systematically generating the necessary queries. To prevent computer-aided attacks, the system should recognize queries which specify identical query sets. To the extent that characteristic formulas are reduced to normal form before processing, the threat is reduced since the same random sample will be selected and, therefore, the same response returned. The threat can be eliminated entirely with two passes over the query set. The first pass computes the function $g(C)$ (see Section 5) from the records in the query set X_C ($g(C)$ could be a function of the ID fields of the records); the second pass uses $g(C)$ to select records for the sample. However, this does not handle the case where a query $g(C)$ is estimated

from queries about disjoint subsets of X_C; threat monitoring may be necessary to detect this type of systematic attack [22].

8. CONCLUSIONS

The random sample queries control proposed here deals directly with the basic principle of compromise by making it impossible for a questioner to control precisely the composition of query sets. Queries for relative frequencies and averages are computed using random samples drawn from the query sets. To ensure accurate and timely statistics, each sample contains a large proportion of the records in the query set and is formed at the time a query is made. As the query system locates records satisfying a characteristic formula C, a selection function which is dependent on C determines whether or not each record is kept for the sample. A parameter p specifies the sampling probability that a record is selected. The cost of implementing the control is extremely low.

For both relative frequencies and averages, the relative error in the statistics decreases as the square root of the query set size. In contrast, the effort required to compromise by removing the sampling errors increases linearly in the query set size owing to larger absolute errors. Therefore, statistics based on large groups are both more accurate and less susceptible to compromise than statistics based on small groups. A minimum query set size restriction can control compromise with small query sets. For frequencies and averages taken over uniform distributions, relative errors between 1 and 10 percent can be obtained for allowable queries, while an enormous number of "equivalent" queries must be posed in order to compromise by removing the sampling errors.

APPENDIX A. ERRORS IN ESTIMATING RELATIVE FREQUENCIES

Let RFREQ(C) be a query for a frequency and let RFREQ*(C) be the sampled frequency. Let n_C denote the size of the query set X_C, and let n_C^* denote the size of the sample X_C^*. Then n_C^* is binomially distributed with parameter p:

$$\Pr[n_C^* = k] = \binom{n_C}{k} p^k (1-p)^{n_C - k}.$$

The mean and variance of the distribution are

$$E(n_C^*) = n_C p$$

$$Var(n_C^*) = n_C p(1-p).$$

Letting F_C^* denote the response RFREQ*(C) $= n_C^*/pN$, the mean and variance of F_C^* are

$$E(F_C^*) = \frac{n_C}{N}$$

$$Var(F_C^*) = \frac{n_C(1-p)}{N^2 p}.$$

Since $E(F_C^*) =$ RFREQ(C), the sampled frequency is an unbiased estimator of the true frequency.

Let

$$f_C^2 = \left(\frac{\text{RFREQ}^*(C) - \text{RFREQ}(C)}{\text{RFREQ}(C)}\right)^2 = \left(\frac{(n_C^*/pN) - (n_C/N)}{n_C N}\right)^2$$

be the squared relative error in RFREQ*(C). The mean-squared relative error (over all choices of the sample) is

$$E(f_C^2) = \frac{1}{(n_C/N)^2}(Var(F_C^*)) = \frac{1}{(n_C/N)^2}\left(\frac{n_C(1-p)}{N^2 p}\right)$$
$$= \frac{1-p}{n_C p}.$$

Thus the root-mean-squared relative error is

$$\hat{R}(f_C) = \sqrt{\frac{1-p}{n_C p}}.$$

APPENDIX B. ERRORS IN ESTIMATING AVERAGES

Let AVG(C, j) be a query for the average value in category j, and let AVG*(C, j) be the sampled average. Let n_C denote the size of the query set X_C, let n_C^* denote the size of the sample X_C^*, and let $\{x_1, \ldots, x_{n_C}\}$ denote the values $\{v_{ij} \mid i \in X_C\}$. Let $E(x)$ and $Var(x)$ be the mean and variance of $\{x_1, \ldots, x_{n_C}\}$:

$$E(x) = \frac{1}{n_C}\sum_{i=1}^{n_C} x_i = \text{AVG}(C, j),$$

$$Var(x) = \frac{1}{n_C}\sum_{i=1}^{n_C}(x_i - E(x))^2.$$

Let $A_{C,j}^*$ denote the response AVG*(C, j); the expected value of $A_{C,j}^*$ is

$$E(A_{C,j}^*) = \sum_{k=0}^{n_C} E(A_{C,j}^*(k))\Pr[n_C^* = k], \tag{B1}$$

where $E(A_{C,j}^*(k))$ is the expected response when $n_C^* = k$. For $k > 0$,

$$E(A_{C,j}^*(k)) = \frac{1}{\binom{n_C}{k}} \sum_{\substack{A \subseteq X_C \\ |A| = k}} \left(\frac{1}{k}\sum_{i \in A} x_i\right).$$

Since each x_i appears in $\binom{n_C - 1}{k - 1}$ of the $\binom{n_C}{k}$ distinct possibilities for A, we have

$$E(A_{C,j}^*(k)) = \frac{1}{\binom{n_C}{k}}\frac{1}{k}\binom{n_C - 1}{k - 1}\sum_{i=1}^{n_C} x_i$$

$$= \frac{1}{\binom{n_C}{k}}\frac{n_C E(x)}{k}\binom{n_C - 1}{k - 1} = E(x).$$

For $k = 0$, we assume the response is 0; that is, $E(A_{C,j}^*(0)) = 0$. Substituting in eq. (B1) gives us

$$E(A_{C,j}^*) = \sum_{k=1}^{n_C} E(x)\Pr[n_C^* = k] = E(x)(1 - (1-p)^{n_C}).$$

The sampled average is thus a biased estimator of the true average. For the values of p of interest here ($p \geq 0.5$) and moderately large n_C ($n_C > 10$), this factor is negligible and can be ignored. To determine the variance of AVG*(C, j), we first evaluate the sum of the squares; for $k > 1$

$$G(k, n_C) = \sum_{\substack{A \subseteq X_C \\ |A|=k}} \left(\sum_{i \in A} x_i \right)^2$$

$$= \sum_{\substack{A \subseteq X_C \\ |A|=k}} \left(\sum_{i \in A} \sum_{j \in A} x_i x_j \right)$$

$$= \sum_{\substack{A \subseteq X_C \\ |A|=k}} \left(\sum_{i \in A} x_i^2 + \sum_{i \in A} \sum_{\substack{j \in A \\ j \neq i}} x_i x_j \right).$$

Since each x_i appears in $\binom{n_C-1}{k-1}$ of the possibilities for A and each pair $x_i x_j$ $(j \neq i)$ appears in $\binom{n_C-2}{k-2}$ of the possibilities for A, we have

$$G(k, n_C) = \binom{n_C - 1}{k - 1} \sum_{i=1}^{n_C} x_i^2 + \binom{n_C - 2}{k - 2} \sum_{i=1}^{n_C} \sum_{\substack{j=1 \\ j \neq i}}^{n_C} x_i x_j$$

$$= n_C E(x^2) \binom{n_C - 1}{k - 1} + \binom{n_C - 2}{k - 2} \sum_{i=1}^{n_C} x_i \left(\sum_{j=1}^{n_C} x_j - x_i \right)$$

$$= n_C E(x^2) \binom{n_C - 1}{k - 1} + \binom{n_C - 2}{k - 2} \left[\left(\sum_{i=1}^{n_C} x_i \right)^2 - \sum_{i=1}^{n_C} x_i^2 \right]$$

$$= n_C E(x^2) \binom{n_C - 1}{k - 1} + [(n_C E(x))^2 - n_C E(x^2)] \binom{n_C - 2}{k - 2}$$

