1979

Secure Statistical Databases with Random Sample Queries

Dorothy E. Denning

Report Number:
79-302

SECURE STATISTICAL DATABASES WITH RANDOM SAMPLE QUERIES[1]

Dorothy E. Denning [2]

Purdue University

April 1979

CSD-TR-302

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 precisely control the formation of query sets. Queries for 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 larger absolute errors. Experiments performed on a simulated database support the analysis.

---

(1) Work reported herein was supported in part by NSF Grant MCS77-04835.

(2) Author's present address: Computer Science Dept., Purdue University, W. Lafayette, IN 47907.

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 [DENN79a].

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 drawn from only a small sample of the entire population [CAMP77, HANS71]. 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 DENN78a, DENN78b, DENN79d, and SCHL79c) 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 uncertainity 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 several fields and each field contains a data value for some attribute (or category). An example of an attribute is SEX, whose two possible values are MALE and FEMALE.

Statistics are obtained through queries of the database. A query is given in terms of a characteristic formula C, which is any logical formula over the values using the operators and (·), or (+), and not ($\bar{}$). 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:

COUNT(C) = n_C, where $n_C = |X_C|$ is the size of X_C, and

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:

$$\begin{aligned} &\text{FREQ(C)} = \text{COUNT(C)} / N = n_C / N \\ &\text{AVG(C, j)} = \text{SUM(C, j)} / \text{COUNT(C)}. \end{aligned} \tag{1}$$

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 k^{th} 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 [DALE77]. Researchers have studied methods of controlling compromise, but have found that each 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
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 [CHIN77, HOFF70]. 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" [DENN79b, DENN79c, SCHL75, SCHL79a, SCHW77b]. 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 questioner subtracts out the effect of the auxiliary characteristics to determine the answer to the query for the original characteristics. Trackers are generally easy to find and apply. One of the most powerful is the general tracker: a formula T such that $2k \leq n_T$

$\leq$ N - 2k [DENN79b, SCHW77b]. 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 $n_C < k$, frequencies and averages can be computed from:

$$\begin{aligned} FREQ(C) &= FRFQ(C + T) + FREQ(C + \overline{T}) - 1 \\ AVG(C, j) &= [AVG(C + T, j)\cdot FREQ(C + T) + AVG(C + \overline{T}, j)\cdot FREQ(C + \overline{T}) \\ &\quad - AVG(T, j)\cdot FREQ(T) - AVG(\overline{T}, j)\cdot FREQ(\overline{T})]/ FREQ(C) \end{aligned} \quad (2)$$

Similar equations are used when $n_C > N - k$ (see DENN79b).

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 [DOBK79]. 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) [DAVI76, DOBK79, KAM77, RFIS78, SCHL75, SCHW77a, SCHW79].

An effective method of preventing a clever intruder from isolating a record by overlapping queries is partitioning the database [YU78]. 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 [FEIG70]. 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 [HANS71, FELL74, NARG72, REED73]. 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 pseudo-random 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 [SCHL77], by adding dummy records to the database [KARP70], or simply by comparing the responses to several queries in order to narrow the range of values containing the confidential value [ACHU78, HAQ77].

A method of indirect rounding is called error inoculation; this control aims to prevent inference by perturbing or replacing the values stored in records [BECK79, BORU71, CAMP77]. 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 permanent "perturbation factor" in the record along with the original data, and applies this factor when the data is used in a query [BECK79].

A variation of error inocolution 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,...,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 [DALE78, SCHL79b]. 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 [HANS71]. 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 sub-sample 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. Rather than applying a query 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 $p = 1 - 1/2^k$ for some $k > 0$. Let r(i) be a function which maps the i^{th} record into a random sequence of m 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 i^{th} record is excluded from the sampled query set whenever r(i) matches g(C) (a "match" exists whenever each non-asterisk 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 $p > 1/2$ (e.g., p = .5, .75, .875, and .9375). For $p < 1/2$, use $p = 1/2^k$; the i^{th} record is included in the sample if and only if r(i) matches g(C).

Example. Suppose that p = 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 7/8 that of X_C.

Strong encryption algorithms are excellent candidates for the functions r and g. 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 strong encryption, two formulae 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 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 frequencies and means are:

$$\text{FREQ}^*(C) = \frac{n_C^*}{pN}, \text{ where } n_C^* = |X_C^*| \text{ is the sampled query set size,}$$

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

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 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:

$$\text{COUNT}^*(C) = \text{FREQ}^*(C) \cdot pN$$

$$\text{SUM}^*(C,j) = \text{AVG}^*(C,j) \cdot \text{COUNT}^*(C)$$

$$\text{COUNT}(C) \simeq \text{FREQ}^*(C) \cdot N$$

$$\text{SUM}(C,j) \simeq \text{AVG}^*(C,j) \cdot \text{COUNT}(C).$$

Indeed, it may be necessary for the database to provide 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 one of 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.

