SIAM J. COMPUT.
Vol. 23, No. 1, pp. 170–184, February 1994


012

# THREE-DIMENSIONAL STATISTICAL DATA SECURITY PROBLEMS*

ROBERT W. IRVING† AND MARK R. JERRUM‡

**Abstract.** Suppose there is a three-dimensional table of cross-tabulated nonnegative integer statistics, and suppose that all of the row, column, and "file" sums are revealed together with the values in some of the individual cells in the table. The question arises as to whether, as a consequence, the values contained in some of the other (suppressed) cells can be deduced from the information revealed.

The corresponding problem in two dimensions has been comprehensively studied by Gusfield [*SIAM J. Comput.*, 17 (1988), pp. 552–571], who derived elegant polynomial-time algorithms for the identification of any such "compromised" cells, and for calculating the tightest bounds on the values contained in all cells that follow from the information revealed. In this note it is shown, by contrast, that the three-dimensional version of the problem is NP-complete.

It is also shown that if the suggested row, column, and file sums for an unknown three-dimensional table are given, with or without the values in some of the cells, the problem of determining whether there exists any table with the given sums is NP-complete. In the course of proving these results, the NP-completeness of some constrained Latin square construction problems, which are of some interest in their own right, is established.

**Key words.** data security, NP-complete problems, Latin squares

**AMS subject classifications.** 68R05, 68Q15, 05B15

## 1. Introduction.

Problems of statistical data security in two dimensions have been studied extensively—see [1] for some indications of early work and [3], [4], and [6] for some recent developments. In this note, we study the obvious extension of the problem to three dimensions, raised as an open problem by Gusfield [3], and show that, as is the case in a variety of other contexts, problems that are solvable in polynomial time in the two-dimensional case become NP-complete when extended to three dimensions.

Consider a three-dimensional table $D$, of size $n \times n \times n$, of nonnegative integer values $D(i, j, k)$, $(1 \leq i, j, k \leq n)$. The table entries $D(i, j, k)$ for fixed $i, k$ and $1 \leq j \leq n$ constitute a *row* of the table, for fixed $j, k$ and $1 \leq i \leq n$ a *column* of the table, and for fixed $i, j$ and $1 \leq k \leq n$ a *file* of the table.

Envisage that a particular table $D$ represents a collection of cross-tabulated statistical data, and that the row, column, and file sums of $D$ are to be disclosed together with the values contained in some of the cells. However, other cells may contain sensitive values that are therefore to be suppressed; the question arises as to whether knowledge of the row, column, and file sums together with the disclosed cells will (a) essentially fix the values of one or more of the suppressed cells and (b) allow such values to be deduced in reasonable (say, polynomial) time (say, by an adversary). A suppressed cell that has the same value in all legal tables, i.e., all tables satisfying the row, column, and file sums and containing the disclosed values, is said to be *compromised*. A suppressed cell that is not compromised is said to be *protected*.

In the case of the corresponding two-dimensional problem, Gusfield [3] gives a $O(n^3)$ algorithm to identify all the fixed cells and to calculate their values. In [3] and [4], Gusfield also describes polynomial-time algorithms to calculate the tightest bounds on the protected cells. As part of this work, a $O(n^3)$ algorithm is presented for the identification of a legal solution in the two-dimensional case. In this context, it turns out that the obvious necessary conditions for a legal solution—namely, that the sums of the row and column sums should be equal and that each row and column sum should be at least equal to the sum of the disclosed entries in that row or column—are also sufficient for the existence of a legal solution.

*Received by the editors November 19, 1990; accepted for publication (in revised form) October 7, 1992.
†Computing Science Department, University of Glasgow, Glasgow, G12 8QQ United Kingdom.
‡Computer Science Department, University of Edinburgh, Edinburgh, EH9 3JZ United Kingdom.

By contrast, in the three-dimensional case, we shall show that the obvious necessary two-dimensional conditions on the row, column, and file sums are not sufficient to guarantee the existence of a legal solution, and indeed that the problem of determining whether a legal solution exists is NP-complete. This result holds even in the interesting special case in which all the cells are suppressed; in other words, if we are given the row, column, and file sums, and are asked whether a legal solution exists. We shall then proceed to show that the NP-completeness of the existence problem also implies the NP-completeness of the problem of identifying compromised cells, at least in the general case where some cells may be revealed.

The NP-completeness proofs involve consideration of special cases that are equivalent to two-dimensional problems of Latin square construction, which are of some interest in their own right. For instance, we show that a special case of the problem of the existence of a legal solution is equivalent to the problem of constructing an $n \times n$ Latin square given independent restricted choices for the various entries, and that this Latin square construction problem is itself NP-complete.

**2. Formal statement of the problem.** Throughout, we assume that, unless otherwise stated, the row, column, and file indices $i$, $j$, and $k$ range over the values $1, \ldots, n$ wherever appropriate. Suppose that for a given $n \times n \times n$ table $D$ of nonnegative integers, and for each $i$, $j$, $k$, the row, column, and file sums are denoted by $R(i, k)$, $C(j, k)$, and $F(i, j)$, respectively. In other words,

$$R(i,k) = \sum_{j=1}^{n} D(i,j,k), \tag{1}$$

$$C(j,k) = \sum_{i=1}^{n} D(i,j,k), \tag{2}$$

$$F(i,j) = \sum_{k=1}^{n} D(i,j,k). \tag{3}$$

If we represent the set of suppressed cells by $S$, i.e., $S = \{(i, j, k) : D(i, j, k)$ not disclosed$\}$, then we can calculate the reduced row, column, and file sums, namely,

$$R^*(i,k) = R(i,k) - \sum_{j:(i,j,k)\notin S} D(i,j,k), \tag{4}$$

$$C^*(j,k) = C(j,k) - \sum_{i:(i,j,k)\notin S} D(i,j,k), \tag{5}$$

$$F^*(i,j) = F(i,j) - \sum_{k:(i,j,k)\notin S} D(i,j,k). \tag{6}$$

We are now in a position to state formally the problem that we wish to pursue.

**2.1. Three-dimensional statistical data—legal solutions (3DSDLS).**

*Instance*: A positive integer $n$, nonnegative integer values $R^*(i, k)$, $C^*(j, k)$, and $F^*(i, j)$, and a subset $S$ of $N^3$, where $N = \{1, 2, \ldots, n\}$.

