1989

Relative compromise of statistical databases

M Miller

Jennifer Seberry

University of Wollongong, jennie@uow.edu.au

Publication Details

Relative compromise of statistical databases

Abstract
Statistical databases are databases in which only statistical type of queries are allowed. The results of the statistical queries are intended for statistical use only. However, it has been shown that using only statistical queries it is often possible to infer an individual's value of a protected field (e.g., using various types of trackers). In such a case we say that the database has been (positively) compromised. Various types of compromise have been studied but until now attention has centred on the inference of exact information from permitted queries. In this paper we introduce a new type of compromise, the 'relative' compromise: a set of records is relatively compromised with respect to a field X if the relative order of magnitude of the X-values of the set is known. This paper shows that even when exact information is protected, relative information may be accessible. We consider several sets of conditions under which this compromise can occur using SUM type of queries of fixed query set size, as well as some of the possible consequences of relative compromise.

Disciplines
Physical Sciences and Mathematics

Publication Details
Relative compromise of statistical databases

M. Miller
Department of Mathematics, Statistics and Computing Science, University of New England, Armidale, NSW

J. Seberry
Department of Computer Science, University College, ADFA, Canberra, ACT

Statistical databases are databases in which only statistical type of queries are allowed. The results of the statistical queries are intended for statistical use only. However, it has been shown that using only statistical queries it is often possible to infer an individual’s value of a protected field (e.g., using various types of trackers). In such a case we say that the database has been (positively) compromised. Various types of compromise have been studied but until now attention has centred on the inference of exact information from permitted queries. In this paper we introduce a new type of compromise, the ‘relative’ compromise: a set of records is relatively compromised with respect to a field if the relative order of magnitude of the X-values of the set is known. This paper shows that when exact information is protected, relative information may be accessible. We consider several sets of conditions under which this compromise can occur using SUM type of queries of fixed query set size, as well as some of the possible consequences of relative compromise.

Keywords and Phrases: Database inference, statistical database, SUM queries, compromise of statistical databases, relative compromise.


INTRODUCTION

A database D is a finite set of N records (or tuples) in which each record has a finite number of fields (or attributes). For the purpose of this study we shall assume that one of these fields or attributes, say X, is to be kept secret for all records i in the database. Furthermore, we assume that the elements x, ɛ X are real numbers.

A statistical database is a database in which only statistical types of queries are allowed, such as COUNT, SUM, AVG, MIN, MAX. Such a query q(S; X) operates on the values of the attribute X of a subset S (called the query set) of the database. The query set is selected by a characteristic formula. For a more detailed explanation of the terms used here refer to Denning (1982), Chapter 6. We shall not exclude the possibility of using key values in the characteristic formula. For simplicity we identify both a characteristic formula and its corresponding query set by the same symbol; and we shall write q(S) instead of q(S; X) when X is understood.

A statistical database is to be used for statistical purposes only and the X-values of the individual records are to be protected from disclosure. If a disclosure of a X-value of any of the individual records occurs we say that the database has been compromised. Various types of compromise have been defined (Davida, et al., 1978; Denning, 1982; Dobkin, et al., 1979), for example:

Definition 1: A database is said to be positively compromised, or simply compromised, if one or more individuals can have their X-values associated with them.

Definition 2: A database is negatively compromised if it is known that some particular value is not the X-value of a particular individual.

Definition 3: If all individual records in a subset S of a database D can be compromised we say that the subset S is completely compromised.

In the absence of any restrictions on the queries a statistical database can be compromised simply by making the query SUM (S) where S is a characteristic formula that uniquely identifies some individual i. Alternatively, we could deduce the value of x, from SUM (D) — SUM (D — S). It is therefore obvious that to protect the field X we need to place some restrictions on the allowed queries. To prevent the above compromise we restrict the query set size k = |S| to be within the range (2, N — 1), and taking into account possible supplementary knowledge of the value n — 1 individuals, we further restrict k to

n ≤ k ≤ N — n

(\*)

Since this restriction is necessary (although not sufficient) to prevent compromise we shall assume it from now on.

Relative Compromise

It was shown (Denning, 1982) that even for k restricted to being close to the value of (\*) it is possible to compromise a database using the techniques of individual, general, dou-
relative and union trackers. These techniques rest on:
(a) the ability to ask queries whose query sets overlap
and/or on
(b) the ability to ask queries of variable query set size
(within our general constraint (6)) and/or on
(c) the ability to ask the number of queries needed to
make the inferences.