Ideally, the function g should use a normal form for formulae C, so that g(C) = g(D) whenever formulae C and D are reducible to eachother. 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 Frequencies

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

$$f_C = \frac{FREQ^*(C) - FREQ(C)}{FREQ(C)}$$

Appendix A shows that the sampled frequency is an unbiased estimator of the true frequency; thus the expected relative error is 0. The expected root mean squared relative error is shown to be:

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

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 $p > .5$, $n_C > 100$ gives less than a 10% error. For $p = .9375$, $n_C > 667$ gives $< 1\%$ error. Low relative errors are possible with high p even though query set sizes are relatively small.

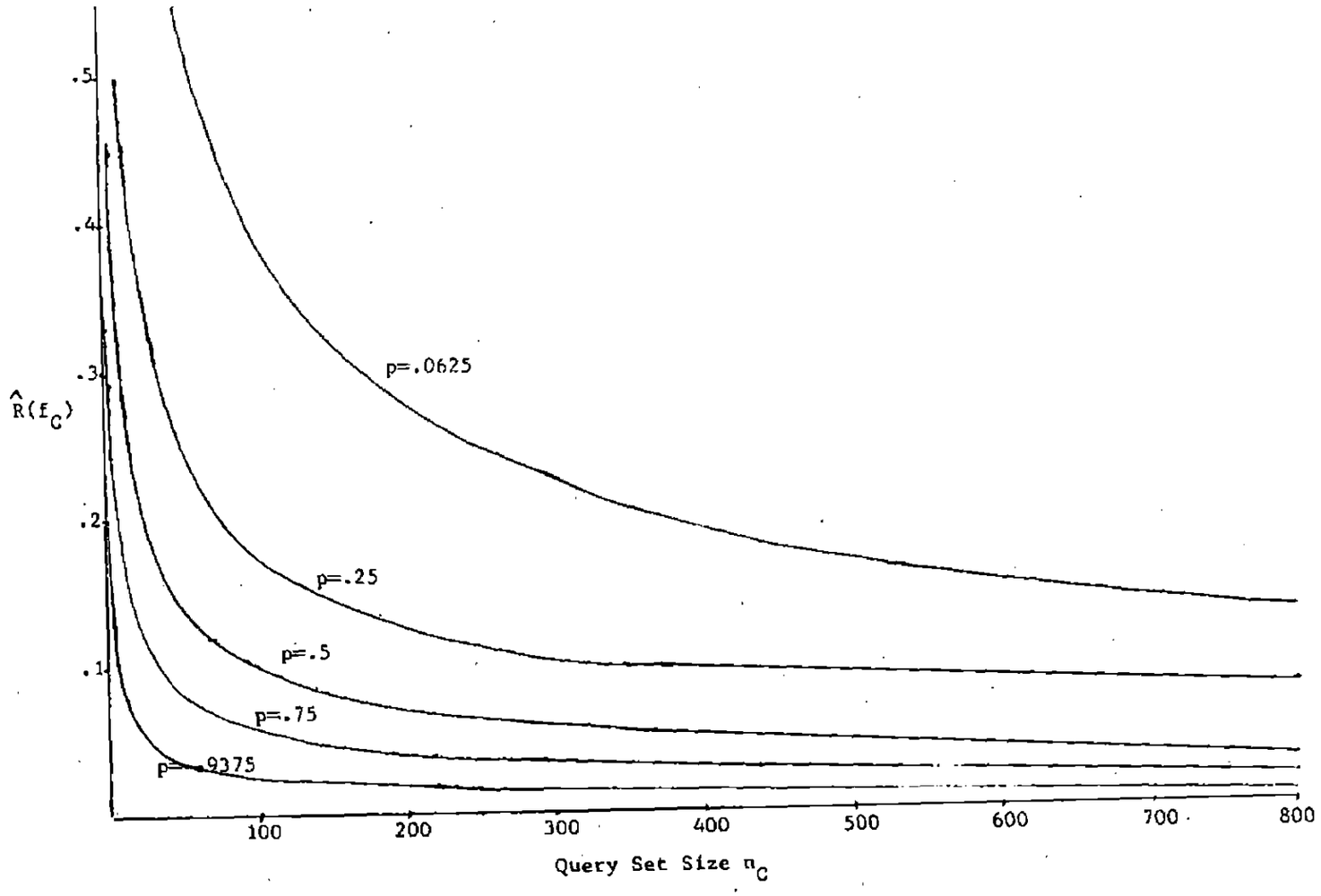

Figure 1. Expected root mean squared relative error in frequency.

6.2 Averages

Let $AVG^*(C,j)$ be the response returned for a query AVG(C,j). The relative error between the sampled average and the actual average is given by:

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

Appendix B shows that $AVG^*(C,j)$ is an unbiased estimator of AVG(C,j), that the expected relative error is 0, and that the expected root mean square relative error can be approximated by:

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

for query set size n_C, where $\bar{x}$ and σ_x are the mean and standard deviation of the data values in category j taken over the query set X_C.

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:

$$\bar{x} = \frac{1}{s} \sum_{i=1}^{s} i = \frac{s + 1}{2}$$

$$\sigma_x^2 = \frac{1}{s} \sum_{i=1}^{s} (i - \bar{x})^2 = \frac{s^2 - 1}{12}$$

Thus,

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

where

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

is the coefficient of variation for the distribution of data values.