*Question*: Does there exist an assignment of nonnegative values to $D(i, j, k)$ for $(i, j, k) \in S$ such that

$$\sum_{j=1}^{n} D(i,j,k) = R^*(i,k), \tag{7}$$

$$\sum_{i=1}^{n} D(i, j, k) = C^*(j, k), \tag{8}$$

$$\sum_{k=1}^{n} D(i, j, k) = F^*(i, j), \tag{9}$$

where each sum is taken over values $(i, j, k) \in S$?

It is immediate that, in order for a solution to exist, the sums of the $R^*$, $C^*$, and $F^*$ values must all be identical and satisfy constraints imposed by consideration of the two-dimensional "slices" of the table, namely,

$$\sum_{k=1}^{n} R^*(i, k) = \sum_{j=1}^{n} F^*(i, j) \quad (1 \le i \le n), \tag{10}$$

$$\sum_{i=1}^{n} F^*(i, j) = \sum_{k=1}^{n} C^*(j, k) \quad (1 \le j \le n), \tag{11}$$

$$\sum_{j=1}^{n} C^*(j, k) = \sum_{i=1}^{n} R^*(i, k) \quad (1 \le k \le n). \tag{12}$$

To see that these necessary conditions are by no means sufficient for the existence of a legal solution, we consider an example in which $n = 2$ and all cells are suppressed.

*Example.* It may be checked easily by exhaustive search that the 3DSDLS instance shown in Fig. 1, in which the row, column, and file sums appear as labels on the appropriate arrows, admits no legal solution, although the necessary two-dimensional conditions are satisfied.

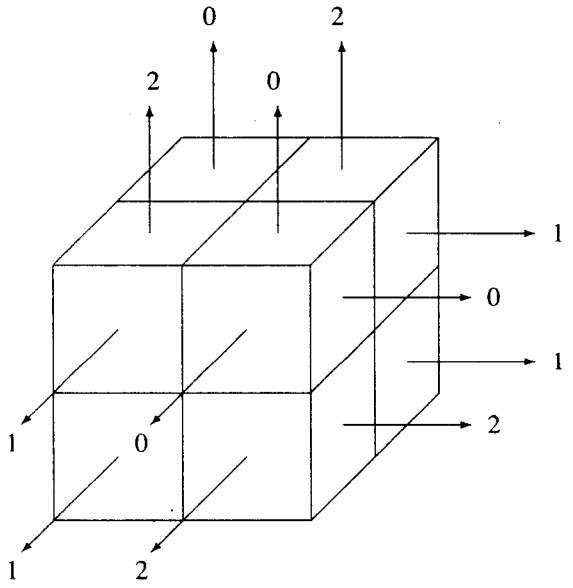



FIG. 1. *An instance of* 3DSDLS *of size* 4 *with no legal solution.*

In order to show that the 3DSDLS problem is NP-complete, we shall restrict our attention to a special case, which can be interpreted as the problem of constructing a Latin square

given restricted choices for the various entries. This special case involves setting $R^*(i, k) = C^*(j, k) = F^*(i, j) = 1$ for all values of $i$, $j$, and $k$. If we define

$$S(i, j) = \{k : (i, j, k) \in S\},$$

then our task is to find a suppressed cell in each file, i.e., an element in $S(i, j)$, which can be given the value 1, subject to the constraint that no two cells can be chosen from the same row and no two can be chosen from the same column. It should be clear that this can be interpreted as the problem of constructing a Latin square of size $n \times n$ where the choice of element in position $(i, j)$ is restricted to the set $S(i, j)$.

**2.2. Latin square construction (LSC).**

*Instance*: A positive integer $n$, and for each $i$, $j$, a subset $S(i, j)$ of $\{1, \ldots, n\}$.

*Question*: Does there exist a Latin square $X$ of size $n \times n$ such that, for all $i$, $j$, $X(i, j) \in S(i, j)$?

*Example*. The first array below constitutes a "yes"-instance of LSC of size 4, with underlined entries indicating one solution. On the other hand, exhaustive search will reveal that the second array constitutes a "no"-instance.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $\{3, \underline{4}\}$ | $\{\underline{1}, 2\}$ | $\{1, \underline{3}, 4\}$ | $\{1, \underline{2}\}$ | $\{1,2\}$ | $\{1,3\}$ | $\{2,4\}$ | $\{3,4\}$ |
| $\{\underline{2}, 4\}$ | $\{1, \underline{3}\}$ | $\{\underline{1}, 2\}$ | $\{3, \underline{4}\}$ | $\{3,4\}$ | $\{2,4\}$ | $\{1,2\}$ | $\{1,3\}$ |
| $\{\underline{1}, 3\}$ | $\{2, \underline{4}\}$ | $\{1, \underline{2}\}$ | $\{1, \underline{3}, 4\}$ | $\{1,2\}$ | $\{3,4\}$ | $\{1,3\}$ | $\{2,4\}$ |
| $\{1, 2, \underline{3}\}$ | $\{\underline{2}, 3\}$ | $\{3, \underline{4}\}$ | $\{\underline{1}, 2, 3\}$ | $\{3,4\}$ | $\{1,2\}$ | $\{3,4\}$ | $\{1,2\}$ |

Clearly a proof of NP-completeness for LSC implies that the more general 3DSDLS problem is NP-complete also.

**3. NP-completeness of LSC.** In preparation for the proof of NP-completeness of LSC, we need to investigate the conditions under which a Latin rectangle may be extended to a larger Latin rectangle or to a full Latin square. The following result is well known—see [5] or [7].

LEMMA 3.1. *An arbitrary Latin rectangle of size $m \times n$ $(m < n)$ over a ground-set of size $n$ may be extended by the addition of $n - m$ additional rows to form a Latin square of size $n \times n$.*

In addition, we need a sufficient condition for a $p \times q$ Latin rectangle to be extendable to an $n \times q$ Latin rectangle. The following result may not be the best possible, but it will suffice for our purposes.

LEMMA 3.2. *Suppose that $L$ is a Latin rectangle of size $p \times q$ with elements chosen from a ground-set of size $n$, and suppose that $n \geq p + 2q - 2$. Then $L$ can be extended to a Latin rectangle of size $n \times q$.*

*Proof.* Suppose that the first $j$ columns of $L$ have already been extended to length $n$, where $0 \leq j \leq q - 1$; we show that the $(j + 1)$th column can also be extended to length $n$.