$$= n_C E(x^2) \binom{n_C - 2}{k - 1} + (n_C E(x))^2 \binom{n_C - 2}{k - 2}.$$

The variance in AVG*(C, j) is then

$$Var(A^*_{C,j}) = \sum_{k=0}^{n_C} Var(A^*_{C,j}(k)) \Pr[n^*_C = k]$$

where $Var(A^*_{C,j}(k))$ is the variance in AVG*(C, j) when $n^*_C = k$. For $k > 1$,

$$Var(A^*_{C,j}(k)) = \frac{1}{\binom{n_C}{k}} \sum_{\substack{A \subseteq X_C \\ |A|=k}} \left(\frac{1}{k} \sum_{i \in A} x_i - E(x) \right)^2$$

$$= \frac{1}{\binom{n_C}{k}} \left[\sum_{\substack{A \subseteq X_C \\ |A|=k}} E^2(x) - \frac{2E(x)}{k} \sum_{\substack{A \subseteq X_C \\ |A|=k}} \sum_{i \in A} x_i + \frac{1}{k^2} \sum_{\substack{A \subseteq X_C \\ |A|=k}} \left(\sum_{i \in A} x_i \right)^2 \right].$$

Substituting $G(k, n_C)$ for the last term, this becomes

$$\begin{aligned}
&\frac{1}{\binom{n_C}{k}}\left[E^2(X)\binom{n_C}{k} - \frac{2n_C E^2(x)}{k}\binom{n_C-1}{k-1} + \frac{1}{k^2}G(k, n_C)\right] \\
&= \left[\frac{1}{k^2\binom{n_C}{k}}\right]G(k, n_C) - E^2(x) \\
&= \frac{n_C E(x^2)\binom{n_C-1}{k-1}}{k^2\binom{n_C}{k}} + \frac{(n_C E(x))^2\binom{n_C-2}{k-2}}{k^2\binom{n_C}{k}} - E^2(x) \\
&= \frac{n_C - k}{k(n_C-1)}E(x^2) - \frac{n_C-k}{k(n_C-1)}E^2(x) = \frac{n_C-k}{k(n_C-1)}Var(x).
\end{aligned} \tag{B2}$$

For $k = 1$,

$$\begin{aligned}
Var(A^*_{C,j}(1)) &= \frac{1}{n_C}\sum_{\substack{A\subseteq X_C \\ |A|=1}}\left(\sum_{i\in A}x_i - E(x)\right)^2 \\
&= \frac{1}{n_C}\sum_{i=1}^{n_C}(x_i - E(x))^2 = Var(x)
\end{aligned}$$

which is the same as would be obtained by substituting $k = 1$ in eq. (B2). For $k = 0$, we assume as before the response is 0; therefore,

$$Var(A^*_{C,j}(0)) = 0.$$

We thus have

$$\begin{aligned}
Var(A^*_{C,j}) &= Var(A^*_{C,j}(0))\Pr[n^*_C = 0] + \sum_{k=1}^{n_C} Var(A^*_{C,j}(k))\Pr[n^*_C = k] \\
&= Var(x)\sum_{k=1}^{n_C}\frac{n_C-k}{k(n_C-1)}\binom{n_C}{k}p^k(1-p)^{n_C-k}.
\end{aligned} \tag{B3}$$

Expression (B3) is not easily evaluated; an approximation is useful. Because the distribution of n^*_C is approximately normal with mean $E(n^*_C) = n_C p$, $Var(A^*_{C,j}(n_C p))$ is a reasonable approximation of $Var(A^*_{C,j})$. In fact, this approximation is a lower bound. We can rewrite eq. (B3) as

$$Var(A^*_{C,j}) = Var(x)\sum_{k=1}^{n_C} f(n_C, k)\Pr[n^*_C = k]$$

where

$$f(n_C, k) = \frac{n_C - k}{k(n_C-1)}.$$

Since $f(n_C, k)$ is concave up for $1 \le k \le n_C$,

$$\sum_{k=1}^{n_C} f(n_C, k)\Pr[n^*_C = k] > f\left(n_C, \sum_{k=1}^{n_C} k\Pr[n^*_C = k]\right).$$