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

Figure 2 shows the function D(s) rises rapidly and quickly approaches the limit:

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

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

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

When the data in a given category are uniformly distributed, the relative errors in averages behave the same as in frequencies but are 40% 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 tradeoff between the error in the statistics and the threat of compromise. Four values of p were used -- .5, .75, .875, .9375, corresponding to specifications of between 1 and 4 bits respectively in the function g(C). A pseudo-random 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 as the value of r(i). The data fields were generated randomly over a uniform distribution.

Three hundred random characteristic formulae were used to measure the error in the statistics. For each formula C, the relative errors in $\text{FREQ}^*(C)$ and $\text{AVG}^*(C,j)$ (for all data fields j) were calculated. Errors were classified according to 10 equal intervals of [0, N]. For each interval, the mean absolute relative errors and the root mean squared relative errors were calculated for frequencies and averages. For comparison, the expected root

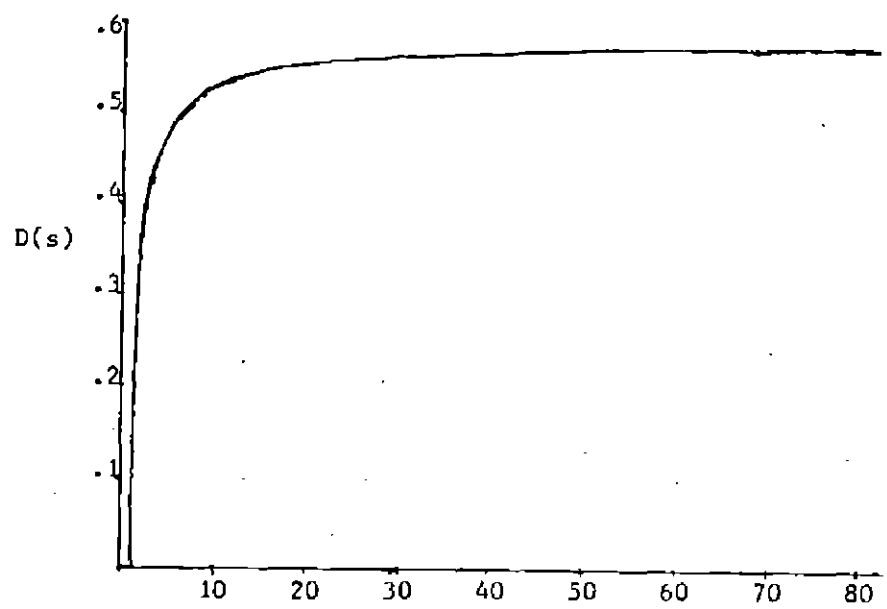

Figure 2. Coefficient of variation D(s) in computation of the root mean squared relative error in the averages for uniform distributions over the interval [1,s].

mean squared errors were also computed for an interval of the form $[k\frac{N}{10}, (k+1)\frac{N}{10}]$ using n_c at the midpoint $\frac{N}{10}(k+\frac{1}{2})$ for formulae (3) and (4).

The results are shown in Table I (a)-(d) for N = 100 and N = 1000, and for p = .5 and p = .9375. Each table gives the mean absolute relative error, the root mean squared relative error, and the expected root mean squared relative error for frequencies and averages. Averages are shown for a variable uniformly distributed in the range [1, 64]. The estimated root mean squared relative errors closely approximate the actual root mean squared errors. The approximation is not as close in the first interval since most of the actual query sets were much smaller than the midpoint of the interval. The mean absolute relative errors are about 20% smaller than the root mean squared relative errors.

Query Set Size Range	Number Queries	FREQ*(C) Mean Abs Rel.Err	FREQ*(C) R.M.Sq. Rel.Err	$\hat{R}(f_C)$	AVG*(C,j) Mean Abs Rel.Err	AVG*(C,j) R.M.Sq. Rel.Err	$\hat{R}(a_{C,j})$
1- 10	50	0.518	0.646	0.447	0.385	0.534	0.284
10- 20	21	0.115	0.150	0.258	0.104	0.132	0.152
20- 30	31	0.127	0.156	0.200	0.102	0.127	0.116
20- 40	15	0.160	0.201	0.169	0.059	0.077	0.097
40- 50	27	0.106	0.131	0.149	0.066	0.077	0.086
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	0.071
70- 80	28	0.079	0.111	0.115	0.053	0.065	0.066
80- 90	20	0.094	0.106	0.108	0.047	0.056	0.062
90-100	6	0.104	0.112	0.103	0.045	0.052	0.059

Table I (a). Mean absolute relative error, root mean squared relative error, and the expected root mean squared relative error for frequencies and averages for N = 100 and p = .5.