We say that the overlap of a set of queries is \(\lambda\) if any
two queries of the set have at most \(\lambda\) individuals in common
and some two queries have exactly \(\lambda\) individuals in common.
If we restrict the overlap \(\lambda\) to \(\lambda = 0\) then
compromise using trackers is not possible. However, such a
database would not be very useful.

(b) On the other hand, if query set size \(k\) were constant
then compromise is still possible (by exploiting the overlap
for MIN or MAX queries with cleverly chosen query sets;
or by solving a system of linear equations for SUM or AVG
queries) even with the smallest possible overlap \(\lambda = 1\) as
long as there is no restriction on the number of queries
allowed concerning the individuals involved in the
queries1.

(c) Thus another possible approach is to restrict the
number of queries allowed. Using this method it was found
that (positive) compromise can be prevented. In particular,
Davida et al. (1978) found that it is not possible to
compromise a database by asking \(M\) queries, each of size \(k\),
concerning \(N\) individuals, if no two queries overlap in
more than one position and if

\[
k \leq N - \sqrt{\frac{N}{M}} + 4(N - \frac{N}{M})
\]

On the other hand, Dobkin, Jones and Lipton (1979)
studied the function \(M = S(N,k,\lambda)\), where \(M\) is the smallest
number of SUM queries that suffices to compromise the
database, \(k\) is the query set size (fixed), \(\lambda\) is the query set
overlap, and \(l\) is the number of \(X\)-values known a priori
to the user. They found that:
(a) \(S(N,k,1,0) \leq 2k - 1\), \(N \geq k^2 + k - 1\)
(b) \(S(N,k,1,1) \leq 2k - 2\), \(N \geq k^2 - 1\)
(c) \(S(N,k,\lambda + \alpha,2\alpha - 1) \leq 2k\), \(N \geq k^2 + 2\alpha\)
(d) \(S(N,k,\lambda,\lambda - 1) \leq S(N,k,1,0)\), \(N \geq k^2\)

They also showed that compromise is impossible if

\[
N < \frac{k^2 - l(l + 1)}{2\alpha} + \frac{k + l + 1}{2}
\]

That is, in this case \(S(N,k,\lambda,0) = \infty\).

They also showed that compromise is impossible if

\[
N < \frac{k^2 - l(l + 1)}{2\lambda} + \frac{k + l + 1}{2}
\]

In this paper we consider the restriction on the number
of queries for SUM type of queries. The compromise of a
database using only SUM queries of fixed query set size \(k\)
always involves the construction of some \(l\) linearly inde-
dependent queries concerning \(l\) individuals. Such a set of
queries can be expressed as a system of \(l\) linearly inde-
dependent equations in \(l\) unknowns. Solving these equations
leads to a complete compromise of the \(l\) individuals. To
prevent this compromise we could restrict the number of
queries allowed so that in any system of equations derived
from the queries there would always be at least one more
unknown than the number of equations. This control could
be implemented either in a static or a dynamic way. We
shall now consider such a system where we allow SUM
queries as long as no subset of all queries asked could be
written as a system of \(l\) linearly independent equations on \(l\)
unknowns. We shall show that even this restriction does
not prevent a new type of compromise, the "relative
compromise".

Definition 4: A subset \(S\) of \(S > 1\) of a database is
relatively compromised if the relative order of magnitude of
the individuals in the subset is known.

Theorem 1: Let \(S\) be a subset of a database, \(|S| = k + 1\).
Then a subset of \(k\) individuals of \(S\) can be relatively compromised
using only \(k\) SUM queries with fixed query set size \(k\).

Proof: Construct queries that can be written as the
following system of \(k\) equations in \(k + 1\) unknowns.

\[
x_1 + x_2 + \cdots + x_k + x_{k+1} = q_1
\]
\[
x_1 + x_2 + \cdots + x_k + x_{k+1} = q_2
\]
\[
\vdots
\]
\[
x_1 + x_2 + \cdots + x_{k-l} + x_{k+1} = q_{k+1}
\]

that is,

\[
AY = Q
\]

where

\[
A = \begin{bmatrix}
0 & 1 & 1 & \cdots & 1 & 1 \\
1 & 0 & 1 & \cdots & 1 & 1 \\
1 & 1 & 0 & \cdots & 1 & 1 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\
1 & 1 & 1 & \cdots & 0 & 1
\end{bmatrix}
\]

and \(Y^r = (x_{k+2}, \ldots, x_{k+1})\).