The rightmost summation is the definition of $E(n_C^*)$, which is $n_C p$. Thus

$$Var(A_{C,j}^*) > Var(x) f(n_C, n_C p)$$
$$= Var(x) \frac{1-p}{p(n_C - 1)}.$$

Let

$$a_{C,j}^2 = \left(\frac{\text{AVG}^*(C,j) - \text{AVG}(C,j)}{\text{AVG}(C,j)}\right)^2 = \left(\frac{\text{AVG}^*(C,j) - E(x)}{E(x)}\right)^2$$

be the squared relative error in $\text{AVG}^*(C, j)$. The mean-squared relative error (over all choices of the sample) is

$$E(a_{C,j}^2) = \left(\frac{1}{E^2(x)}\right) Var(A_{C,j}^*) \simeq \frac{Var(x)}{E^2(x)} \left(\frac{1-p}{p(n_C - 1)}\right).$$

Thus the root-mean-squared relative error is approximated by

$$\hat{R}(a_{C,j}) \simeq CV(x) \sqrt{\frac{1-p}{p(n_C - 1)}}$$

where $CV(x) = (Var(x))^{1/2}/E(x)$ is the coefficient of variation for the distribution of x.

APPENDIX C. COMPROMISE WITH SMALL QUERY SETS

Let a_k (for $k = 0, \ldots, N$) be the probability that C specifies a query set size of k, and let

$$A(z) = \sum_{k=0}^{N} a_k z^k,$$
$$A'(z) = \sum_{k=1}^{N} a_k k z^{k-1},$$

be the generating function and its derivative for the distribution $a_0, \ldots, a_N$. Let F denote $\text{RFREQ}(C)$ and F^* denote $\text{RFREQ}^*(C)$. If the sampled frequency F^* is $1/N$, the probability that the true frequency F is also $1/N$ is given by

$$\Pr\left[F = \frac{1}{N} \mid F^* = \frac{1}{N}\right] = \frac{\Pr[F = 1/N \text{ and } F^* = 1/N]}{\sum_{k=1}^{N} \Pr[F^* = 1/N \mid F = k/N] a_k}$$
$$= \frac{p a_1}{\sum_{k=1}^{N} k p (1-p)^{k-1} a_k} = \frac{a_1}{A'(1-p)}.$$

If the sampled frequency F^* is 0, the probability that the true frequency F is also 0 is given by

$$\Pr[F = 0 \mid F^* = 0] = \frac{\Pr[F = 0 \text{ and } F^* = 0]}{\Pr[F^* = 0]}$$
$$= \frac{a_0}{\sum_{k=0}^{N} (1-p)^k a_k} = \frac{a_0}{A(1-p)}.$$

Consider the special case where the a_k are geometrically distributed with parameter λ for $0 < \lambda < 1$ (see Section 7.1). Then

$$a_k = \frac{\lambda^k(1-\lambda)}{1-\lambda^{N+1}}$$

and

$$A(z) = \left(\frac{1-\lambda}{1-\lambda^{N+1}}\right)\left(\frac{1-(\lambda z)^{N+1}}{1-\lambda z}\right),$$

$$A'(z) = \left(\frac{1-\lambda}{1-\lambda^{N+1}}\right)\left(\frac{-(N+1)\lambda(\lambda z)^N(1-\lambda z)+(1-(\lambda z)^{N+1})\lambda}{(1-\lambda z)^2}\right).$$

For large N,

$$a_k \simeq \lambda^k(1-\lambda).$$

Thus

$$a_1 \simeq \lambda(1-\lambda),$$

$$A'(1-p) = \left(\frac{1-\lambda}{1-\lambda^{N+1}}\right) \cdot \left(\frac{(-N+1)\lambda[\lambda(1-p)]^N[1-\lambda(1-p)]+(1-[\lambda(1-p)]^{N+1})\lambda}{[1-\lambda(1-p)]^2}\right)$$

$$\simeq \frac{(1-\lambda)\lambda}{[1-\lambda(1-p)]^2},$$

giving

$$\Pr\left[F = \frac{1}{N} \mid F^* = \frac{1}{N}\right] = \frac{a_1}{A'(1-p)} \simeq [1-\lambda(1-p)]^2.$$