Consider a bipartite graph $G$ with vertex set $V = U \cup W$. In $U$ there are $s = n - p$ vertices, one for each of rows $p + 1, \ldots, n$ in the rectangle, and in $W$ there are $n$ vertices, one for each element of the ground-set, which we may take to be $\{1, \ldots, n\}$. Vertices $u \in U$ and $w \in W$ are joined if and only if element $w$ already appears neither in column $j + 1$ nor in row $u$ (and therefore $w$ is a candidate for position $(u, j)$ in the rectangle). It is clear that the $(j + 1)$th column can be extended to length $n$ if and only if the graph $G$ has a matching of size $s$.

By Hall's theorem, $G$ will have a matching of size $s$ provided that, for each $k$ $(1 \leq k \leq s)$, the vertices in every $k$-subset of $U$ are collectively adjacent to at least $k$ vertices in $W$. We consider two cases:

*Case* (a) $k > j$. In this case, no element can appear in all the $k$ rows corresponding to the $k$ vertices of $U$. Hence, these $k$ vertices are collectively adjacent to all $n$ vertices in $W$, except for the $p$ vertices corresponding to the elements in column $j+1$, therefore, to $n-p$ vertices. Since $n-p \geq k$, the required condition for a matching of size $s$ is met.

*Case* (b) $k \leq j$. In this case, it is possible that the $k$ rows have up to $k$ elements in common, so that the best we can claim is that the $k$ vertices in $U$ are collectively adjacent to at least $n-p-k$ vertices in $W$. Nonetheless, since $n-p \geq 2(q-1)$, $q-1 \geq j$, and $j \geq k$, it follows that $n-p-k \geq k$, and the required condition is again satisfied.

Hence, by Hall's theorem, $G$ has a matching of size $s$, and therefore the $(j+1)$th column can be extended to length $n$ as claimed. □

We are now in a position to prove the NP-completeness of our Latin square construction problem LSC.

THEOREM 3.3. *Latin square construction is NP-complete.*

*Proof.* Membership in NP is immediate, for we need simply guess an element $X(i, j) \in S(i, j)$ for each $i$, $j$, and it is straightforward to verify, in polynomial time, whether the resulting square $X$ is a Latin square.

To show that LSC is NP-complete, we describe a polynomial-time transformation from the known NP-complete problem 3-Satisfiability (3-SAT)—see [2].

Given an instance of 3-SAT involving $m$ variables $v_1, v_2, \ldots, v_m$ and $n$ clauses $C_1, C_2, \ldots, C_n$, we construct an instance of LSC of size $2mn$, which admits a Latin square if and only if the original 3-SAT instance is satisfiable. Corresponding to each $v_k$ there are precisely $2n$ elements in the ground-set $S$ for the derived instance of LSC, denoted by $u_{kl}, \overline{u}_{kl}$ $(1 \leq l \leq n)$. It remains to describe the sets $S(i, j)$ $(1 \leq i, j \leq 2mn)$.

For $1 \leq k \leq m$, $1 \leq l \leq n$, we define

$$\begin{aligned} S((k-1)n+l, 1) &= \{u_{kl}, \overline{u}_{kl}\}, \\ S((k-1)n+l, 2) &= \{u_{kl}, u_{k,l+1}\}, \\ S((k-1)n+l, 3) &= \{u_{k,l+1}, \overline{u}_{k,l}\}, \end{aligned}$$

where here, and subsequently, $l+1$ is taken modulo $n$ in the range $[1, n]$.

For $1 \leq l \leq n$ we define

$$S(mn+l, 1) = \{w_{1l}, w_{2l}, w_{3l}\},$$

where

$$w_{hl} = \begin{cases} u_{kl} & \text{if the } h\text{th literal in } C_l \text{ is } v_k, \\ \overline{u}_{kl} & \text{if the } h\text{th literal in } C_l \text{ is } \overline{v}_k. \end{cases}$$

Finally, for all subscript pairs $i$, $j$ not covered by the above, we set $S(i, j) = S$. It is clear that the entire construction can be carried out in time bounded by a polynomial in the length of the original 3-SAT expression.

We now have to establish that the derived instance of LSC admits the construction of a $2mn \times 2mn$ Latin square if and only if the original instance of 3-SAT has a satisfying assignment.

Suppose first that the LSC instance admits a Latin square $X$. For a given value of $k$, consider positions $(k-1)n+l, t$ $(1 \leq l \leq n, 1 \leq t \leq 3)$ in the square. It is straightforward to verify that there are just two possibilities, either

(a)
$$X((k-1)n+l, 1) = \bar{u}_{kl},$$
$$X((k-1)n+l, 2) = u_{kl},$$
$$X((k-1)n+l, 3) = u_{k,l+1}$$

for all $l$ $(1 \le l \le n)$, or

(b)
$$X((k-1)n+l, 1) = u_{kl},$$
$$X((k-1)n+l, 2) = u_{k,l+1},$$
$$X((k-1)n+l, 3) = \bar{u}_{kl}$$

for all $l$ $(1 \le l \le n)$.

We assign variable $v_k$ to be true or false accordingly as Case (a) or Case (b) applies.

Now, for each $l$, $1 \le l \le n$, we consider $X(mn+l, 1)$. Because $X$ is a Latin square, we must have $X(mn+l, 1)$ equal to $w_{hl}$, where $w_{hl}$ represents a true literal; otherwise, by (a) and (b) above, we would have $X(mn+l, 1) = X((k-1)n+l, 1)$ for some $k$. So, in the derived assignment, every clause contains at least one true literal, showing that the original instance of 3-SAT is satisfiable.

On the other hand, suppose that the instance of 3-SAT is satisfiable, and consider a particular satisfying assignment. If $v_k$ is true in this assignment, we choose

$$X((k-1)n+l, 1) = \bar{u}_{kl},$$
$$X((k-1)n+l, 2) = u_{k,l},$$
$$X((k-1)n+l, 3) = u_{k,l+1}$$

for all $l$ $(1 \le l \le n)$, while if $v_k$ is false, we choose

$$X((k-1)n+l, 1) = u_{kl},$$
$$X((k-1)n+l, 2) = u_{k,l+1},$$
$$X((k-1)n+l, 3) = \bar{u}_{kl}$$

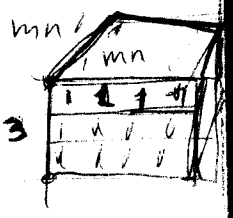
for all $l$ $(1 \le l \le n)$.

