TRANSACTIONS ON DATA PRIVACY 13 (2020) 67–90

Solving the Disclosure Auditing Problem for Secondary Cell Suppression by Means of Linear Programming

Ton de Waal[*,**], **Wieger Coutinho**[***]

[*]Statistics Netherlands, PO Box 24500, 2490 HA, The Hague, The Netherlands.

[**]Tilburg University, PO Box 90153, 5000 LE, Tilburg, The Netherlands.

[***]Dunea, Plein van de Verenigde Naties 11-15, 2719 EG, Zoetermeer, The Netherlands.

E-mail: t.dewaal@cbs.nl

Received 28 March 2019; received in revised form 21 October 2019; accepted 8 April 2020

Abstract. National Statistical Institutes (NSIs) have the obligation to protect the privacy of individual persons or enterprises against disclosure of potentially sensitive information. For this reason, NSIs protect tabular data against disclosure of sensitive information before they are released. For tabular magnitude data, the starting point of this protection process usually is a sensitivity measure for individual cells. Such a sensitivity measure defines when a cell value is considered safe for publication or not. An often used method to protect a table with unsafe cells against disclosure of sensitive information is cell suppression. [5] argues that the standard criterion for deciding whether a table after suppression is safe or not is somewhat inconsistent and proposes a new criterion. [5] also gives a mixed-integer programming problem formulation for applying this new criterion. The problem with that formulation is that it is quite large and very hard to solve for even moderately sized tables. To be more precise, that mixed-integer programming problem formulation suggests that the auditing problem based on the criterion of [5] is NP-hard. The general assumption among operations research experts is that the computing time for NP-hard problems is non-polynomial in their input parameters. In the current paper, we propose solving a number of smaller and computationally much easier linear programming problems instead of solving one large mixed-integer programming problem. Solving linear programming problems can be done in time polynomial in their input parameters.

Keywords. Cell Suppression, Mathematical programming problems, Sensitivity measures, Statistical disclosure control, Tabular magnitude data

1 Introduction

National Statistical Institutes (NSIs) have the task to provide policy makers, researchers and the general public with accurate and detailed information on the current state of society and its development over time. Such information can, for instance, be published in the form of statistical tables, which is in fact the traditional and still most important way to publish statistical information. While NSIs on the one hand have the obligation to provide detailed statistical information, they also have the obligation to protect the privacy of individual persons and enterprises against disclosure of potentially sensitive information.

Such disclosure of sensitive information on individual persons or enterprises may occur when microdata are released, for instance when an attacker manages to link a data record in the released microdata to the right person or enterprise in the population. Disclosure of sensitive individual information may also occur when tabular data are released. A simple example is a table containing the turnover of enterprises cross-classified by branch of industry and region. If there is only one enterprise in the population with a certain combination of branch of industry and region, the turnover of this enterprise can immediately be disclosed from the published cell value.

For this reason, NSIs protect data against disclosure of sensitive information before these data are released. This protection process is referred to as statistical disclosure control (SDC) or statistical disclosure limitation. For a general introduction to SDC, see, for example, [14], [29], and [30].

Tabular data can be subdivided into frequency tables, where (estimated) counts are published, and magnitude tables, where (estimated) totals of a variable, e.g. turnover of enterprises cross-classified by branch of industry and region, are published. In this paper we focus on SDC for tabular magnitude data.

For tabular magnitude data, the starting point for SDC usually is a sensitivity measure for individual cells in the table to be protected (see, e.g., [1] and [2]). Such a sensitivity measure defines when a cell value is considered safe for dissemination or not. Safe cell values may in principle be published, whereas unsafe (or sensitive) cells must be protected. When all cell values in a table are considered safe, the table itself is considered safe. If one or more cell values are unsafe, the table is considered unsafe and needs protection.

An often used method to protect an unsafe table against disclosure of sensitive information is cell suppression. When cell suppression is applied, the values of one or more cells are replaced by some symbol, e.g. a cross ($\times$). When cell suppression is applied, the first step is to suppress the values of the unsafe cells. This is called primary cell suppression. Besides suppression of the unsafe cells, it is usually also necessary to suppress the values of some safe cells, as otherwise the values of (some of) the unsafe cells may be re-calculated from the marginal totals and the values of the non-suppressed cells. This is referred to as secondary suppression.

Secondary cell suppressions are found by solving a problem of the kind: find the "best" set of secondary suppressions such that the "table with suppressed cells is safe". This problem is referred to as the *secondary cell suppression problem*. It consists of two important aspects: find the "best" suppressions and determine whether a table with suppressions is safe. "Best" is defined by means of some target function, which needs to be minimized. Examples of such target functions are: the total suppressed value, the number of suppressed cell values, and the number of contributions to the suppressed cells. The problem of determining whether a table after suppression is safe is called *the auditing problem*. The auditing problem is a fundamental part of the secondary cell suppression problem.

