Information Loss in Partitioned Statistical Databases

Jan Schlörer
Klinische Dokumentation, Universität Ulm, Eythstr. 2, D-7900 Ulm, FRG

Partitioning is an interesting approach to protecting statistical databases. It is highly secure and can, among other things, take into account the dynamics of a statistical database. This paper shows that partitioning often will either result in high information loss or in a serious distortion of important statistical functions. In order to make partitioning practical, these difficulties must be overcome.

1. INTRODUCTION

Statistical databases aim to provide information about groups of persons or organizations, while protecting the confidentiality of the individual respondents represented in the database. This objective is difficult to achieve, since frequencies and other common statistics may disclose information on identifiable individuals.[1–8]

Statistics may be distributed as micro- or macrostatistics. *Microstatistics* or *microdata*, such as the public use files of the U.S. Census Bureau,[9] consist of anonymous individual records. Frequencies, sums, and other *macrostatistics* may stem from dedicated statistical databases containing, for instance, census or other survey data,[9–13] or from multipurpose databases. We concentrate on multipurpose databases.

We assume that the integrity of the data is important in other uses of the database, so that the raw data cannot be modified or transformed.[14–16] We are therefore compelled to control the statistical output. Output controls fall into two broad categories: *output perturbations* add noise to the released statistics;[8,12,17–22] *output restrictions* impose limits on the set of allowable queries.[11,13,23–26] The practicality of a control depends on its security, cost, and information loss.

Partitioning of statistical databases[21,27–30] is an output restriction technique with several attractive features:

1. It is highly secure.
2. The dynamics of a database may aggravate the risk of disclosure. To some extent, partitioning can cope with this problem.
3. Partitioning may be extended to protect multiple overlapping statistical views of the same database.[28]

However, there remain difficulties. The cost may be considerable for large databases. Chin and his coworkers already suspected that partitioning might lead to perceptible information loss. The evidence presented in this paper suggests that information loss is indeed the biggest drawback of partitioning, as formulated below. It is also shown that dummy records, which have been proposed to reduce information loss and update waiting times,[21,27,28] lead into, as yet unsolved, problems. These difficulties must be overcome if partitioning is to become practical.

Sections 2 and 3 describe our model of a statistical database and the basic rules of partitioning. Sections 4 and 5 examine information loss in partitioned statistical databases, and section 6 investigates some consequences of dummy records.

2. STATISTICAL DATABASE MODEL

For statistical purposes, a database can be viewed as a collection of *N logical records*. Each record describes one respondent, e.g. a person or an organization. Record i contains values $x_{i1}, \ldots, x_{iM}$ for M attributes $A_1, \ldots, A_M$. Each *attribute* A_j has $|A_j|$ possible values in its domain. Some attributes have non-numeric values; an example is SEX, whose two possible values are MALE and FEMALE. Others have numeric values; an example is a student's GRADEPOINT.

Our model describes neither the database schema nor its implementation, but rather a conceptual view of the data which is typical for many statistical applications. Recently, Chin and Ozsoyoglu[28] employed a modification of the data abstraction model[31] for modelling a set of partially overlapping views of our form. The points we are going to make apply to their model as well.

Statistics are calculated for subsets of records having common attribute values. A user can specify such a subset by a *characteristic formula C*, which, informally, is any logical formula over the values of the attributes using the operators OR (+), AND (&), and NOT. An example is

$$C = (\text{SEX} = \text{FEMALE})\ \& \\ [(\text{MAJOR} = \text{BOT}) + (\text{MAJOR} = \text{ZOOL})] \quad (1)$$

which, for a student database, specifies all female students majoring in either botany or zoology. The set of records whose values match a formula C is called the *query set* of C. We write C to denote both a formula and its query set; when treating C as a set, we exchange the logical operators (+, &) for set operators ($\cup$, $\cap$). $|C|$ denotes the number of records in C, that is, the *size* or *frequency* of the query set. A query set can have size $|C| = 0$. 'ALL' denotes a formula whose query set is the entire database; thus $|\text{ALL}| = N$, and each query set is a subset of ALL.