Query Set Size Range	Number Queries	FREQ*(C) Mean Abs Rel.Err	FREQ*(C) R.M.Sq. Rel.Err	$\hat{R}(f_C)$	AVG*(C,j) Mean Abs Rel.Err	AVG*(C,j) R.M.Sq. Rel.Err	$\hat{R}(a_{C,j})$
1- 100	40	0.232	0.348	0.141	0.082	0.117	0.081
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	0.020
900-1000	1	0.029	0	0.032	0.000	0	0.018

Table I (b). Mean absolute relative error, root mean squared relative error, and the expected root mean squared relative error for frequencies and averages for N = 1000 and p = .5.

Query Set Size Range	Number Queries	FREQ*(C) Mean Abs Rel.Err	FREQ*(C) R.M.Sq. Rel.Err	$\hat{R}(f_C)$	AVG*(C,j) Mean Abs Rel.Err	AVG*(C,j) R.M.Sq. Rel.Err	$\hat{R}(a_{C,j})$
1- 10	39	0.079	0.102	0.115	0.019	0.081	0.073
10- 20	27	0.053	0.065	0.067	0.029	0.045	0.039
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

Table I (c). Mean absolute relative error, root mean squared relative error, and the expected root mean squared relative error for frequencies and averages for N = 100 and p = .9375.

Query Set Size Range	Number Queries	FREQ*(C) Mean Abs Rel.Err	FREQ*(C) R.M.Sq. Rel.Err	$\hat{R}(f_C)$	AVG*(C,j) Mean Abs Rel.Err	AVG*(C,j) R.M.Sq. Rel.Err	$\hat{R}(a_{C,j})$
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

Table I (d). Mean absolute relative error, root mean squared relative error, and the expected root mean squared relative error for frequencies and averages for N = 1000 and p = .9375.

7. COMPROMISE

RSQs control compormise 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 romoval 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 satisfying formula C. If FREQ(C) = 1/N, then the questioner can deduce whether or not that individual also has an additional property "a" by posing the query FREQ(C·a) [HOFF70], since

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

This technique can be used to compromise under RSQs only if the questioner can infer with high probability that a response $\text{FREQ}^*(C) = 1/N$ (or 0) implies FREQ(C) = 1/N (or 0). In appendix C, we show that

$$E1 = \Pr[\text{FREQ}(C) = 1/N \mid \text{FREQ}^*(C) = 1/N] = \frac{a_1}{A'(1-p)}$$

$$E2 = \Pr[\text{FREQ}(C) = 0 \mid \text{FREQ}^*(C) = 0] = \frac{a_{0.}}{A(1-p)}$$

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

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

is the generating function for the distribution of $a_0, \ldots, a_N$.

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), and

$$E1 = [1 - \lambda(1-p)]^2$$

$$E0 = 1 - \lambda(1-p)$$

Figure 3 shows a graph of the cumulative distribution function $A_k = Pr[n_C \geq k] = a_0 + \ldots + a_k$ for $\lambda = .5$ (corresponding to mean query set size of $\lambda/(1-\lambda) = 1$). Figure 4 displays E1 and E0 for $\lambda = .5$ as a function of p. The odds are 50% that a response of 0 is correct for all p and that a response of 1/N is correct for $p > .41$. For $p > .9$, the odds are 90% that a response of 1/N is correct and 95% that a response of 0 is correct.

The conclusion is that inference of the true value of FREQ(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.

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 FREQ(C) and AVG(C,j)

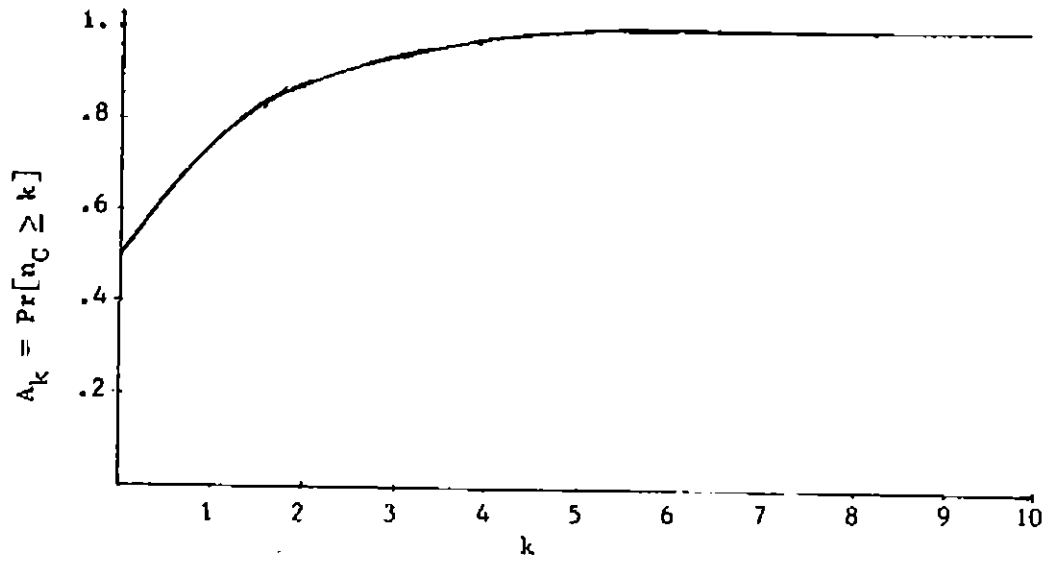

Figure 3. Cumulative distribution A_k of query set size for a geometric distribution with parameter $\lambda = .5$.