The secondary cell suppression problem has been described extensively in the literature, see, for instance, [3], [4], [6], [8], [9], [10], [14], [17], [24], [29] and [30]. Essential in all these descriptions is the same criterion for deciding whether a table is safe after cell suppression. This criterion is the de facto standard criterion for determining whether a table with cell suppressions is safe.

[5] argues that this standard criterion for deciding whether a table after cell suppression is safe is inconsistent with the sensitivity measure for individual cells and proposes an alternative criterion that is based on applying the sensitivity measure for single cells to aggregations of suppressed cells. This criterion is an extension of ideas previously described by [4], [11], [23], [25], [26] and [27]. In [5] also a mixed-integer programming problem formulation

for solving the auditing problem is proposed, using this alternative criterion. Section 2 of this paper recapitulates the standard criterion, the arguments given by [5] against it, and the alternative criterion proposed by [5].

The safeness of a table with cell suppressions (or an individual cell) is always dependent on which criterion one uses. We hope that in the rest of the paper it is clear from the context which (kind of) criterion is used to decide whether a table with cell suppressions (or individual cell) is safe.

The aim of the current paper is two-fold. First, we show that, in any case in a theoretical sense, our proposed approach for the auditing problem is more efficient than the approach of [5] that was based on a mixed-integer problem formulation. Second, we show how our proposed approach for the auditing problem can, in principle, be extended to solving the cell suppression problem.

The fundamental issue with the mixed-integer programming problem formulation for the auditing problem given by [5] is that it is quite large and hard to solve for even moderately sized tables. To be more precise, that mixed-integer programming problem suggests that the auditing problem based on the criterion of [5] is NP-hard. The general assumption among operations research experts is that the computing time for NP-hard problems is non-polynomial in their input parameters. In the current paper, we propose solving a number of smaller and computationally much easier linear programming (LP) problems instead of solving one large mixed-integer programming problem. Solving linear programming problems can be done in time polynomial in their input parameters. The proposed approach based on solving a number of linear programming problems is discussed in Section 3.

Using the approach for the auditing problem described in Section 3, one can develop an approach for solving the secondary cell suppression problem, in any case to suboptimality. A simple example of such an approach is given in Section 4. This approach is based on combining the new approach for the auditing problem with the so-called hypercube method (see, e.g., [13], [21] and [22]).

Section 5 gives some test results for our approaches for the auditing problem as well as the cell suppression problem to show the feasibility of these approaches in practice. For this, we developed simple prototype software. We do not claim any superiority of our prototype software over any available software for the auditing problem or the cell suppression problem, nor over any software based on the mixed-integer programming problem formulation in [5].

Finally, Section 6 concludes the paper with a brief discussion.

2 Cell suppression

2.1 Prior/posterior rule

As mentioned in the Introduction, cell suppression consists of two steps: primary suppression of unsafe cells and secondary suppression of some additional cell values. Which cells are unsafe is determined by a sensitivity measure. Well-known classes of sensitivity measures are the prior/posterior rule and the dominance rule. The prior/posterior rule considers a cell as unsafe if one of the contributions can be re-calculated to within a certain threshold percentage of its value using the total cell value and assumed background information. The threshold percentage and the assumed background information are determined by the parameters of the prior/posterior rule. The dominance rule considers a

cell as unsafe if a substantial part of its value is due to only a few contributors. The parameters of the dominance rule determine what is meant by "a substantial part" and "a few contributors".

In this paper we will focus on the prior/posterior rule. The prior/posterior rule uses two parameters p and q where $p < q$. It is assumed that, prior to the publication of the table, everyone can estimate the contribution of each contributor to the table to within q percent. It is also assumed that an attacker knows who contributed to the table, and to which cell each contributor to the table contributes. A cell is considered unsafe if a non-zero contribution of an individual contributor to that cell can be estimated by someone else to within p percent after publication of the table. The prior/posterior rule can be formulated in a version that is more easily applicable in practice. The formulation given below is based on Section 5 in [5] and can be applied to tables with negative contributions. Note that since we assume that a priori each contribution can be estimated to within q percent, it is a priori also known whether this contribution is positive or negative, assuming that $q \leq 100$ which is generally the case. In this paper we indeed assume that $q \leq 100$.