Let \(X_{k+1}\) be the column vector with entries all equal to
\(x_{k+1}\) and let \(X\) be the column vector with entries \(x_1, x_2, \ldots, x_k\).
Then

\[
(J - \delta X = Q - X_{k+1})
\]

where \(J\) is the \(k \times k\) unit matrix and \(I\) is the \(k \times k\) identity
matrix.

Then

\[
X = (\frac{1}{k-1} J - \delta Q - X_{k+1})
\]

\[
Q^* = \frac{1}{k-1} X_{k+1}
\]

Thus \(x_1, x_2, \ldots, x_k\) can be expressed in terms of \(x_{k+1}\) as

\[
x_1 = q_1^* - \frac{1}{k-1} x_{k+1}
\]
Knowing $q^*_i$, we can find the relative order of magnitude of the elements $X_1, X_2, \ldots, X_k$ (it will be the same as the relative order of magnitude of the corresponding $q^*_i$ values).

**Example**

Suppose we make three queries about four individuals with the following results:

1. $X_2 + X_3 + X_4 = 1050$
2. $X_1 + X_3 + X_4 + X_5 = 70$
3. $X_1 + X_2 + X_4 + X_5 = 1020$

Then

1. $X_1 = 20 - \frac{1}{2}X_4$
2. $X_2 = 1000 - \frac{1}{2}X_4$
3. $X_3 = 50 - \frac{1}{2}X_4$

and so $X_1 < X_3 < X_2$.

Note that in some situations a relative compromise with supplementary knowledge could lead to a more serious compromise of the database. Consider for example the situation when it is known that $X_i$ are all nonnegative and that any individual with $X_i > 900$ has AIDS while any individual with $X_i < 100$ does not have AIDS.

Then we could infer from the above example that:

1. $0 \leq X_4 \leq 40$
2. $0 \leq X_1 \leq 20$
3. $900 \leq X_2 \leq 1000$
4. $30 \leq X_3 \leq 50$

and hence that the individual corresponding to $X_2$ must have AIDS while none of the other individuals involved in the queries have AIDS.

Note also that we know rather more than just the order of the $X$-values; in fact, we know the differences $X_i - X_j (= q^*_i - q^*_j)$ of any two of them. Hence knowing one of the values (if for example the user could plant his/her own X-value into the queries) would result in the complete compromise of all the individuals involved in the queries.

The next theorem shows that relative compromise is possible even if we further restrict the number of queries allowed.

**Theorem 2:** Let $S$ be a subset of a database, $|S| = k + l$. Then a subset of $k$ individuals of $S$ can be relatively compromised using only $k$ SUM queries with fixed query set size $k$.

**Proof:** Construct queries that can be written as the following system of $k$ equations on $k + l$ unknowns.

$x_2 + x_3 + \ldots + x_k + x_{k+1} + \ldots + x_{k+l} = q_1$
$x_1 + x_3 + \ldots + x_k + x_{k+1} + \ldots + x_{k+l} = q_2$
$\ldots$
$x_1 + x_2 + \ldots + x_{k-l} + x_{k+1} + \ldots + x_{k+l} = q_k$

This can be written as

$(J - I)X = Q - X_{k+l}$

where $X$ is the column vector with entries $X_1, X_2, \ldots, X_k$ and $X_{k+l}$ is the column vector with entries all equal to $x_{k+1} + x_{k+2} + \ldots + x_{k+l}$.

Then

$X = (J - I)^{-1}Q - X_{k+l}$

and so

1. $x_1 - q^*_1 - \frac{1}{k-1}(x_{k+1} + \ldots + x_{k+l})$
2. $x_2 - q^*_2 - \frac{1}{k-1}(x_{k+1} + \ldots + x_{k+l})$
3. $\ldots$

Since we know the values $q^*_i$ we can find the relative order of magnitude of the elements $X_1, X_2, \ldots, X_k$.

**Example**

Consider a set of six individuals and suppose we ask four queries revealing the following SUMs:

1. $X_2 + X_3 + X_4 + X_5 + X_6 = 1330$
2. $X_1 + X_3 + X_4 + X_5 + X_6 = 1080$
3. $X_1 + X_2 + X_4 + X_5 + X_6 = 1360$
4. $X_1 + X_2 + X_3 + X_5 + X_6 = 380$

Then

1. $X_1 = 160 - \frac{1}{3}(X_2 + X_6)$
2. $X_2 = 910 - \frac{1}{3}(X_5 + X_6)$
3. $X_3 = 70 - \frac{1}{3}(X_5 + X_6)$
4. $X_4 = 3010 - \frac{1}{3}(X_5 + X_6)$

and so we deduce that $X_1 < X_2 < X_3 < X_4$.