An *m-set* is a query set which can be specified using the values of m, but no fewer attributes. The query set in (1)

CCC–0010–4620/83/0026–0218 $03.00

is a 2-set. An *elementary m-set* is an m-set specified by a formula of the form

$$E_m = (A_1 = a_{i_1}) \,\&\, \ldots \,\&\, (A_m = a_{i_m}) \qquad (2)$$

where each a_{i_j} is some value in the domain of attribute A_j $(1 \leq j \leq m)$. An elementary m-set cannot be decomposed without introducing additional attributes. Each m-set can be expressed as a union of elementary m-sets. For example, the 2-set C in (1) is the union of the elementary 2-sets (SEX = FEMALE) & (MAJOR = BOT) and (SEX = FEMALE) & (MAJOR = ZOOL).

Given $A_1, \ldots, A_m$, the total number of elementary m-sets is

$$s_m = \prod_{j=1}^{m} |A_j| \qquad (3)$$

These s_m sets define an m-dimensional table or *m-table*, where each attribute corresponds to one dimension of the table. A database with M attributes has 2^M such tables, corresponding to the 2^M possible subsets of the attributes. A table need not correspond to a physical structure of the database. There is one 0-table, consisting of the elementary 0-set ALL, and one M-table, where the records in each elementary M-set are indistinguishable, except possibly for a statistically irrelevant identifier field. Figure 1 provides an example; for more details see Refs 24 and 32.

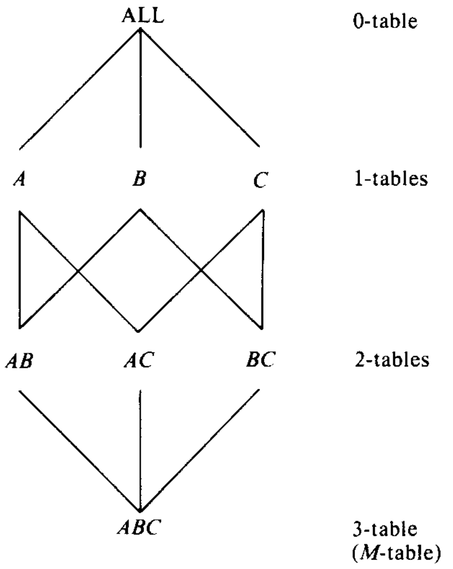

Figure 1. The 2^M tables over $M = 3$ attributes A, B, and C.

Statistics are calculated from the records in a query set C and have the general form $f(C, D)$, where D is a possibly empty set of attributes and f is a statistical function. Some examples are:

$$f(C, D) = |C| \qquad (4)$$

$$f(C, D) = \mathrm{SUM}(C, A_j) = \sum_{i \in C} x_{ij} \qquad (5)$$

$$f(C, D) = \mathrm{MEAN}(C, A_j) = \mathrm{SUM}(C, A_j)/|C| \qquad (6)$$

For a frequency or count (4) the set D is empty. In sums (5) and arithmetic means (6), D contains a single attribute A_j with numeric values. For the query set C of (1) and A_j = GRADEPOINT, SUM(C, A_j) is the sum of gradepoints for all female students majoring in either botany or zoology, and MEAN(C, A_j) is the mean of their gradepoints. An attribute is called a *characteristic attribute* if it appears in C, and a *data attribute* or *property*,[28] if it appears in D; an attribute can appear in both.

It is customary to speak of *tables of statistics*, e.g. of frequency tables. These 'statistical' tables correspond to our m-tables. In the simplest case, each so-called *cell* in a statistical table corresponds to an elementary m-set and exhibits some statistic for that set, e.g. its frequency. In other cases, the cells of a statistical table may correspond to m-sets which are unions of elementary m-sets.

3. PARTITIONING

We employ four basic rules of partitioning.[21,27–30] They ensure that no statistic corresponding to a single individual is ever released. An example appears in Section 5.

R1. Using the values of μ $(1 \leq \mu \leq M)$ *partitioning attributes* the database is partitioned into p mutually exclusive, non-overlapping record sets, the so-called atomic populations or *A-populations* AP$_1, \ldots,$ AP$_p$.

R2. A-populations of *size 1 are prohibited*. According to McLeish[29] we permit A-populations of arbitrary size $|\mathrm{AP}| \geq 2$. In addition, we permit A-populations of size 0, even though they can lead to negative disclosure,[12,25] that is they may disclose that someone lacks a certain property. Section 5 shows that it is usually impractical to avoid empty A-populations. Recalling that N is the number of records in the database, we observe that there are at most $N/2$ non-empty A-populations.