We denote the number of contributors to a table by R and the number of suppressed cell values by S_C. A suppressed cell value is denoted by x_i $(i = 1, \ldots, S_C)$ and the contribution of contributor r to x_i by x_i^r $(i = 1, \ldots, S_C; r = 1, \ldots, R)$. In this paper we assume that each contributor contributes to one cell only, i.e. there are no holdings contributing to several cells, and hence that x_i^r equals zero for all but one $i = 1, \ldots, S_C$. We also assume that there are no collusions of several contributors to a table, trying to disclose sensitive information of another contributor to the table. We have $x_i = \sum_{i=1}^{R} x_i^r$. We use the notation $x_i^{[r]}$ to denote decreasingly ordered absolute contributions to suppressed cell x_i $(i = 1, \ldots, S_C)$, i.e. $|x_i^{[1]}| \geq |x_i^{[2]}| \geq \ldots \geq |x_i^{[R]}| \geq 0$.

Let us suppose that a contributor s in a suppressed cell i wants to estimate the contribution x_i^t of contributor t to cell i. Contributor s can derive an upper bound for x_i^t by subtracting his own contribution x_i^s to suppressed cell i and lower bounds for the values of x_i^r $(r \neq s, t)$ from x_i. The lower bound for x_i^r from the perspective of contributor s is based on his prior knowledge (parameter q of the prior/posterior rule) and equals $(1 - \frac{q}{100})x_i^r$ if $x_i^r \geq 0$ and $(1 + \frac{q}{100})x_i^r$ if $x_i^r < 0$. The upper bound for x_i^t, $U_s(x_i^t)$, from the perspective of contributor s is hence given by

$$U_s(x_i^t) = x_i - x_i^s - (1 - \frac{q}{100}) \sum_{r: x_i^r \geq 0, r \neq s,t} x_i^r - (1 + \frac{q}{100}) \sum_{r: x_i^r < 0, r \neq s,t} x_i^r = x_i^t + \frac{q}{100} \sum_{r \neq s,t} |x_i^r|$$

In a similar way contributor s can derive that the lower bound on x_i^t from his perspective, $L_s(x_t^r)$, is given by

$$L_s(x_i^t) = x_i^t - \frac{q}{100} \sum_{r \neq s,t} |x_i^r|$$

Cell i is unsafe if and only if $U_s(x_i^t) < x_i^t + (\frac{p}{100})|x_i^t|$ (or equivalently $L_s(x_i^t) > x_i^t - (\frac{p}{100})|x_i^t|$) for some contributors s and t. So, cell i is unsafe if and only if $\frac{q}{100} \sum_{r \neq s,t} |x_i^r| < \frac{p}{100}|x_i^t|$ for all contributors s and t $(s \neq t)$. That is, cell i is safe if and only if

$$\frac{q}{100} \sum_{r \neq s,t} |x_i^r| \geq \frac{p}{100}|x_i^t| \tag{1}$$

for all contributors s and t $(s \neq t)$. Since $\frac{q}{100} \sum_{r \neq 1,2} |x_i^{[r]}| \leq \frac{q}{100} \sum_{r \neq s,t} |x_i^r|$ and $\frac{p}{100}|x_i^{[1]}| \geq \frac{p}{100}|x_i^t|$, (1) is satisfied for all contributors s and t $(s \neq t)$ if and only if

$$\frac{q}{100}\sum_{r\neq 1,2}|x_i^{[r]}| \geq \frac{p}{100}|x_i^{[1]}| \tag{2}$$

This extended formulation of the prior/posterior rule that allows for negative contributions reduces to the traditional prior/posterior rule that allows for nonnegative contributions only when all contributions are nonnegative.

We can re-express (2) as: cell i is unsafe if and only if

$$\frac{q}{100}\sum_{r=1}^{R}|x_i^{[r]}| - \frac{q}{100}|x_i^{[2]}| - \frac{q}{100}|x_i^{[1]}| < \frac{p}{100}|x_i^{[1]}|$$

i.e. if and only if

$$(p+q)|x_i^{[1]}| + q|x_i^{[2]}| - q\sum_{r=1}^{R}|x_i^{[r]}| > 0 \tag{3}$$

By defining the sensitivity function

$$S_{p,q}(x_i) = (p+q)|x_i^{[1]}| + q|x_i^{[2]}| - q\sum_{r=1}^{R}|x_i^{[r]}| \tag{4}$$

we can express (3) as: cell i is unsafe if and only if $S_{p,q}(x_i) > 0$. The sensitivity function (4) is subadditive (see [1] and [2] for a definition of subadditivity). In our case, this means that $S_{p,q}(\bigcup_{i\in I} x_i) \leq \sum_{i\in I} S_{p,q}(x_i)$ for any set of suppressed cells I, where $\bigcup_{i\in I} x_i$ denotes the artificial cell consisting of the union of all data corresponding to the cells x_i for $i = 1, \ldots, I$. That is, a combination of cells is at most as sensitive in terms of the sensitivity function $S_{p,q}$ as the sum of values of the sensitivity function for the individual cells. This implies that a combination of safe cells is also safe.