So far we have considered only the case of relative compromise of queries of fixed query set size $k$ with overlap $k - 1$. Let us now consider a more general situation when the overlap is not necessarily equal to $k - 1$.

**Theorem 3:** Let $S$ be a subset of a database, $|S| = k + l$. If $m$ SUM queries with fixed query set size $k$ and over-
lap $\lambda^*$ lead to the complete compromise of $m$ individuals
then a subset of $m$ individuals of $S$ can be relatively com-
romised using only $m$ SUM queries of query set size $k = k^* + l - m$ with overlap $\lambda - \lambda^* + l - m$.

Proof: (i) $\lambda > l - m$
Suppose $m$ individuals of $S$ can be compromised using $m$
SUM queries of query set size $k^*$ with overlap $\lambda^*$. Let these
$m$ queries be written as a system of $m$ linearly independent
equations

$$AX = Q$$

where $A$ is an $m \times m$ query matrix, $Q$ is a column vector $(q_1, q_2, \ldots, q_m)$ and $X$ is a column vector $(x_1, x_2, \ldots, x_m)$.

Then since $A$ is a nonsingular matrix with constant
rowsum $k^*$, it has an inverse whose rowsum is constant and
equal to $\frac{1}{k}$.

Now, we can construct queries corresponding to the
following equations.

$$AX + Y - Q'$$

where $Y$ is a column vector whose entries are all equal to $x_m = \ldots = x_m$.

This can be written as

$$A'X' = Q'$$

where $A' = A + B$ and $X'$ is the concatenated vector of $X$ and $Y$.

Clearly, (2) corresponds to $m$ SUM queries of query set
size $k = k^* + \lambda - \lambda^*$ concerning $m + \lambda - \lambda^*$ individuals,
with overlap $\lambda$.

Now we can solve (1) for $x_1, x_2, \ldots, x_m$ as

$$X = A^{-1}Q' - \frac{1}{\lambda}Y$$

Hence the order of magnitude of $x_1, x_2, \ldots, x_m$ is the same as the
order of magnitude of the entries in $A^{-1}Q'$.

(ii) $\lambda > l - m$
Then we can use the $m \times m$ identity matrix $I$ in place of $A$
and the proof is essentially the same as for Case (i).

Example

Complete compromise is possible by asking the following
seven queries about seven individuals of a database ($k^* =
1, k^* = 3$).

$$x_1 + x_2 + x_3 = q_1$$
$$x_3 + x_2 + x_4 = q_2$$
$$x_4 + x_3 + x_5 = q_3$$
$$x_5 + x_4 + x_6 = q_4$$
$$x_6 + x_5 + x_7 = q_5$$
$$x_7 + x_6 + x_8 = q_6$$
$$x_8 + x_7 + x_9 = q_7$$

We can write this as

$$AX = Q$$

where the query matrix

$$A = \begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 \\
\end{bmatrix}$$

If $q_1 = 500, q_3 = 140, q_5 = 250, q_7 = 270, q_9 = 140, q_2 = 230$
then $x_1 = 100, x_3 = 200, x_5 = 0, x_7 = 10, x_9 = 30, x_2 = 40$.

We shall use the matrix $A$ to form seven queries about nine
individuals as follows. Let $X$ be the column vector
$(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ and let $Y$ be the column vector with
entries all equal to $x_9 + x_8$. Then we can construct the SUM
queries corresponding to

$$AX + Y = Q'$$

Then $X = A^{-1}Q' - \frac{1}{\lambda}(x_9 + x_8)$ where

$$A^{-1} = \frac{1}{k} \begin{bmatrix}
2 & -1 & 2 & -1 & -1 & 2 & -1 \\
-2 & -1 & 2 & -1 & 1 & 1 & -1 \\
2 & -1 & 2 & -1 & 1 & -1 & 1 \\
-2 & -1 & 2 & 1 & -1 & 1 & 1 \\
-2 & -1 & 2 & 1 & -1 & 1 & 1 \\
-2 & -1 & 2 & 1 & -1 & 1 & 1 \\
-2 & -1 & 2 & 1 & -1 & 1 & 1 \\
\end{bmatrix}$$