R3. A statistic $f(C, D)$ can only be permitted if its query set C is a *union of A-populations*. In particular, $f(C, D)$ is always restricted, if there exists some A-population AP such that

$$|\mathrm{AP} \cap C| \neq 0$$

and

$$|\mathrm{AP} - C| \neq 0$$

Sometimes a statistic may be restricted, even if C is a union of A-populations,[28] but this need not concern us here.

R4. *Only the μ partitioning attributes of R1 may serve as characteristic attributes.* The remaining attributes may function only as data attributes $\in D$ in computing $f(C, D)$.

4. RESTRICTED FUNCTIONS OF ATTRIBUTES

Suppose the administrator of a student database decides that GRADEPOINT is a data attribute, whereas MAJOR is a partitioning attribute. Then, according to R4, the frequency table in Fig. 2 is not permitted. Indeed, this table would not only violate R4. Our experience with real databases[33] shows that it would almost certainly violate R3 as well: the 'illegal' characteristic attribute GRADEPOINT usually will dissect A-populations. Next suppose we use GRADEPOINT, grouped as in Fig. 2, as

MAJOR GRADEPOINT	BOT	CS	EE	...	ZOOL
0.0–1.9	5	28	9	...	7
2.0–2.4	8	47	24	...	12
2.5–2.9	29	91	85	...	53
3.0–3.4	13	39	30	...	22
3.5–4.0	15	16	18	...	10

Figure 2. A 2-dimensional frequency table for MAJOR × GRADEPOINT.

a partitioning attribute. Then partitioning, as presently formulated, seems to prohibit queries like

$$f(C, D) = \text{MEAN}[(\text{MAJOR} = \text{BOT}), \text{GRADEPOINT}]$$

This latter problem is a minor one. It appears that most such queries would not pose a serious security problem, provided the query set is a union of A-populations and contains sufficiently many records. Another difficulty is more serious. In practice it is often desirable to employ a quantitative attribute (with numeric values) as a data attribute in some queries, and, usually after some grouping, as a characteristic attribute in other queries. The next section shows that number and domain sizes of partitioning attributes are severely limited. This, together with R4, removes much of the desirable flexibility. In particular, there remains little freedom to employ grouped quantitative attributes like GRADEPOINT as characteristic attributes.

5. PARTITIONING WITHOUT DUMMY RECORDS

We begin with an example. The leftmost column in Fig. 3 depicts the 3-table ABC for a small database. The attributes A, B, and C have domains $\{0, 1, 2\}$, $\{0, 1\}$, and $\{0, 1\}$, respectively. The notation follows Chin:[34] 010 denotes $(A = 0)$ & $(B = 1)$ & $(C = 0)$, 20* means $(A = 2)$ & $(B = 0)$, and so on. There are $N = 15$ records, $s_3 = 12$ elementary 3-sets, and a quotient $s_3/N = 0.8$. Six out of 15, that is, 40% of the records belong to elementary 3-sets of size 1. For the sake of illustration, this percentage is exaggerated. Typically only 5–15% of the records in m-tables with $s_m/N \approx 0.8$ belong to elementary m-sets of size 1.[33]

In addition, Fig. 3 shows four partitions of the database, set up according to rules R1 and R2. Partition 1 is sequential: each elementary 3-set of size 1 is merged with the next such set which happens to occur. Partition 2 tries to save the 2-table AB. This turns out to be impossible, since $|21^*| = 1$. One can only form A-populations $\text{AP}_5 = 110$, $\text{AP}_6 = 111 \cup 210$, and $\text{AP}_8 = 211$. Partition 3 aims to save the 2-table AC and fails for a similar reason. Only Partition 4 succeeds in saving BC. Figure 4 displays the frequency tables resulting from the four partitions. Frequencies whose query sets violate R3 are indicated by '—'.