Similarly, for large N,

$$a_0 \simeq (1-\lambda)$$

and

$$A(1-p) = \left(\frac{1-\lambda}{1-\lambda^{N+1}}\right)\left(\frac{1-[\lambda(1-p)]^{N+1}}{1-\lambda(1-p)}\right) \simeq \frac{1-\lambda}{1-\lambda(1-p)}.$$

Therefore,

$$\Pr[F = 0 \mid F^* = 0] = \frac{a_0}{A(1-p)} \simeq 1-\lambda(1-p).$$

ACKNOWLEDGMENTS

The author is deeply grateful to P. Denning for his help with the analysis and for providing numerous editorial suggestions, and to J. Schlörer for suggesting the worst-case analysis of compromise by removing the sampling errors and for

noting a serious problem with the original proposal. The author is also grateful to J. Schlörer and F. Chin for carefully reading this paper and offering many helpful suggestions.

REFERENCES

1. ACHUGBUE, J. O., AND CHIN, F. Y. Output perturbation for protection of statistical data bases. Dep. Computing Science, Univ. Alberta, Alberta, Canada, Jan. 1978.
2. BECK, L. L. A security mechanism for statistical databases. *ACM Trans. Database Syst. 5*, 3 (Sept. 1980), 316–338.
3. BORUCH, R. F. Maintaining confidentiality in educational research: A systematic analysis. *Am. Psychol. 26* (1971), 413–430.
4. CAMPBELL, D. T., BORUCH, R. F., SCHWARTZ, R. D., AND STEINBERG, J. Confidentiality-preserving modes of access to files and to interfile exchange for useful statistical analysis. *Eval. Quart. 1*, 2 (May 1977), 269–299.
5. CHIN, F. Y. Security in statistical databases for queries with small counts. *ACM Trans. Database Syst. 3*, 1 (March 1978), 92–104.
6. DALENIUS, T. Towards a methodology for statistical disclosure control. *Särtryck ur Statistisk tidskrift 15* (1977), 429–444.
7. DALENIUS, T., AND REISS, S. P. Data-swapping—A technique for disclosure control. Confidentiality in Surveys, Rep. 31, Dep. Stat., Univ. Stockholm, Stockholm, Sweden, May 1978.
8. DAVIDA, G. I., ET AL. Data base security. *IEEE Trans. Softw. Eng. SE-4*, 6 (Nov. 1978), 531–533.
9. DEMILLO, R. A., DOBKIN, D., AND LIPTON, R. J. Even data bases that lie can be compromised. *IEEE Trans. Softw. Eng. SE-4*, 1 (Jan. 1978), 73–75.
10. DENNING, D. E. A review of research on statistical database security. In *Foundations of Secure Computation*, R. A. DeMillo et al., Eds. Academic, New York, 1978.
11. DENNING, D. E. Are statistical data bases secure? *Proc. AFIPS 1978 NCC*, vol. 47, AFIPS Press, Arlington, Va., pp. 525–530.
12. DENNING, D. E., AND DENNING, P. J. Data security. *Comput. Surv. 11*, 3 (Sept. 1979), 227–249.
13. 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.
14. DENNING, D. E., AND SCHLÖRER, J. A fast procedure for finding a tracker in a statistical database. *ACM Trans. Database Syst. 5*, 1 (March 1980), 88–102.
15. DENNING, D. E. Complexity results relating to statistical confidentiality. Computer Science and Statistics: 12th Ann. Symp. Interface, Waterloo, Canada, May 1979, pp. 252–256.
16. DOBKIN, D., JONES, A. K., AND LIPTON, R. J. Secure databases: Protection against user influence. *ACM Trans. Database Syst. 4*, 1 (March 1979), 97–106.
17. FEIGE, E. L., AND WATTS, H. W. Protection of privacy through microaggregation. In *Data Bases, Computers, and the Social Sciences*, R. L. Bisco, Ed. Wiley-Interscience, New York, 1970.
18. FELLER, W. *An Introduction to Probability Theory and Its Applications I.* Wiley, New York, 1950.
19. FELLEGI, I. P., AND PHILLIPS, J. L. Statistical confidentiality: Some theory and applications to data dissemination. *Ann. Econ. Soc. Meas. 3*, 2 (April 1974), 399–409.
20. HANSEN, M. H. Insuring confidentiality of individual records in data storage and retrieval for statistical purposes. *Proc. AFIPS 1971 FJCC*, vol. 39, AFIPS Press, Arlington, Va., pp. 579–585.
21. HAQ, M. I. On safeguarding statistical disclosure by giving approximate answers to queries. *Int. Computing Symp.*, 1977, pp. 491–495.
22. HOFFMAN, L. J., AND MILLER, W. F. Getting a personal dossier from a statistical data bank. *Datamation 16*, 5 (May 1970), 74–75.
23. 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.
24. KARPINSKI, R. H. Reply to Hoffman and Shaw. *Datamation 16*, 10 (Oct. 1970), 11.
25. NARGUNDKAR, M. S., AND SAVELAND, W. Random rounding to prevent statistical disclosure. *Proc. Am. Stat. Assoc., Soc. Stat. Sect.* (1972), 382–385.
26. NATIONAL BUREAU OF STANDARDS. Data encryption standard. FIPS PUB. 46, Washington, D.C., Jan. 1977.