Suppose the responses to the SUM queries were 550, 260, 190, 300, 320, 190, 280. Then we can express

$$x_1, x_2, x_3, x_4, x_5, x_6, x_7$$
in terms of $x_9, x_8$ as

$$x_1 = \frac{700}{6} - \frac{1}{3}(x_9 + x_8)$$
$$x_2 = \frac{1300}{6} - \frac{1}{3}(x_9 + x_8)$$
$$x_3 = \frac{1300}{6} - \frac{1}{3}(x_9 + x_8)$$
$$x_4 = \frac{100}{6} - \frac{1}{3}(x_9 + x_8)$$
$$x_5 = \frac{160}{6} - \frac{1}{3}(x_9 + x_8)$$
$$x_6 = \frac{260}{6} - \frac{1}{3}(x_9 + x_8)$$
$$x_7 = \frac{340}{6} - \frac{1}{3}(x_9 + x_8)$$

and so $x_9 > x_3 < x_4 < x_5 < x_6 < x_9 < x_7 < x_9 < x_9 < x_9 < x_9$.

Note that the relative compromise of Theorem 3 is only
possible for overlap $\lambda \geq l - m$ or, equivalently, for the
number of queries $m \geq l - \lambda$. Thus there is a trade off
between the number of queries and the overlap.

If we allow $m$ queries about $m + 1$ individuals and $m$
is such that the block design $(m + 1, k, \lambda)$ (Street, 1977) exists
then it is possible to get relative compromise given the
overlap $\lambda > 0$ and fixed query set size $k$. However, in this
case we obtain two disjoint sets of relatively compromised
individuals.

Theorem 4: Let $S$ be a subset of a database where $|S| =
m + 1$. If the block design $(m + 1, k, \lambda)$ exists then a subset of
$S$ can be relatively compromised using $m$ SUM queries
with overlap $\lambda$ and query set size $k$.
Proof: If \(m, k\) and \(\lambda\) are such that the block design \((m+1, k, \lambda)\) exists then we can use for the query matrix the first \(m\) rows of the incidence matrix, \(D\), of the design. For the sake of our calculations we assume that the \(m+1\)st row was also used and that the answer was some unknown constant, say \(C\).

Thus

\[
DX = Q
\]

where \(X^T = (x_1x_2, \ldots, x_mx_m)\) and \(Q^T = (q_1q_2, \ldots, q_m)\), where \(C\) is not known. Since \(D\) is the incidence of an \((m+1, k, \lambda)\) design it satisfies

\[
DJ = kJ
\]

and

\[
D^{-1} = \frac{1}{k-\lambda} D^T - \frac{\lambda}{k(k-\lambda)} J
\]

Hence

\[
X = \left(\frac{1}{k-\lambda} D^T - \frac{\lambda}{k(k-\lambda)} J\right) Q
\]

and so \(x_p, x_2, \ldots, x_m\) can be expressed in terms of the unknown \(C\)

\[
x_i = q_i - c_i C
\]

where \(c_i\) takes on two different values, say \(c_1\) and \(c_2\). Now we can compare all the \(x_i\) with the corresponding \(c_i\) value equal to \(c_1\) and find their relative order of magnitude. Similarly, we can also order the elements \(x_i\) where the corresponding value of \(c_1\) is \(c_2\).

Example

Suppose we were allowed to make the 6 SUM queries about 7 individuals which correspond to the following set of 6 linearly independent equations in 7 unknowns

\[
\begin{align*}
x_1 + x_3 + x_7 &= 1300 \\
x_1 + x_2 + x_3 &= 230 \\
x_2 + x_3 + x_4 &= 1010 \\
x_3 + x_4 + x_5 &= 1030 \\
x_4 + x_5 + x_6 &= 120 \\
x_1 + x_6 + x_7 &= 210
\end{align*}
\]

The query matrix of these 6 queries corresponds to the first 6 rows of the 7 \(\times\) 7 incidence matrix \(D\) of the \((7,3,1)\) block design. Suppose we asked the query corresponding to the last row of the matrix \(D\) and the system refused to answer that query. That is, we have

\[
x_1 + x_6 + x_7 = C
\]

where \(C\) is unknown. Then the system of equations can be written as

\[
DX = Q
\]

where

\[
X^T = (x_1x_2x_3x_4x_5x_6x_7), \quad Q^T = (1300, 230, 1010, 1030, 120, 210, C)
\]

\[
D = \begin{bmatrix}
1 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 1
\end{bmatrix}
\]