The so-called $p\%$-rule, extended to allow for negative contributions, is defined as the prior/posterior rule with parameters p and $q = 100$. In this paper, we will use the prior/posterior rule, although for computational convenience we will use $q = 100$ in the examples.

We note that the prior/posterior and the $p\%$-rule are in fact equivalent. Namely, a prior/posterior rule with parameters p and q is equivalent to a $p^*\%$-rule, with $p^* = \frac{p}{q}$ as one can see by dividing the left hand-side of (3) by q.

In the rest of Section 2 we will discuss the auditing problem.

2.2 The traditional criterion for the auditing problem

After determining and suppressing the unsafe cells, some of the suppressed cell values may be re-calculated from the remaining information in the table. Consider for example Table 1, which is taken from [5]. In this table all contributions are known to be nonnegative, and $x_{1,1}$ and $x_{2,1}$ are primary suppressions. Obviously, $x_{1,1}$ and $x_{2,1}$ must both have the value 100.

As already mentioned in the Introduction, in order to protect suppressed unsafe cells against recalculation, it is usually necessary to suppress additional cell values. In Table 2 some cells have been secondarily suppressed.

Table 1: A table with primary suppressions

	C_1	C_2 C_3	Total	
R_1	$x_{1,1}$	1	3	104
R_2	$x_{2,1}$	2	1	103
R_3	70	3	2	75
Total	270	6	6	282

Table 2: Primary and secondary suppressions

	C_1	C_2	C_3	Total
R_1	$x_{1,1}$	1	$x_{1,3}$	104
R_2	$x_{2,1}$	2	$x_{2,3}$	103
R_3	70	3	2	75
Total	270	6	6	282

In the traditional criterion for the auditing problem, which can only be applied to tables with nonnegative cell values, a central role is played by the so-called *suppression intervals* of the suppressed cell values. The suppression interval of a suppressed cell is the interval of all possible values that this cell could take for someone who knows nothing more than the published table and the nonnegativity of the cell values. The suppression interval can be calculated by solving simple LP problems. For instance, the minimum of $x_{1,1}$ can be found by solving the LP problem

$$\begin{gathered}\text{Minimize } x_{1,1} \text{ subject to} \\ x_{1,1} + x_{1,3} = 103 \\ x_{1,3} + x_{2,3} = 4 \\ x_{1,1} + x_{2,1} = 200 \\ x_{2,1} + x_{2,3} = 101 \\ x_{i,j} \geq 0\end{gathered}$$

This yields a minimum of 99 for $x_{1,1}$. Similarly, we find that the maximum for $x_{1,1}$ is 101. The suppression interval for $x_{1,1}$ is hence $[99, 101]$. Note that, unlike the (p, q)-rule, the traditional criterion for the auditing problem does not assume any prior knowledge on the values of the contributions, apart from nonnegativity of the cell values.

Traditionally, a table with suppressed cells is considered safe if the suppression interval for each sensitive cell is "sufficiently" wide. "Sufficiently" wide is usually operationalized by requiring that the upper bound on the suppression interval is at least equal to that value for which the cell would be safe according to sensitivity measure, i.e. the prior/posterior rule in our case, and similarly that the lower bound on the suppression interval is at most equal to that value for which the cell would be safe according to sensitivity measure. We will refer to this criterion as the traditional auditing criterion. In Sections 2.3 and 5.1 we will give some examples illustrating how "sufficiently" wide is operationalized.

2.3 Why do we need an alternative criterion for the auditing problem?

The traditional criterion for the auditing problem as described in Section 2.2 suffers from a flaw as pointed out by [5], namely that the traditional criterion for the auditing problem may be inconsistent with the applied sensitivity measure. We will give an example to illustrate the point.

Suppose that in Table 3 only cell $R_1 \times C_1$ is unsafe according to the used sensitivity measure, the prior/posterior rule with $p = 20$ and $q = 100$, and that the other cells are safe. Suppose furthermore that the largest two contributions to cell $R_1 \times C_1$ equal 155 and 4, respectively (and hence that the sum of all other contributions to cell $R_1 \times C_1$ equals 1), and that the largest contribution to cell $R_2 \times C_1$ equals 28.