The example database contains a total of $\prod_{j=1}^{3} (|A_j| + 1) = 36$ (Ref. 13) elementary m-sets, $0 \leq m \leq 3$. Each partition permits the 0-set ALL, but suppresses roughly two thirds of the remaining 35 elementary sets. It is particularly distressing that about 50% of the elementary 1-sets are restricted, even though 1-dimensional frequency tables are as safe as any statistical output can be. For each single attribute, one may arbitrarily permute the 15 entries in the 15 records, but will always get the same 1-dimensional frequency tables. Like almost all databases this one is 1-transformable.[14, 16] The example database is even 2-transformable: publishing the frequencies for the 2-tables AB, AC, and BC will not disclose the frequencies for ABC. The interested reader may enjoy determining the four databases which are consistent with this particular set of 2-dimensional frequency tables.

Unfortunately, any partition will restrict at least some elementary 1-sets, as soon as there are elementary μ-sets of size 1:

Lemma

Assume the $\mu \geq 1$ partitioning attributes define a μ-table with at least one elementary μ-set E of size 1. Then rule R3 restricts at least 2 elementary 1-sets.

Proof. By (2) E has characteristic formula

$$E = E_\mu = (A_1 = a_{i_1}) \ \& \ldots \& \ (A_\mu = a_{i_\mu})$$

Characteristic formula	Size of elementary 3-set	Partition 1 (sequential)									Partition 2 (for AB)								Partition 3 (for AC)								Partition 4 (for BC)						
ABC		1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7
000	1	|									|								|								|						
001	1	|									|									|								|					
010	3		|									|						|										|				
011	2			|									|						|										|			
100	2				|									|						|						|						
101	1					|								|							|						|					
110	0						|								|							|								|		
111	1					|										|					|								|			
200	2							|									|						|								|	
201	1								|								|							|			|					
210	1								|							|								|				|				
211	0									|								|							|							|

Figure 3. A database with attributes A, B, and C is partitioned in four ways. '|' indicates membership of an elementary 3-set in the A-population whose number appears as column header. For further details see text.

Table	Characteristic formula ABC	Frequency	Frequencies returned to the user			
			Partition 1	Partition 2	Partition 3	Partition 4
ALL	***	15	15	15	15	15
A	0**	7	7	7	7	—
	1**	4	4	—	4	—
	2**	4	4	—	4	—
B	*0*	8	—	8	—	8
	1	7	—	7	—	7
C	**0	9	—	—	—	9
	**1	6	—	—	—	6
AB	00*	2	2	2	—	—
	01*	5	5	5	—	—
	10*	3	—	3	—	—
	11*	1	—	—	—	—
	20*	3	—	3	—	—
	21*	1	—	—	—	—
AC	0*0	4	—	—	4	—
	0*1	3	—	—	3	—
	1*0	2	2	—	2	—
	1*1	2	2	—	2	—
	2*0	3	—	—	—	—
	2*1	1	—	—	—	—
BC	*00	5	—	—	—	5
	*01	3	—	—	—	3
	*10	4	—	—	—	4
	*11	3	—	—	—	3
ABC	000	1	— (1)	— (1)	— (1)	— (1)
	001	1	— (1)	— (1)	— (2)	— (2)
	010	3	3 (2)	3 (2)	— (1)	— (3)
	011	2	2 (3)	2 (3)	— (2)	— (4)
	100	2	2 (4)	— (4)	2 (3)	— (1)
	101	1	— (5)	— (4)	— (4)	— (2)
	110	0	0 (6)	0 (5)	0 (5)	0 (5)
	111	1	— (5)	— (6)	— (4)	— (4)
	200	2	2 (7)	— (7)	2 (6)	2 (6)
	201	1	— (8)	— (7)	— (7)	— (2)
	210	1	— (8)	— (6)	— (7)	— (3)
	211	0	0 (9)	0 (8)	0 (8)	0 (7)

Figure 4. Frequency tables for partitions 1–4 of Fig. 3. Rule R3 of partitioning suppresses the frequencies denoted by '—'. For each elementary 3-set the corresponding A-population is displayed in brackets.