Further, for $1 \le l \le n$, we choose

$$X(mn+l, 1) = w_{hl},$$

where $w_{hl}$ is a true literal in clause $C_l$, and

$$X(mn+l, 2) = \bar{u}_{1,l+1},$$
$$X(mn+l, 3) = \begin{cases} u_{1,l+1} & \text{if } v_1 \text{ is false,} \\ \bar{u}_{1,l+2} & \text{if } v_1 \text{ is true.} \end{cases}$$




It is straightforward to verify that this gives a Latin rectangle of size $(m+1)n \times 3$. Provided $(m-1)n \ge 4$, which can be assumed without loss of generality, the condition of Lemma 3.2

is satisfied; this Latin rectangle can then be extended, first, to form a Latin rectangle of size $2mn \times 3$, and then, by Lemma 3.1, to form a Latin square of size $2mn \times 2mn$, bearing in mind that $S(i, j) = S$ for all outstanding positions $(i, j)$. □

In view of the earlier observation that LSC is a special case of 3DSDLS, we have the following corollary.

COROLLARY 3.4. *3DSDLS is NP-complete, even in the special case where all the row, column and file sums are* 1.

**4. The special case of all cells suppressed.** We now consider the interesting special case of 3DSDLS in which all cells are suppressed. This special case is a natural problem in its own right, which we refer to as the 3D contingency table problem (3DCT).

**4.1 Three-dimensional contingency tables (3DCT).**

*Instance*: A positive integer $n$, and for each $i, j, k$, nonnegative integer values $R(i, k)$, $C(j, k)$, $F(i, j)$.

*Question*: Does there exist an $n \times n \times n$ contingency table X of nonnegative integers such that

$$\sum_{j=1}^{n} X(i, j, k) = R(i, k),$$

$$\sum_{i=1}^{n} X(i, j, k) = C(j, k),$$

$$\sum_{k=1}^{n} X(i, j, k) = F(i, j)$$

for all $i, j, k$?

We now establish the NP-completeness of 3DCT even in the special case where all the $R$, $C$, and $F$ values are 0 or 1. This we achieve by viewing this special case as a two-dimensional partial Latin square construction problem, and by showing that there is a polynomial-time transformation from the basic Latin square construction problem, already known to be NP-complete, to this new problem.

The partial Latin square construction problem is specified in the following section.

**4.2 Partial Latin square construction (PLSC).**

*Instance*: A positive integer $n$, subsets $R(i)$ and $C(j)$ of $N = \{1, \ldots, n\}$ for each $i, j$, and a subset $\mathcal{N}$ of $N^2$ such that (a) $|R(i)| = |k : (i, k) \in \mathcal{N}|$, and (b) $|C(j)| = |k : (k, j) \in \mathcal{N}|$.

*Question*: Does there exist a partial Latin square $X$ with

(i) $X(i, j)$ defined for all $(i, j) \in \mathcal{N}$;

(ii) $X(i, j) \in R(i) \cap C(j)$ for all such $(i, j)$?

Note that it follows at once that, in a "yes" instance of PLSC, (a) $k \in R(i) \Rightarrow X(i, j) = k$ for some $j$, and (b) $k \in C(j) \Rightarrow X(i, j) = k$ for some $i$.

*Example.* In Fig. 2, the sets $R(i)$ and $C(j)$ appear to the left of the rows and above the columns, respectively; the set $R(i) \cap C(j)$ appears in position $(i, j)$; cells corresponding to pairs $(i, j) \notin \mathcal{N}$ contain the symbol —; and the underlined elements indicate that this is a "yes"-instance of the problem.

Our first objective is to show that the special case of 3DCT in which the row, column, and file sums are all 0 or 1 can be interpreted as an instance of PLSC. Given such an instance of 3DCT of size $n$ with row, column, and file sums $R(i, k)$, $C(j, k)$, and $F(i, j)$, define

$$\mathcal{N} = \{(i, j) : F(i, j) = 1\},$$

$$R(i) = \{k : R(i, k) = 1\},$$

$$C(j) = \{k : C(j, k) = 1\}.$$

| | {1, 3} | {1, 2, 4} | {3, 4} | {1, 2, 3, 4} |
|---|---|---|---|---|
| {1, 2, 3} | — | $\underline{1}$, 2 | $\underline{3}$ | 1, $\underline{2}$, 3 |
| {1, 4} | $\underline{1}$ | — | — | 1, $\underline{4}$ |
| {2, 3, 4} | — | $\underline{2}$, 4 | 3, $\underline{4}$ | 2, $\underline{3}$, 4 |
| {1, 3, 4} | 1, $\underline{3}$ | 1, $\underline{4}$ | — | $\underline{1}$, 3, 4 |

FIG. 2. *An instance of* PLSC *of size* 4.

The set $\mathcal{N}$ represents the set of files with sum equal to 1. The sole position in such a file, say, file $(i, j)$, occupied by a 1 must correspond to both a row with sum equal to 1—i.e., an element of $R(i)$—and a column with sum equal to 1—i.e., an element of $C(j)$. It follows that a solution to the given instance of the special case of 3DCT corresponds exactly to a solution of the derived instance of PLSC.

We are now in a position to establish the NP-completeness of PLSC.

THEOREM 4.1. *The partial Latin square construction problem* (PLSC) *is* NP-*complete.*

*Proof.* Membership in NP is immediate, since we can guess an element in the ground-set for each cell in the set $\mathcal{N}$ and verify in polynomial time that the various row and column constraints are satisfied. To prove NP-completeness, we describe a polynomial-time transformation from the basic Latin square construction problem (LSC), already known to be NP-complete by Theorem 3.3.

Given a positive integer $n$ and sets $S(i, j) \subseteq N = \{1, \ldots, n\}$ forming an instance of LSC, we construct an instance of PLSC over the same ground-set $N$ but of increased size $n' = n + \sum_{ij} |S(i, j)|$. The set of free cells $\mathcal{N}$ is defined as a disjoint union of $n^2$ component sets $\mathcal{N}_{ij}$, which are constructed as follows. Let $\lambda, \mu$ be mappings from $N^2$ to $\{n+1, n+2, \ldots, n'\}$ satisfying

(i) $\mu(i, j) - \lambda(i, j) + 1 = |S(i, j)|$, for all $i, j \in N$;

(ii) the intervals $\{[\lambda(i, j), \mu(i, j)] : i, j \in N\}$ form a partition of the set $\{n+1, n+2, \ldots, n'\}$. (The notation $[a, b]$ denotes the set of all integers not less than $a$ and not greater than $b$.)

For each $i, j$ in the range $1 \leq i, j \leq n$, define

$$\mathcal{N}_{ij} = [l, m] \times [l, m] \cup \{(i, l), (l, j)\} - \{(l, l)\},$$

where $l = \lambda(i, j)$ and $m = \mu(i, j)$; the set of free cells is then $\mathcal{N} = \bigcup_{ij} \mathcal{N}_{ij}$. Figure 3 is a pictorial representation of a typical component set $\mathcal{N}_{ij}$.