27. REED, I. S. Information theory and privacy in data banks. *Proc. AFIPS 1973*, vol. 42, AFIPS Press, Arlington, Va., pp. 581–587.
28. REISS, S. B. Medians and database security. In *Foundations of Secure Computation*, R. A. DeMillo et al., Eds. Academic, New York, 1978.
29. SCHLÖRER, J. Identification and retrieval of personal records from a statistical data bank. *Methods Inform. Med. 14*, 1 (Jan. 1975), 7–13.
30. SCHLÖRER, J. Confidentiality and security in statistical data banks. In *Data Documentation: Some Principles and Applications in Science and Industry*, W. Guas and R. Henzler, Eds. Proc. Workshop Data Documentation, 1975, Verl. Dok., Munchen, 1977, pp. 101–123.
31. SCHLÖRER, J. Disclosure from statistical databases: Quantitative aspects of trackers. Inst. Medizinische Statistik und Dokumentation, Univ. Giessen, Giessen, W. Germany, Mar. 1979. To appear in *ACM Trans. Database Syst.*
32. SCHLÖRER, J. Security of statistical databases: Multidimensional transformation. Rep. TB-IMSD 2/78, Inst. Medizinische Statistik und Dokumentation, Univ. Giessen, Giessen, W. Germany, Mar. 1979.
33. SCHLÖRER, J. Statistical database security: Some recent results. Inst. Medizinische Statistik und Dokumentation, Univ. Giessen, Giessen, W. Germany, 1979. Presented at Medical Informatics, Berlin, 1979.
34. SCHWARTZ, M. D., DENNING, D. E., AND DENNING, P. J. Securing data bases under linear queries. *Proc. IFIP Congress 77*, North-Holland, Amsterdam, 1977, pp. 395–398.
35. SCHWARTZ, M. D. Inference from statistical data bases. Ph.D. Dissertation, Dep. Computer Sciences, Purdue Univ., W. Lafayette, Ind., Aug. 1977.
36. SCHWARTZ, M. D., DENNING, D. E., AND DENNING, P. J. Linear queries in statistical databases. *ACM Trans. Database Syst. 4*, 2 (June 1979), 156–167.
37. YU, C. T., AND CHIN, F. Y. A study on the protection of statistical data bases. *ACM SIGMOD Int. Conf. Management of Data*, 1977, pp. 169–181.

Received April 1979; revised December 1979; accepted February 1980