R2 prohibits A-populations of size 1. Thus, E must be merged with at least one other elementary μ-set E^*: AP = $E \cup E^*$. The characteristic formulae of E and E^* differ for at least one attribute. Assume that E contains ($A_k = \kappa$), whereas E^* contains ($A_k = \kappa^*$). Consequently, the elementary 1-sets

$$E_1 = (A_k = \kappa)$$
$$E^*{}_1 = (A_k = \kappa^*)$$

are not unions of A-populations, and R3 restricts them. ▮

Our empirical data[33] show clearly that s_μ/N must be very small, if elementary μ-sets of size 1 are to be entirely avoided. We counted thousands of tables from 27 real statistical databases; Fig. 5 lists some of the results. Figure 6 shows that it is even more difficult to avoid A-populations of size 0. Partitioning is facilitated, if the frequencies in the μ-table approach equidistribution, that is if the individual attributes are approximately equidistributed with weak mutual interdependencies. However, most of our distributions exhibit some, and often considerable, skewness. Moreover, even for a perfectly equidistributed μ-table the expected number of elementary μ-sets of size 1 is Ne^{-N/s_μ} (Ref. 10). This figure is close to zero for $s_\mu/N = 0.1$, but it exceeds $0.01N$ for $s_\mu/N \gtrsim 0.22$.

s_μ/N	t	P_{10}	P_{50} (median)	P_{90}
0.005–<0.006	128	0	0.02	0.06
0.009–<0.010	88	0	0.05	0.09
0.020–<0.025	284	0	0.19	0.30
0.045–<0.050	232	0.31	0.49	0.64
0.095–<0.100	86	0.82	1.12	1.46
0.200–<0.250	462	1.90	2.69	3.70
0.450–<0.500	250	3.70	4.99	7.60

Figure 5. Elementary μ-sets of size 1 in μ-tables from 27 actual statistical databases. t denotes the number of observed μ-tables. Entry x in column P_q (q = 10, 50, 90) means that, in q % of the t observed tables, x or fewer % of the N records belong to elementary μ-sets of size 1.

s_μ/N	t	P_{10}	P_{50} (median)	P_{90}
0.005–<0.006	128	0	1.8	24.7
0.009–<0.010	88	0	7.8	25.5
0.020–<0.025	284	0	17.7	43.8
0.045–<0.050	232	10.2	32.7	51.8
0.095–<0.100	86	16.9	39.6	61.8
0.200–<0.250	462	38.1	58.9	71.1
0.450–<0.500	250	55.5	69.5	77.0

Figure 6. Empty elementary μ-sets of size 0 in μ-tables from 27 actual statistical databases. t and P_q are as in Fig. 5. However, entries in column P_q are in % of s_μ (not in % of N as in Fig. 5). The absolute number of empty elementary sets exceeds that of elementary sets with 1 record in most tables with $s_\mu/N \gtrsim 0.01$.

6. DUMMY RECORDS

Chin and Ozsoyoglu have proposed *dummy records* in order to reduce information loss and escape update waiting times.[21,27,28] Dummy records serve to pad A-populations of size 1 and enable processing of records in pairs. In this way compromise resulting from updates in the database can be avoided. In a dummy record, the value of a data attribute is a value of a random variable Z with expectation $E(Z) = 0$, and with a standard deviation depending on protection requirements. Frequencies (4), sums (5), and means (6) will do to explain the resulting problems.

There are many data attributes with non-negative values, such as GRADEPOINT, SALARY, or WEIGHT. The expectation $E(Z) = 0$ implies that some dummy records must contain negative, i.e. 'impossible' values. This is a minor problem, provided most of the true entries are sufficiently large to guarantee positive sums and means.

Zero bias is an important requirement for output perturbations.[8,12,13,17,20–22] That is, the difference $f - E(F)$ between a true answer f and the expectation of its perturbed estimate F should be zero or, at least, as small as possible.

Now suppose the dummy records enter all statistical functions. Sums remain unbiased, since

$$E(\sum Z_i) = \sum E(Z_i) = 0$$

But, owing to the dummy records, frequencies become too high, and means are too low. Both frequencies and means are biased.

Users generally will insist that 1-dimensional frequency tables must be available. Then, by our lemma, the number of dummy records in the database must at least equal the number of elementary μ-sets of size 1. The effect is considerable bias (see Fig. 5). For $s_\mu/N \approx 0.5$ frequencies will be about 5% too high. Even for $s_\mu/N \approx 0.1$ the error will be close to 1%. Arithmetic means are underestimated by roughly the same percentage.

Next assume the dummy records enter only sums, but no frequencies, and that

$$\text{mean} = \text{sum including dummy records/true frequency}$$

The three statistics remain unbiased and mutually consistent. But this solution violates R2, a central postulate of partitioning: some frequencies and means are computed using A-populations of size 1.