We now specify the row and column sets. For $k$ in the range $1 \leq k \leq n$, set $R(k) = C(k) = N$. For $k$ outside this range, i.e., for $n+1 \leq k \leq n'$, let $i, j \in N$ be the unique integers satisfying $\lambda(i, j) \leq k \leq \mu(i, j)$, and set $R(k) = C(k) = S(i, j)$. It can be checked that $R$, $C$, and $\mathcal{N}$ together form a consistent instance of PLSC. This completes the construction.

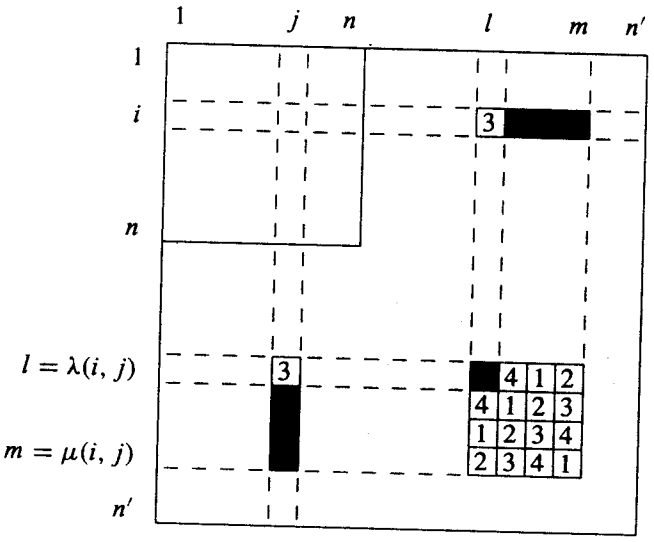



FIG. 3. *The set $\mathcal{N}_{ij}$, and a possible numbering when $S(i, j) = \{1, 2, 3, 4\}$.*

To verify the construction, we must show that the derived instance of PLSC admits a solution if and only if the original instance of LSC does. Suppose first that the LSC instance admits a Latin square $X$, so that $X(i, j) \in S(i, j)$ for all $i, j$. We shall construct a partial Latin square $Y$ consistent with the derived instance of PLSC, i.e., satisfying $Y(i, j) \in R(i) \cap C(j)$ for all $(i, j) \in \mathcal{N}$.

It is enough to describe the restriction of $Y$ to a typical component set $\mathcal{N}_{ij}$, since $\mathcal{N}$ is a disjoint union of such sets. Let $l = \lambda(i, j)$, $m = \mu(i, j)$, and $s = |S(i, j)|$. We consider the domain $\mathcal{N}_{ij}$ in two parts: the square with missing corner cell $[l, m] \times [l, m] - \{(l, l)\}$, and the two isolated cells $(i, l)$ and $(l, j)$. On the isolated cells, we simply take $Y(i, l) = Y(l, j) = X(i, j)$. In order to deal with the square, imagine that the missing corner cell is temporarily reinstated. Define $Y$ on this completed square so that the resulting $s \times s$ table of values forms a Latin square over the ground-set $S(i, j)$, with $Y(l, l) = X(i, j)$. Then simply remove the cell $(l, l)$ from the domain of definition of $Y$. Figure 3 is intended to illustrate how the restriction of $Y$ to the set $\mathcal{N}_{ij}$ might appear in the case $S(i, j) = \{1, 2, 3, 4\}$.

It may readily be checked that $Y(i, j) \in R(i) \cap C(j)$ for all cells $(i, j) \in \mathcal{N}$. The only other condition we must verify is that $Y$ is indeed a partial Latin square, i.e., that no row or column of $Y$ contains a duplicate value. By symmetry, we need only check this condition for the rows. For $i$ in the range $n + 1 \leq i \leq n'$, the fact that row $i$ of $Y$ cannot contain duplicate values is clear by construction. So suppose $1 \leq i \leq n$, which is the only other possibility. Row $i$ of the derived instance of PLSC contains precisely $n$ cells that are elements of $\mathcal{N}$, and these correspond to the $n$ cells forming row $i$ in the original instance of LSC. $Y$ assigns to each of the $n$ cells in the PLSC instance the same value that $X$ assigns to the corresponding cell in the LSC instance, and these $n$ values must all be distinct. Thus we have shown that the PLSC instance admits a solution if the LSC instance does.

For the converse, suppose that the derived PLSC instance admits a partial Latin square $Y$ that is consistent with the various row and column constraints. We shall construct a Latin square $X$ consistent with the original instance of LSC. Consider the restriction of $Y$ to the set $\mathcal{N}_{ij}$, for some $1 \leq i, j \leq n$. As before, let $l = \lambda(i, j)$ and $m = \mu(i, j)$. The restriction of $Y$ to the array of cells $[l, m] \times [l, m]$ forms a Latin square on ground-set $S(i, j)$ with the top left corner cell removed. The values $Y(i, l)$ and $Y(l, j)$ must both equal the missing element of this Latin square, and hence $Y(i, l) = Y(l, j) \in S(i, j)$. Now set $X(i, j)$ equal to $Y(i, l)$.

Since the individual cell constraints are clearly satisfied, it only remains to show that $X$ is indeed a Latin square, i.e., that no row or column of $X$ contains duplicate values. By symmetry, we need only check this condition for rows. But the $n$ values occurring in the $i$th row of $X$ are equal, by construction, to the $n$ corresponding values occurring in the $i$th row of $Y$. Hence the $i$th row of $X$ cannot contain duplicate values. We have thus demonstrated that the LSC instance admits a solution if the PLSC instance does. This completes the validation of the reduction. □

COROLLARY 4.2. 3DCT *is* NP*-complete, even in the special case where all the row, column and file sums are* 0 *or* 1.

**5. NP-completeness of identifying compromised cells.** We now consider the question of identifying the compromised cells in the general three-dimensional problem. In other words, if we are given a particular legal solution to an instance of 3DSDLS, and we focus attention on a particular (suppressed) cell, we wish to establish whether there exists a second solution in which that cell holds a different value. We shall show that determining whether a particular cell is compromised is also an NP-complete problem.

First, we give a formal description of the problem.