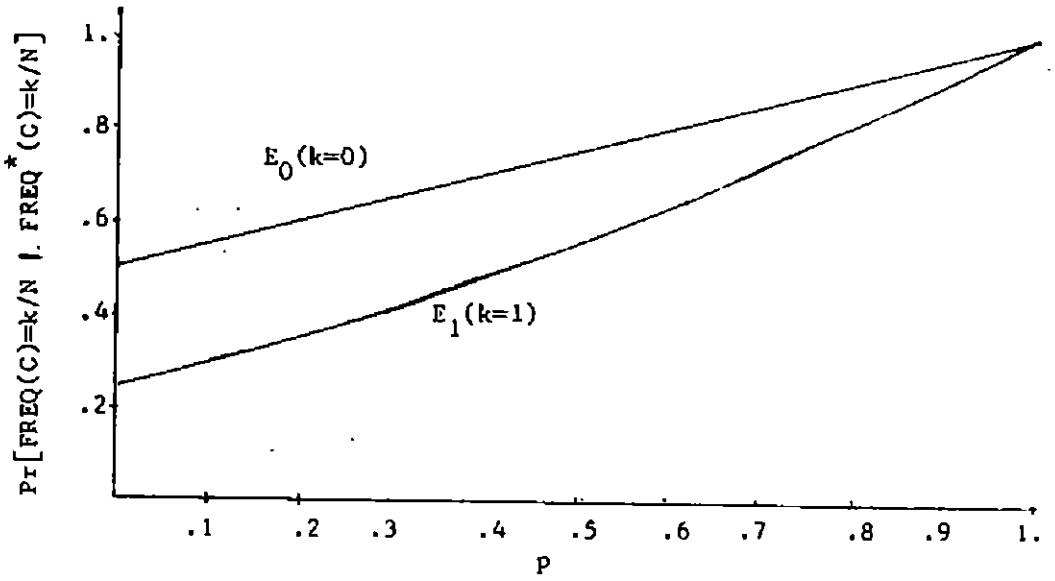

Figure 4. Probabilities E_0 and E_1 that the sampling frequency is the true frequency as a function of p.

using eq. (2). Table II gives the mean absolute relative error in the estimates for 50 random attacks using p = .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 absolute relative error in the estimates was over 200% for N = 100 and over 700% 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 absolute relative errors in averages were nearly 500% and seemed to be independent of N.

For comparison, the same tracker attacks were also studied under a simple rounding control (rather than RSQs). When rounding errors were comparable to the RSQ errors, the tracker attacks were more likely to subvert the rounding control than the random sample control. In many cases, the tracker revealed the exact value despite the rounding control.

N	Mean Rel. Err. for FREQ(C)	Mean Rel. Err. for AVG(C,j)
100	2.22	4.42
500	4.48	5.89
1000	7.59	5.69

Table II. Mean absolute relative error in the estimates for 50 random tracker attacks using p = .9375.

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

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

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 $\geq$ 50yrs)", C_1 might be "$\overline{\text{FEMALE}}\cdot\overline{(\text{AGE} < 50\text{yrs})}$". Alternatively, C_i could be obtained by "OR-ing" 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 $\geq$ 50yrs) + MALE·PREGNANT".

The second method averages m estimates for a query q(C) using disjoint subsets of the query set X_C. The i^{th} estimate, denoted $\hat{q}_i(C)$, is computed from the responses to queries using formulae $C_{i1},\ldots, C_{iz_i}$, where

$$X_C = \bigcup_{k=1}^{z_i} X_{C_{ik}} \quad \text{and} \quad X_{C_{ik}} \cap X_{C_{ik'}} = \emptyset \quad \text{for } k \neq k' \ .$$

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

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

For frequencies, the i^{th} estimate is obtained by summing the responses:

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

For example, if C = "FFMALE", FREQ(C) could be estimated from:

$$\widehat{FREQ}_1(C) = FREQ^*(FEMALE \cdot PREGNANT) + FREQ^*(FEMALE \cdot \overline{PREGNANT})$$

$$\widehat{FREQ}_2(C) = FREQ^*(FEMALE \cdot (AGE < 20yrs)) + FREQ^*(FEMALE \cdot (AGE \geq 20yrs))$$

etc.

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 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 $q^*(C_j)$, 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 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 $\hat{q}_i(C)$. 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 FREQ(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 = FRFQ(C). From Appendix A, the mean and variance of F are:

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

$$\sigma^2_{\hat{F}} = \frac{1}{m^2} \sum_{i=1}^{m} \mathrm{Var}(F_i^*) = \frac{1}{m^2} \; m \; \frac{n_C (1-p)}{N^2 p} = \frac{n_C (1-p)}{m N^2 p}$$

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 [FELL50]; therefore, the confidence intervals for the true frequency F given the estimate $\hat{F}$ are:

$$\Pr[F \in [\hat{F} \pm 1.645 \sigma_{\hat{F}}]] \simeq .90$$

$$\Pr[F \in [\hat{F} \pm 1.9600 \sigma_{\hat{F}}]] \simeq .95$$

$$\Pr[F \in [\hat{F} \pm 2.575 \sigma_{\hat{F}}]] \simeq .99$$

If we assume that an intruder requires a 95% 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 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 = .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 = .9375, 100 queries are required to estimate frequencies for query sets of size 100.