7. CONCLUSION

The frequency distributions in many real statistical databases limit number and domain sizes of partitioning attributes. Elementary μ-sets of size 1 already appear at a modest s_μ/N. Dummy records are required if severe information loss is to be avoided. Dummy records in their turn introduce bias into frequencies and arithmetic means, unless one resolves to violate a central postulate of partitioning. Confining partitioned databases to sum queries, which stay unbiased by dummy records, is not a viable solution. A practical system must offer frequencies, means, and other statistics.

The *audit expert*, an elegant audit-based output restriction technique for sum queries,[23] may also be suspected to induce noticeable information loss for increasing s_μ/N. The audit expert is clearly more liberal than partitioning. It regards each query set as a vector and admits up to $N-1$ overlapping 'basic' query sets (basis vectors) of size ≥ 2, whereas partitioning permits at most $N/2$ non-overlapping A-populations of size ≥ 2; but the audit expert lacks syntactic control. It cannot tell innocent overlaps of query sets with unrelated formulae from dangerous overlaps that might lead to disclosure. Consequently, it will suppress too many answers.

It appears premature to dismiss these protection techniques out of hand. The introduction named several attractive features of partitioning. There is another such property. Output restriction may work at the table- or at the cell-level.[24] *Table-level* controls restrict or permit entire tables of statistics. *Cell-level* controls aim to restrict sensitive cells of a table (e.g. those corresponding to a single individual), while permitting certain non-sensitive ones. Cell-level controls tend to be more precise, thereby enhancing security, but they tend also to be more costly. Partitioning combines table-level and cell-level characteristics, the audit expert is a cell-level control. Both techniques belong to the, at present very small, group of cell-level controls that are both precise and, at least in principle, tractable for multipurpose databases. They deserve further study.

The audit expert requires testing on actual databases. For partitioning, one might investigate the effect of assigning a count of, say, -1 to 50% of the dummy records. A particularly interesting question seems whether some of the highly demanding postulates of partitioning can be partially relaxed without undue loss of protection effect; this might allow one to use more partitioning attributes.

Acknowledgements

The empirical data are largely due to the work of L. Zick. D. Denning and F. Chin, and one referee helped a lot by commenting on earlier versions. R. and I. Rauther gave valuable stylistic advice. Research supported by Stiftung Volkswagenwerk.

REFERENCES

1. T. Dalenius, Towards a methodology for statistical disclosure control. *Statistisk tidskrift* **15** (5), 429–444 (1977).
2. W. De Jonge, Compromising statistical databases responding to queries about means. January (revised November 1981), Vakgroep Informatica, Vrije Universiteit, Amsterdam (to appear in *ACM Transactions on Database Systems*).
3. D. E. Denning, *Cryptography and Data Security*. Addison-Wesley, Reading, Mass., U.S.A. (1982).
4. D. E. Denning, P. J. Denning and M. D. Schwartz, The tracker: a threat to statistical database security. *ACM Transactions on Database Systems* **4** (1), 76–96 (1979).
5. D. E. Denning and J. Schlörer, A fast procedure for finding a tracker in a statistical database. *ACM Transactions on Database Systems* **5** (1), 88–102 (1980).
6. D. Dobkin, A. K. Jones and R. J. Lipton, Secure databases: protection against user influence. *ACM Transactions on Database Systems* **4** (1), 97–106 (1979).
7. S. P. Reiss, Medians and database security. In: R. A. DeMillo, D. P. Dobkin, A. K. Jones and R. J. Lipton (eds), *Foundations of Secure Computation*, Academic Press, New York, 57–91 (1978).
8. J. Schlörer, *Query Based Output Perturbations to Protect Statistical Databases*. October, Klinische Dokumentation, Univ. Ulm, Ulm, W.Germany (1982).
9. Office of Federal Statistical Policy and Standards, *Statistical Policy Working Paper 2: Report on Statistical Disclosure and Disclosure Avoidance Techniques*. U.S. Dept. of Commerce, U.S. Government Printing Office, Washington, D.C. (1978).
10. H. Block and L. Olsson, Bakvägsidentifiering. *Statistisk tidskrift* **14** (2), 135–144 (1976).
11. L. H. Cox, Suppression methodology and statistical disclosure control. *Journal of the American Statistical Association* **75** (370), 377–385 (1980).
12. I. P. Fellegi and J. L. Phillips, Statistical confidentiality: some theory and applications to data dissemination. *Annals of Economic and Social Measurement* **3** (2), 399–409 (1974).
13. L. Olsson, Protection of output and stored data in statistical data bases. *ADB-Information No. 4*, Statistiska Centralbyrån, Stockholm, Sweden (1975).
14. T. Dalenius and S. P. Reiss, Data swapping: a technique for disclosure control. *Proceedings of the Section on Survey Research Methods*, American Statistical Association, Washington, D.C., 191–196 (1978).
15. S. P. Reiss, M. Post and T. Dalenius, *Nonreversible Privacy Transformations*. November, Department of Computer Science, Brown Univ., Providence, RI, U.S.A. (1981).