**5.1. Three-dimensional statistical data—compromised cells (3DSDCC).**

*Instance*: A positive integer $n$, a subset $S$ of $N^3$, where $N = \{1, \ldots, n\}$, nonnegative integer values $D(i, j, k)$ for each $(i, j, k) \in S$, and a particular triple $(i_0, j_0, k_0) \in S$.

*Question*: Does there exist a set of nonnegative integer values $D'(i, j, k)$ $((i, j, k) \in S)$ such that

(i) $\sum_{i=1}^{n} D'(i, j, k) = \sum_{i=1}^{n} D(i, j, k)$,

(ii) $\sum_{j=1}^{n} D'(i, j, k) = \sum_{j=1}^{n} D(i, j, k)$,

(iii) $\sum_{k=1}^{n} D'(i, j, k) = \sum_{k=1}^{n} D(i, j, k)$,

and

(iv) $D'(i_0, j_0, k_0) \neq D(i_0, j_0, k_0)$,

where all sums are taken over $(i, j, k) \in S$?

As was the case with our earlier problem, we prove 3DSDCC NP-complete by considering a special case that we can interpret in terms of Latin squares. Again, this is the special case in which all the row, column, and file sums are equal to 1. As earlier, we can interpret any solution to such an instance as a Latin square $X$ of size $n \times n$ in which $X(i, j) \in S(i, j)$, where $S(i, j) = \{k : (i, j, k) \in S\}$.

In fact, the NP-completeness of the Latin square nonuniqueness problem that we are about to describe has a slightly stronger consequence than we need, namely, that the problem of determining whether there is any legal solution, other than the one given, is NP-complete. The NP-completeness of 3DSDCC follows at once from this, for if we had a polynomial-time algorithm for the latter problem, we could apply it at most a polynomial number of times to determine whether there is any other legal solution.

**5.2. Latin square nonuniqueness (LSNU).**

*Instance*: A positive integer $n$, subsets $S(i, j)$ of $N = \{1, \ldots, n\}$ for each $i, j$, and a Latin square $X$ of size $n \times n$ with $X(i, j) \in S(i, j)$ for all $i, j$.

*Question*: Does there exist a Latin square $X'$ of size $n \times n$ such that $X'(i, j) \in S(i, j)$ for all $i, j$, and $X' \neq X$?

As observed above, the NP-completeness of 3DSDCC will follow from the NP-completeness of LSNU. Before proving this result, we need some further notation and a lemma.

For arbitrary values of $i, j$ $(1 \leq i, j \leq 2n)$, we write $p(i, j)$ for the value of $i + j - 1$ taken mod$n$ in $[1, n]$, so that, clearly, $p(i, j + n) = p(i, j)$.

LEMMA 5.1. *For fixed $i$, let*

$$S(j) = \begin{cases} \{p(i,j),\ p(i,j)+n\} & (1 \le j \le n), \\ \{p(i,j),\ p(i,j+1)+n\} & (n+1 \le j \le 2n). \end{cases}$$

*Then the sets $S(j)$ $(1 \le j \le 2n)$ have exactly two sets of distinct representatives, namely, $s(j)$ and $t(j)$ defined by*

$$s(j) = \begin{cases} p(i,j) & (1 \le j \le n), \\ p(i,j+1)+n & (n+1 \le j \le 2n) \end{cases}$$

*and*

$$t(j) = \begin{cases} p(i,j)+n & (1 \le j \le n), \\ p(i,j) & (n+1 \le j \le 2n). \end{cases}$$

*Proof.* First of all we observe the following intersections:

$$S(j) \cap S(j+n) = \{p(i,j)\} \quad (1 \le j \le n),$$
$$S(j+1) \cap S(j+n) = \{p(i,j+1)+n\} \quad (1 \le j \le n),$$

where $j+1$ is taken mod$n$ in $[1,n]$. So, if we choose $p(i,1)$ as the representative for $S(1)$, we are forced to choose $p(i,2)+n$ as the representative for $S(n+1)$, which in turn forces us to choose $p(i,2)$ as the representative for $S(2)$, and so on, leading to the system $s$ defined above. On the other hand, if we choose $p(i,1)+n$ as the representative for $S(1)$, we are forced to choose $p(i,2n)$ as the representative for $S(2n)$, which in turn forces us to choose $p(i,n)+n$ as the representative for $S(n)$, and so on, leading to the system $t$ defined above. These are the only two possibilities. □

The following further lemma is analogous to the previous one.

LEMMA 5.2. *For fixed $j$, let*

$$S(i) = \begin{cases} \{p(i,j),\ p(i,j)+n\} & (1 \le i \le n), \\ \{p(i,j),\ p(i+1,j)+n\} & (n+1 \le i \le 2n). \end{cases}$$

*Then the sets $S(i)$ $(1 \le i \le 2n)$ have exactly two sets of distinct representatives, namely, $s(i)$ and $t(i)$ defined by*

$$s(i) = \begin{cases} p(i,j) & (1 \le i \le n), \\ p(i+1,j)+n & (n+1 \le i \le 2n) \end{cases}$$

*and*

$$t(i) = \begin{cases} p(i,j)+n & (1 \le i \le n), \\ p(i,j) & (n+1 \le i \le 2n). \end{cases}$$

THEOREM 5.3. *Latin square nonuniqueness is NP-complete.*

*Proof.* Membership in NP is immediate. We need merely guess values $X'(i, j) \in S(i, j)$ for each $(i, j)$, and easily verify in polynomial time that $X'$ is a Latin square, and that $X' \neq X$.

To prove NP-completeness, we describe a polynomial-time transformation from LSC, known to be NP-complete by Theorem 3.3, to LSNU.

Given an instance of LSC of size $n$ with ground-set $\{1, \ldots, n\}$ and subsets $S(i, j)$ $(1 \leq i, j \leq n)$, we construct from it an instance of LSNU of size $2n$, with ground-set $\{1, \ldots, 2n\}$ and subsets $S'(i, j)$ $(1 \leq i, j \leq 2n)$ as follows.

We set

$$S'(i, j) = \begin{cases} S(i, j) \cup \{p(i, j) + n\} & \text{for } 1 \leq i, j \leq n, \\ \{p(i, j), p(i, j) + n\} & \text{for } n + 1 \leq i \leq 2n, 1 \leq j \leq n, \\ & \text{and } 1 \leq i \leq n, n + 1 \leq j \leq 2n, \\ \{p(i, j), p(i - 1, j) + n\} & \text{for } n + 1 \leq i, j \leq 2n. \end{cases}$$

The Latin square for this instance of LSNU is defined by

$$X(i, j) = \begin{cases} p(i, j) + n & \text{for } 1 \leq i, j \leq n, \\ p(i, j) & \text{for } n + 1 \leq i \leq 2n, 1 \leq j \leq n, \\ & \text{and } 1 \leq i \leq n, n + 1 \leq j \leq 2n, \\ p(i - 1, j) + n & \text{for } n + 1 \leq i, j \leq 2n. \end{cases}$$

Verification that $X$ is a Latin square and that $X(i, j) \in S(i, j)$ for all $i, j$ is straightforward.