Next, let $A_1^*, \ldots, A_m^*$ be the responses for m independent queries which estimate AVG(C, j). Let A = AVG(C, j), and let $\bar{x}$ and σ_x^2 denote the mean and variance of the data values in category j for the records in the query set X_C (i.e., $\bar{x} = A$). Let

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

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

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

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

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

For large m and n_C, the distribution of $\hat{A}$ is approximately normal, whereupon the 95% confidence interval is defined by:

$$\Pr[A \in [\hat{A} \pm 1.960\, \sigma_{\hat{A}}]] \simeq .95$$

The length of this interval is given by

$$I = 3.92\sigma_{\hat{A}} \simeq 3.92\,\sigma_x \sqrt{\frac{1-p}{mp\,n_C}}$$

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

$$m > (1.96)^2 \frac{\sigma_x^2}{\bar{x}^2} \frac{1 - p}{H^2 p\, n_C} \qquad (5)$$

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 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 AVG(D,j) is not directly answerable). Rewriting eq.(2), we have:

$$AVG(D,j) = AVG(D+T,j)n_{D+T} + AVG(D+\bar{T},j)n_{D+\bar{T}} - AVG(T,j)n_T - AVG(\bar{T},j)n_{\bar{T}} .$$

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 $A_C = AVG(C,j)$ represent the unknown AVG and let $A_D = AVG(D,j)$. 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 $\hat{A}_D$ will have a relative error $\leq h$, for $0 \leq h \leq 1$ 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 (5) gives:

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

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):

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

In Section 6.2, we showed that $D^2(s)$ is approximately 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 = .1 and A_D near the average, this is

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

For n_C = 100, over 853 estimates are required for p = .9375 and over 12,800 for p = .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 = .9375 and over 1,280,000 for p = .5. For h = .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 formulae 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. Although it is more difficult to recognize queries about disjoint subsets of a query set, a larger number of queries are needed to obtain reliable estimates. Furthermore, it should not be difficult to detect this type of systematic attack with "threat-monitoring" [HOFF70].

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 frequencies and averages are computed using random samples drawn from the query sets. To insure 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 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 due to larger absolute errors. Therefore, statistics based on large groups are both more accurate and less susceptable 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% 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.

ACKNOWLEDGEMENTS

I am deeply grateful to Peter Denning for helping me with the analysis and for providing numerous editorial suggestions, and to Jan Schlörer for suggesting the worst-case analysis of compromise by removing the sampling errors and for noting a serious problem with my original proposal.

REFERENCES

ACHU78 Achugbue, J. O. and Chin, F. Y., "Output Perturbation for Protection of Statistical Data Bases, "Dept. of Computing Science, Univ. of Alberta (Jan. 1978).

BECK79 Beck, L. L., "A Security Mechanism for Statistical Databases," Dept. of Comp. Sci. and Eng., Southern Methodist Univ., (Jan. 1979).

BORU71 Boruch, R. F., "Maintaining Confidentiality in Educational Research: A Systematic Analysis," Amer. Psychologist 26, (1971), 413-430.

CAMP77 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," Evaluation Quarterly 1, 2 (May 1977), 269-299.

CHIN77 Chin, F. Y., "Security in Statistical Data Bases for Queries with Small Counts," ACM TODS 3, 1 (1978), 92-104.

DALE77 Dalenius, T., "Towards a Methodology for Statistical Disclosure Control," Sartryck ur Statistisk tidskrift 15 (1977), 429-444.

DALE78 Dalenius, T. and Reiss, S. P., "Data-Swapping -- A Technique for Disclosure Control," Confidentiality in Surveys, Report No. 31 (May 1978), Dept. Statist., Univ. Stockholm.

DAVI76 Davida, G. I. et al., "Data Base Security," TR-CS-76-14, Dept. of EE and Computer Science, Univ. of Wisconsin, Milwaukee (July 1976).

DEMI77 DeMillo, R. A., Dobkin, D. and Lipton, R. J., "Even Data Bases that Lie Can be Compromised," IEEE Trans. on Software Design SE-4 1 (Jan. 1977), 73-75.

DENN78a Denning, D. E., "A Review of Research on Statistical Database Security," in Foundations of Secure Computation (DeMillo et al. ed.), Academic Press, 1978.

DENN78b Denning, D. E., "Are Statistical Data Bases Secure?" Proc. AFIPS 1978 NCC, 525-530.

DENN79a Denning, D. E. and Denning, P. J., "Data Security", TR-301, Computer Sciences, Purdue University, March 1979.

DENN79b Denning, D. E., Denning, P. J., and Schwartz, M. D., "The Tracker: A Threat to Statistical Data Base Security," ACM TODS 4, 1 (March 1979), 76-96.