16. J. Schlörer, Security of statistical databases: multidimensional transformation. *ACM Transactions on Database Systems* **6** (1), 95–112 (1981).

17. L. L. Beck, A security mechanism for statistical databases. *ACM Transactions on Database Systems* **5** (3), 316–338 (1980).

18. L. H. Cox and L. R. Ernst, *Controlled Rounding*. June (revised January 1981), U.S. Bureau of the Census, Washington, D.C. (1980).

19. T. Dalenius, A simple procedure for controlled rounding. *Statistisk tidskrift* **19** (3), 202–208 (1981).

20. D. E. Denning, Secure statistical databases with random sample queries. *ACM Transactions on Database Systems* **5** (3), 291–315 (1980).

21. G. Ozsoyoglu and M. Ozsoyoglu, Update handling techniques in statistical databases. *Proceedings of the First LBL Workshop on Statistical Database Management*, Lawrence Berkeley Laboratory, Univ. California, Berkeley, CA, U.S.A. (1982).

22. J. Schlörer and D. E. Denning, Protecting query based statistical output in multipurpose database systems. In: V. Fåk (ed.), *Proceedings of IFIP's First Security Conference*, North-Holland, Amsterdam, Netherlands (to appear).

23. F. Y. Chin and G. Ozsoyoglu, Auditing and inference control in statistical databases. *IEEE Transactions on Software Engineering* **SE-8** (6), 574–582 (1982).

24. D. E. Denning, J. Schlörer and E. Wehrle, *Memoryless Inference Controls for Statistical Databases*. August, Computer Sciences Department, Purdue Univ., W. Lafayette, IN, U.S.A. (1982).

25. M. I. Hag, Insuring individual's privacy from statistical data base users. *Proceedings AFIPS National Computer Conference* **44**, AFIPS Press, Arlington, VA, U.S.A., 941–946 (1975).

26. J. Schlörer, Confidentiality of statistical records: a threat-monitoring scheme for on line dialogue. *Methods of Information in Medicine* **15** (1), 36–42 (1976).

27. F. Y. Chin and G. Ozsoyoglu, Security in partitioned dynamical statistical databases. *Proceedings of the IEEE COMPSAC Conference*, 594–601 (1979).

28. F. Y. Chin and G. Ozsoyoglu, Statistical database design. *ACM Transactions on Database Systems* **6** (1), 113–139 (1981).

29. M. McLeish, *Further Results on the Security of Partitioned Dynamic Statistical Databases*. July (revised June 1982), Department of Computer Science, Univ. Alberta, Edmonton, Alberta, Canada (1981).

30. C. T. Yu and F. Y. Chin, A study on the protection of statistical data bases. *Proceedings of the ACM SIGMOD International Symposium on Management of Data*, Association for Computing Machinery, New York, 169–181 (1977).

31. J. M. Smith and D. C. P. Smith, Database abstractions: aggregation and generalization. *ACM Transactions on Database Systems* **2** (2), 105–133 (1977).

32. D. E. Denning, *A Security Model for the Statistical Database Problem*. January, Computer Sciences Department, Purdue Univ., W. Lafayette, IN, U.S.A. (1983).

33. J. Schlörer and L. Zick, *Empirical Investigations on the Identification Risk in Statistical Databases*. June, Klinische Dokumentation, Universität Ulm, Ulm, W. Germany (1982).

34. F. Y. Chin, Security in statistical databases for queries with small counts. *ACM Transactions on Database Systems* **3** (1), 92–104 (1978).

Received August 1982