We now show that there is a Latin square $X'$ with $X'(i, j) \in S(i, j)$ for all $i, j$, and $X'$ nonidentical to $X$ if and only if the original instance of LSC is solvable, and indeed if this is the case then $X'(i, j) \neq X(i, j)$ for all $i, j$.

First of all, if the LSC instance is solvable, with $n \times n$ Latin square $Y$, $Y(i, j) \in S(i, j)$ $(1 \leq i, j \leq n)$, we may choose

$$X'(i, j) = \begin{cases} Y(i, j) & \text{for } 1 \leq i, j \leq n, \\ p(i, j) + n & \text{for } n + 1 \leq i \leq 2n, 1 \leq j \leq n, \\ & \text{and } 1 \leq i \leq n, n + 1 \leq j \leq 2n, \\ p(i, j) & \text{for } n + 1 \leq i, j \leq 2n. \end{cases}$$

to give a $2n \times 2n$ Latin square $X'$, with $X'(i, j) \neq X(i, j)$ for all $i, j$.

On the other hand, let us consider the circumstances under which a different $2n \times 2n$ Latin square may exist. If for some $(i_0, j_0)$, $X'(i_0, j_0) \neq X(i_0, j_0)$, then it is immediate that $X'(i', j') \neq X(i', j')$ for some $(i', j')$ with $1 \leq i' \leq n, n + 1 \leq j' \leq 2n$. By Lemma 5.1, it follows that $X'(i, j) \neq X(i, j)$ for all $i, j$ $(n + 1 \leq i \leq 2n, 1 \leq j \leq 2n)$. A further application of Lemma 5.2 reveals that this is true also for $1 \leq i \leq n, n + 1 \leq j \leq 2n$. Hence, in order that $X'$ may be a Latin square, every value $X'(i, j)$ $(1 \leq i, j \leq n)$ must belong to $S(i, j)$, and therefore these values constitute a Latin square for the original instance of LSC. □

Finally, we consider the special case in which all cells are suppressed. This version of the problem can be expressed as described in the following section.

**5.3. Contingency table nonuniqueness (CTNU).**

*Instance*: A positive integer $n$ and an $n \times n \times n$ table $X$ of nonnegative integers.

*Question*: Does there exist an $n \times n \times n$ table $X'(\neq X)$ such that $X'$ and $X$ have identical row, column, and file sums?

Superficially, this version of the problem might appear easier than the general 3DS-DCC problem. For example, the existence of nonzero entries in positions $(i, j, k')$, $(i, j', k)$, $(i', j, k)$, and $(i', j', k')$ for some $i, i', j, j', k, k'$ would constitute an obvious sufficient condition for a "yes" answer—these entries could be reduced by 1 and the entries in positions $(i, j, k)$, $(i, j', k')$, $(i', j, k')$, and $(i', j', k)$ increased by 1 to give a second legal solution.

However, contingency table nonuniqueness is still NP-complete, as we now show. We first of all observe, in the same spirit as previously, that the special case of CTNU in which all row, column and file sums are 0 or 1 can be phrased as a partial Latin square problem, as follows.

**5.4. Partial Latin square nonuniqueness (PLSNU).**

*Instance*: A positive integer $n$, subsets $R(i)$ and $C(j)$ of $N = \{1, \ldots, n\}$ for each $i, j$, a subset $\mathcal{N}$ of $N^2$ such that (a) $|R(i)| = |k : (i, k) \in \mathcal{N}|$, and (b) $|C(j)| = |k : (k, j) \in \mathcal{N}|$, and values $X(i, j)$ for all $(i, j) \in \mathcal{N}$ satisfying

(i) $X(i, j) \in R(i) \cap C(j)$;

(ii) $i \neq i' \Rightarrow X(i, j) \neq X(i', j)$;

(iii) $j \neq j' \Rightarrow X(i, j) \neq X(i, j')$.

*Question*: Do there exist values $X'(i, j)$ for all $(i, j) \in \mathcal{N}$ satisfying (i), (ii), and (iii) above, such that $X'(i_0, j_0) \neq X(i_0, j_0)$ for some $i_0, j_0$?

THEOREM 5.4. CTNU *is* NP-*complete.*

*Proof.* Membership in NP is obvious, since we can guess $X'$ and easily verify the required conditions in polynomial time.

The key to the proof of NP-completeness is the observation that the polynomial-time transformation from LSC to PLSC given in the proof of Theorem 4.1 can, with relatively minor adjustments, be made parsimonious, i.e., so that there is a one-to-one correspondence between solutions to the original LSC instance and solutions to the derived PLSC instance. Hence, this parsimonious version gives us a polynomial-time transformation from LSNU to PLSNU, and in view of Theorem 5.3, establishes the NP-completeness of the latter problem. It remains to describe the details of this parsimonious transformation.

Define $n'$, $\lambda$, and $\mu$ as in the proof of Theorem 4.1; recall that $n'$ denotes the size of the derived instance of PLSC. As before, the set $\mathcal{N}$ of free cells is constructed as a disjoint union of component sets $\mathcal{N}_{ij}$. For each $i, j$ in the range $1 \leq i, j \leq n$ define

$$\mathcal{N}_{ij} = \{(k, l), (k, k), (l, k) : l + 1 \leq k \leq m\} \cup \{(i, l), (l, j)\},$$

where $l = \lambda(i, j)$ and $m = \mu(i, j)$. (Note that at this point the construction diverges somewhat from that employed in the proof of Theorem 4.1.) As before we define $\mathcal{N} = \bigcup_{ij} \mathcal{N}_{ij}$.