DENN79c Denning, D. E. and Schlörer, J., "A Fast Procedure for Finding a Tracker in a Statistical Dababase," Computer Sciences, Purdue Univ., and Institut für Medizinische Statistik und Dokumentation, Universität Giessen, W. Germany, Feb. 1979.

DENN79d Denning, D. E., "Complexity Results Relating to Statistical Confidentiality," Computer Science and Statistics: 12 Annual Symposium on the Interface, Waterloo (May 1979).

DOBK79 Dobkin, D., Jones, A. K., and Lipton, R. J., "Secure Data Bases: Protection Against User Inference," ACM TODS 4, 1 (March 1979), 97-106.

FEIG70 Feige, E. L. and Watts, H. W., "Protection of Privacy Through Micro-aggregation," in R. L. Bisco (ed.) Data Bases, Computers, and the Social Sciences, Wiley-Interscience (1970).

FELL50 Feller, W., An Introduction to Probability Theory and Its Applications I, Wiley, N, Y. 1950.

FELL74 Fellegi, I. P. and Phillips, J. L., "Statistical Confidentiality: Some Theory and Applications to Data Dissemination," Annals Econ. Soc'l Measurement 3, 2 (April 1974), 399-409.

HANS71 Hansen, M. H., "Insuring Confidentiality of Individual Records in Data Storage and Retrieval for Statistical Purposes," Proc AFIPS FJCC 39 (1971) 579-585.

HAQ77 Haq, M. I. "On Safeguarding Statistical Disclosure by Giving Approximate Answers to Queries," E. Morlet and D. Ribbens (Eds.), Int'l Computing Symp., North-Holland Pub., (1977).

HOFF70 Hoffman, L. J. and Miller, W. F., "Getting a Personal Dossier from a Statistical Data Bank," Datamation 16 5 (May 1970), 74-75.

KAM77 Kam, J. B., and Ullman, J. D., "A Model of Statistical Databases and their Security," ACM TODS 2, 1 (March 1977), 1-10.

KARP70 Karpinski, R. H., "Reply to Hoffman and Shaw," Datamation 16, 10 (Oct. 1970), 11.

KLEIN75 Kleinrock, L., Queueing Systems I, Wiley, N. Y., (1975), Appendix I.

NARG72 Nargundkar, M. S. and Saveland, W., "Random Rounding to Prevent Statistical Disclosure," Proc. Amer. Stat. Assoc., Soc. Stat. Sec. (1972), 382-385.

REED73 Reed, I. S., "Information Theory and Privacy in Data Banks," Proc. AFIPS 42 (1973), 581-587.

REIS78 Reiss, S. B., "Medians and Database Security," in Foundations of Secure Computation, ed. DeMillo et al., Academic Press (1973).

SCHL75 Schlörer, J., "Identification and Retrieval of Personal Records from a Statistical Data Bank," Methods of Information in Medicine 14, 1 (Jan. 1975), 7-13.

SCHL77 Schlörer, J., "Confidentiality and Security in Statistical Data Banks," in Guas, W., Henzler, R. (eds.) Data Documentation: Some Principles and

Applications in Science and Industry, Proc. Workshop Data Doc. 1975, Verl.Dok., Munchen 1977, 101-123.

SCHL79a Schlörer, J. Disclosure from Statistical Databases: Quantitative Aspects of Trackers," Institut für Medizinische Statistik Und Dokumentation, Universität Giessen, W. Germany (Mar. 1979), to appear in ACM TODS.

SCHL79b Schlörer , J., "Security of Statistical Databases: Multidimensional Transformation," TB-IMSD 2/78, Institut für Medizinische Statistik und Dokumentation, Universität Giessen, (Mar. 1979).

SCHL79c Schlörer, J., "Statistical Database Security: Some Recent Results," Institut für Medizinische Statistik und Dokumentation, Universität Giessen, 1979.

SCHW77a Schwartz, M. D., Denning, D. E., and Denning, P. J., "Securing Data Bases under Linear Queries," Proc. IFIP Congress, North-Holland Pub., (1977), 395-398.

SCHW77b Schwartz, M. D., "Inference from Statistical Data Bases," Dept. of Computer Sciences, Purdue Univ., Ph.D. Thesis (August 1977).

SCHW79 Schwartz, M. D., "Linear Queries in Statistical Data Bases," TR 216, Computer Sciences, Purdue Univ (Nov. 1976); to appear in ACM TODS (1979).

YU78 Yu, C. T., and Chin, F. Y., "A Study on the Protection of Statistical Data Bases," ACM SIGMOD Conf. on Management of Data, Toronto, Canada, (August 1977), 169-181.

APPENDIX A. ERRORS IN ESTIMATING FREQUENCIES

Let FREQ(C) be a query for a frequency, and let $FREQ^*(C)$ 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:

$$\overline{n_C^*} = n_C p$$

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

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

$$\overline{F_C^*} = \frac{n_C}{N}$$

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

Since $\overline{F_C^*} = FREQ(C)$, the sampled frequency is an unbiased estimator of the true frequency.