The inverse of \(D\) is

\[
D^{-1} = \begin{bmatrix}
-2 & 2 & -1 & -1 & -1 & 2 & -1 \\
-1 & 2 & 2 & -1 & -1 & -1 & 2 \\
-1 & 2 & -1 & 2 & 2 & -1 & -1 \\
-1 & -1 & 2 & -1 & 2 & 2 & -1 \\
-1 & -1 & -1 & 2 & -1 & 2 & -1 \\
-2 & -1 & -1 & -1 & 2 & -1 & 2
\end{bmatrix}
\]

\[
X = D^{-1}Q = \begin{bmatrix}
1320 - C \\
-180 + 2C \\
6120 - C \\
240 - C \\
120 - C \\
-180 + 2C \\
360 + 2C
\end{bmatrix}
\]

Thus we can calculate the values of \(x_1, x_2, x_3, x_4, x_5, x_6, x_7\) in terms of \(C\) and since \(C\) appears with only two different coefficients we can find the relative order of magnitude of two subsets of the 7 unknowns, namely

\[
x_5 < x_4 < x_3 < x_2, \quad \text{and} \quad x_6 < x_7.
\]

Note that if in the above example the sixth query were also disallowed so that the result of that query was unknown, say \(C\), then we could still get some relative compromise. In particular, we could compare \(x_1\) with \(x_2\); \(x_3\) with \(x_4\); and \(x_1\) with \(x_5\), This is easily seen by inspecting the last two columns of the inverse matrix \(D^{-1}\).

In general, using the incidence matrix of a block design \((m+1, k, \lambda)\) for a query matrix we can guarantee to get some relative compromise whenever \(m+1-n\) queries about \(m+1\) unknowns (with constant query set size \(k\) and overlap \(\lambda\)) are allowed provided \(m+1 > 2^n\).

CONCLUSION

In this paper we have described the idea of relative compromise of statistical databases and some conditions under which it can occur using SUM type of queries. We also showed that in some cases relative compromise with supplementary knowledge can lead to a more serious compromise of a database. It remains an open problem to find the general conditions (in terms of the number of records in a database, query set size, query overlap and the minimum number of queries needed) under which relative compromise can occur.
ACKNOWLEDGEMENT
The advice and suggestions by the referees has been much appreciated by the authors.

REFERENCES

BIOPGRAPHICAL NOTES
Jennifer Seberry is Professor and Head of the Computer Science Department at University College, ADFA, Canberra. Dr Seberry is especially interested in Cryptography, Authentication and Computer Security. She has co-authored six books including Cryptography: An Introduction to Computer Security with Joseph Pieprzyk (Prentice Hall, 1988) and The Cryptographic Significance of the Knapsack Problem with Luke O'Connor (Aegaean Press, 1988). Professor Seberry is a member of the Australian Computer Society and the Australian Operations Research Society since 1970. She is also a member of the International Association for Cryptologic Research.

Mirka Miller is a lecturer in the Department of Mathematics, Statistics and Computing Science, University of New England, Armidale. Her main interests in computer science are databases, especially security and integrity of databases. She is a member of the Australian Computer Society, the European Association for Theoretical Computer Science and the Australian Mathematical Society. Ms Miller is currently working on her PhD thesis in security of databases, under the supervision of Professor Seberry.

CALL FOR PAPERS
AUSTRALIAN COMPUTER SCIENCE CONFERENCE
ACSC-13
7-9 February 1990

ACSC-13 will be held at Monash University, Melbourne. Papers in all fields of Computer Science are invited. As in previous years, papers will be refereed, and the proceedings published as Australian Computer Science Communications Vol. 12, No. 1. Presentation at ACSC is not held to preclude later publication elsewhere.

Papers should be submitted by 1 September 1989. Refereeing will be completed in time to allow revision, where needed, by 24 November 1989.

Papers should be submitted as nearly as possible in final form, in A4 format, and should not exceed ten pages. The first page should include title and abstract, and should leave space for but not include author's name and affiliation. Authors should supply an extra page giving name(s), address(es), e-mail address(es) and phone numbers, with paper title. Type font should be not smaller than 10 point save in footnotes etc. Page layout should allow a 25 mm margin for binding. (Assume each paper will begin on a right-hand page.) Please supply three copies of the paper and of the extra page.

CALL FOR REFEREES
We would be grateful to receive offers for refereeing ACSC-13 papers. Volunteers should indicate address, phone, email address, and area(s) of expertise. There is no bar on contributors also acting as referees.

Address for Papers and Referee Offers:
ACSC-13
Computer Science, Monash University, Clayton, Victoria 3168

Email addresses for correspondence:
(Re papers) acsc13papers@bruce.oz
(Re refereeing) acsc13referees@bruce.oz