We now specify the row and column sets. For $k$ in the range $1 \leq k \leq n$, set $R(k) = C(k) = N$. For $k$ outside this range, i.e., for $n + 1 \leq k \leq n'$, let $i, j \in N$ be the unique integers satisfying $\lambda(i, j) \leq k \leq \mu(i, j)$. Let $l = \lambda(i, j)$, $m = \mu(i, j)$, $s = |S(i, j)|$, and let $\{v_1, \ldots, v_s\}$ be an enumeration of the elements of the set $S(i, j)$. Then define

$$R(k) = C(k) = \begin{cases} S(i, j), & \text{if } k = l; \\ \{v_{k-l}, v_{k-l+1}\}, & \text{otherwise.} \end{cases}$$

It can be checked that $R$, $C$, and $\mathcal{N}$ together form a consistent instance of PLSC. This completes the construction. Figure 4 is intended to illustrate a fragment of the derived instance of

PLSC, and features the rows and columns that lie within the range $[l, m]$. (The conventions employed here are the same as those of Fig. 2.) For this example, we have again taken $S(i, j) = \{1, 2, 3, 4\}$. The intention is that this $4 \times 4$ square should replace the $4 \times 4$ square at the bottom-right corner of Fig. 3.

| | {1, 2, 3, 4} | {1, 2} | {2, 3} | {3, 4} |
|---|---|---|---|---|
| {1, 2, 3, 4} | — | $\underline{1}, 2$ | $\underline{2}, 3$ | $3, \underline{4}$ |
| {1, 2} | $\underline{1}, 2$ | $1, \underline{2}$ | — | — |
| {2, 3} | $\underline{2}, 3$ | — | $2, \underline{3}$ | — |
| {3, 4} | $3, \underline{4}$ | — | — | $\underline{3}, 4$ |

FIG. 4. *A fragment of the derived instance of* PLSC.

The verification of the reduction relies on the following observation. Let $Y$ be any partial Latin square that is consistent with the derived instance of PLSC. Consider the restriction of $Y$ to the set $\mathcal{N}_{ij}$. Recall that $\{v_1, \ldots, v_s\}$ is an enumeration of the set $S(i, j)$, and suppose that $Y(i, l) = v_t$. Observe that column $l$ of $Y$ is completely forced: from cell $(l+1, l)$ to cell $(m, l)$ the entries must read $v_1, v_2, \ldots, v_{t-1}, v_{t+1}, v_{t+2}, \ldots, v_s$; these entries constrain the diagonal, which from cell $(l+1, l+1)$ to cell $(m, m)$ must read $v_2, v_3, \ldots, v_t, v_t, v_{t+1}, \ldots, v_{s-1}$; these in turn constrain row $l$, which from cell $(l, l+1)$ to $(l, m)$ must read $v_1, v_2, \ldots, v_{t-1}, v_{t+1}, v_{t+2}, \ldots, v_s$. Figure 4 illustrates the pattern that emerges when $v_1 = 1$, $v_2 = 2$, $v_3 = 3$, $v_4 = 4$, and $t = 3$.

It follows from this chain of reasoning that $Y(l, j) = v_t = Y(i, l)$. Moreover, once $Y(i, j)$ has been chosen, there is precisely one way to extend $Y$ to the remaining cells in $\mathcal{N}_{ij}$. Armed with this observation, the validation of the reduction proceeds as in the proof of Theorem 4.1; the reduction is clearly parsimonious. □

**6. The case of "slice" sums.** In an alternative extension to three dimensions of the original two-dimensional problem, suppose that we are given the sums of all two-dimensional "slices" of a three-dimensional table rather than the sums of the one-dimensional rows, columns, and files. In practice, for example, we might be given the row and column sums for three two-dimensional tables relating each pair of three distinct attributes of a population. Such a population could be represented in a three-dimensional table relating all three attributes, and the row and column sums for the two-dimensional tables would translate into the sums of the two-dimensional slices in the three-dimensional table.

The question arises as to the status, in this context, of problems corresponding to those investigated in earlier sections when row, column, and file sums are given. We conclude by summarizing the main results that hold for this version of the problem. They are again NP-completeness results, with the notable exception of the contingency table problem.

Consider an $n \times n \times n$ table $D$ of nonnegative integer values $D(i, j, k)$, and define the slice sums

$$X(i) = \sum_{j,k} D(i, j, k),$$

$$Y(j) = \sum_{i,k} D(i, j, k),$$

$$Z(k) = \sum_{i,j} D(i, j, k).$$

The following theorem and corollary are analogous to the results established by Gusfield in [3, §5] and can be proved by similar methods, with a simple induction argument replacing the use of network flow.

THEOREM 6.1. *Suppose that, for each $i$, $j$, $k$, we are given nonnegative integer values $X(i)$, $Y(j)$, $Z(k)$ such that $\sum_i X(i) = \sum_j Y(j) = \sum_k Z(k) = T$. Then there exists a table $D$ for which $X(i)$, $Y(j)$, and $Z(k)$ are the slice sums. Furthermore, the tightest upper and lower bounds on the value of $D(i, j, k)$ are* $\min(X(i), Y(j), Z(k))$ *and* $\max(0, X(i)+Y(j)+Z(k)-2T)$, *respectively.*

COROLLARY 6.2. *In the context of the previous theorem, only the cells in the slice with the largest sum can have a nonzero lower bound, and this can happen only if that slice has sum $> 2T/3$.*

The results of the above theorem contrast with the NP-completeness of the corresponding problem when row, column, and file sums are given (3DCT) and perhaps give us some hope that, when the values of some cells are revealed as well as the slice sums, there may be polynomial-time algorithms to determine a legal solution and to identify compromised cells. However, this turns out not to be so.

As far as finding a legal solution is concerned, it is not hard to see that the special case in which all slice sums over suppressed cells are equal to 1 is just the three-dimensional matching problem, which is well known to be NP-complete [2].

Finally, the problem of identifying compromised cells in this context can be proved NP-complete by a reduction and argument similar to, but easier than, that used in the proof of Theorem 5.3.

## REFERENCES

[1] D. DENNING, *Cryptography and Data Security*, Addison-Wesley, Reading, MA, 1982.
[2] M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability*, Freeman, San Francisco, 1979.
[3] D. GUSFIELD, *A graph theoretic approach to statistical data security*, SIAM J. Comput., 17 (1988), pp. 552–571.
[4] ———, *Faster dection of compromised data in 2-D tables*, Tech. Report CSE-89-30, Computer Science Division, University of California, Davis, CA, 1989.
[5] M. HALL, JR., *Combinatorial Theory*, Blaisdell, Waltham, MA, 1967.
[6] M.-Y. KAO AND D. GUSFIELD, *Efficient detection and protection of information in cross tabulated tables: Linear invariant test*, SIAM J. Disc. Math., 6 (1993), pp. 460–476.
[7] C. L. LIU, *Introduction to Combinatorial Mathematics*, McGraw-Hill, New York, 1968.