Let

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

be the squared relative error in $FREQ^*(C)$. The mean squared relative error (over all choices of the sample) is:

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

$$= \frac{1-p}{n_C p}$$

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, n_C^* 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 $\bar{x}$ and σ_x^2 be the mean and variance of $\{x_1, \ldots, x_{n_C}\}$:

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

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

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

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

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

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

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

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

$$= \frac{1}{\binom{n_C}{k}} \frac{n_C \bar{x}}{k} \binom{n_C - 1}{k - 1} = \bar{x}$$

For $k = 0$, we assume the response is 0; that is, $\overline{A_{C,j}^*(0)} = 0$.

Therefore,

$$\overline{A^{*}_{C,j}} = \sum_{k=0}^{n_C} \bar{x} \Pr[n_C^{*} = k] = \bar{x} \quad .$$

This implies the sampled average is an unbiased estimator of the true average. 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:

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

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

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

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

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

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

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

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

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

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

$$= \frac{1}{\binom{n_C}{k}}\left[\bar{x}^2\binom{n_C}{k} - \frac{2n_C\bar{x}^2}{k}\binom{n_C-1}{k-1} + \frac{1}{k^2}G(k,n_C)\right]$$

$$= \left[\frac{1}{k^2\binom{n_C}{k}}\right]G(k, n_C) - \bar{x}^2$$

$$= \frac{n_C\overline{x^2}\binom{n_C-2}{k-1}}{k^2\binom{n_C}{k}} + \frac{(n_C\bar{x})^2\binom{n_C-2}{k-2}}{k^2\binom{n_C}{k}} - \bar{x}^2$$

$$= \frac{n_C-k}{k(n_C-1)}\overline{x^2} - \frac{n_C-k}{k(n_C-1)}\bar{x}^2 = \frac{n_C-k}{k(n_C-1)}\sigma_x^2 \qquad (6)$$

For k = 1,

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

$$= \frac{1}{n_C} \sum_{i=1}^{n_C} (x_i - \bar{x})^2 = \sigma_x^2$$

which is the same as would be obtained by substituting k = 1 in eq. (6). For k = 0, we assume as before the response is 0; therefore,

$$\mathrm{Var}(A^*_{C,j}(0)) = \bar{x}^2$$

Therefore,

$$\mathrm{Var}(A^*_{C,j}) = \mathrm{Var}(A^*_{C,j}(0)) \Pr[n^*_C = 0] + \sum_{k=1}^{n_C} \mathrm{Var}(A^*_{C,j}(k)) \Pr[n^*_C = k]$$

$$= \bar{x}^2 (1-p)^{n_C} + \sigma_x^2 \sum_{k=1}^{n_C} \frac{n_C - k}{k(n_C - 1)} \binom{n_C}{k} p^k (1-p)^{n_C - k} .$$

For large query sets, the distribution of n^*_C is approximately normal with mean $\overline{n^*_C} = n_C p$ and variance $\mathrm{Var}(n^*_C) = p(1-p)n_C$. The value of $\mathrm{Var}(A^*_{C,j}(k))$ at $k = n_C p$ approximates $\mathrm{Var}(A^*_{C,j})$ in the range $k = n_C p \pm \sqrt{\mathrm{Var}(n^*_C)}$. Now,

$$\mathrm{Var}(A^*_{C,j}) = \overline{\mathrm{Var}(A^*_{C,j}(k))},$$

and, since the graph of $\mathrm{Var}(A^*_{C,j}(k))$ is concave up,

$$\mathrm{Var}(A^*_{C,j}) \geq \mathrm{Var}(A^*_{C,j}(\bar{k})) = \mathrm{Var}(A^*_{C,j}(n_C p)) .$$

Thus,

$$\mathrm{Var}(A^*_{C,j}) \simeq \sigma_x^2 \frac{1-p}{p(n_C - 1)}$$

Let

$$a^2_{C,j} = \left(\frac{AVG^*(C,j) - AVG(C,j)}{AVG(C,j)}\right)^2 = \left(\frac{AVG^*(C,j) - \bar{x}}{\bar{x}}\right)^2$$

be the squared relative error in $AVG^*(C,j)$. The mean squared relative error (over al choices of the sample) is:

$$\overline{a^2_{C,j}} = \left(\frac{1}{\bar{x}^2}\right) Var(A^*_{C,j}) \simeq \left(\frac{\sigma_x^2}{\bar{x}^2}\right)\left(\frac{1-p}{p(n_C-1)}\right)$$

Thus, the root mean squared relative error is approximated by:

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

APPENDIX C. COMPROMISE WITH SMALL QUERY SETS

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

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

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

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

$$\Pr[F = 1/N \mid F^* = 1/N] = \frac{\Pr[F = 1/N \text{ and } F^* = 1/N]}{\Pr[F^* = 1/N]}$$

$$= \frac{pa_1}{\sum_{k=0}^{N} kp(1-p)^{k-1} a_k} = \frac{a_1}{A'(1-p)}$$

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 mean $\lambda/(1-\lambda)$ for $0 < \lambda < 1$. 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,

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

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

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

giving

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

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,

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