Table 3: An unprotected table (example 1)

	C_1	C_2	Total
R_1	160	340	500
R_2	50	60	110
R_3	610	270	880
Total	820	670	1490

According to the traditional criterion for the auditing problem, the upper bound on the suppression interval of cell $R_1 \times C_1$ should be at least 190. Namely, when the upper bound on the suppression interval equals 190, the second largest contributor to $R_1 \times C_1$ can derive that the upper bound on the largest contribution to $R_1 \times C_1$ is $186 = 190 - 4 - 0$, with 4 being his own contribution to cell $R_1 \times C_1$ and 0 a lower bound on the other contributions to cell $R_1 \times C_1$ according to the used (p, q)-rule with $q = 100\%$. This upper bound exceeds the actual value of the largest contribution to cell $R_1 \times C_1$ (i.e. 155) by exactly 20%. Analogously, the lower bound on the suppression interval of $R_1 \times C_1$ should be at most 130, for then the second largest contributor to $R_1 \times C_1$ can derive a lower bound on the largest contribution to $R_1 \times C_1$ of $130 - 4 - 2 = 124$, with 4 being his own contribution and 2 an upper bound on the other contributions to cell $R_1 \times C_1$ according to the used (p, q)-rule.This lower bound is exactly 20% less than 155.

The suppression interval of cell $R_1 \times C_1$ in Table 4 is given by $[100, 210]$, which can be checked in the manner explained in Section 2.2. Since $210 > 190$ and $100 < 130$, Table 4 is considered safe according to the traditional criterion for the auditing problem.

Table 4: A safe table according to the traditional auditing criterion (example 1)

	C_1	C_2	Total
R_1	$\times$	$\times$	500
R_2	$\times$	$\times$	110
R_3	610	270	880
Total	820	670	1490

Now, consider Table 5, which has been obtained from Table 3 by recoding categories R_1 and R_2 into one category. Note that from Table 4 it is clear that values of cells $C_1 \times R_1$ and $C_1 \times R_2$ add up to $820 - 610 = 210$, and that the values of cells $C_2 \times R_1$ and $C_2 \times R_2$ add up to $670 - 270 = 400$. Table 4 contains a bit more information than Table 5 since Table 5 does not give the totals for categories R_1 and R_2 separately, only the total for R_1 and R_2 together.

Table 5: A recoded table (example 1)

	C_1	C_2	Total
$R_1\&R_2$	210	400	610
R_3	610	270	880
Total	820	670	1490

The inconsistency between the traditional criterion for the auditing problem and the sensitivity measure becomes clear from Table 5. Namely, cell $(R_1\&R_2) \times C_1$ in Table 5 is unsafe according to the used sensitivity measure, since the second largest contibutor (with contribution 28) can re-calculate the largest contribution (i.e. 155) to within 20%, as can easily checked. So, although Table 5 contains a bit less information than Table 4, Table 5 is considered unsafe whereas Table 4 is considered safe!

In the above example, cell suppression was basically equivalent to recoding the table. The same kind of inconsistency as noted can also occur when cell suppression is not equivalent to recoding the table. We illustrate this by a small extension of the above example, and consider Table 6.

Table 6: An unprotected table (example 2)

	C_1	C_2	C_3	Total
R_1	160 (155 / 4)	380 (80 / 50)	340 (90 / 50)	880
R_2	50 (28 /10)	80 (24 / 16)	60 (18 / 12)	190
R_3	610 (110 / 100)	800 (250 / 200)	270 (80 / 60)	1680
Total	820	1260	670	2750

In Table 6, we mention the two largest contributions per cell between brackets. We again assume that cell $R_1 \times C_1$ is unsafe, whereas the other cells are safe. Again, we also assume that largest two contributions to cell $R_1 \times C_1$ equal 155 and 4, respectively and that the largest contribution to cell $R_2 \times C_1$ equals 28. We again use a prior/posterior rule with parameters $p = 20$ and $q = 100$.

According to the traditional criterion for the auditing problem, Table 7 is considered safe for the same reason as for Table 4.

Indeed, τ-ARGUS (see, e.g. [7] and [12]) will give Table 7 as output, if we measure information loss due to suppression by the sum of the suppressed cell values.

Table 7: A safe table according to the traditional auditing criterion (example 2)

	C_1	C_2	C_3	Total
R_1	$\times$	380	$\times$	880
R_2	$\times$	80	$\times$	190
R_3	610	800	270	1680
Total	820	1260	670	2750

However, the largest contributor to cell $R_2 \times C_1$ can derive that the upper bound on the largest contribution to cell $R_1 \times C_1$ is $820 - 610 - 28 - 0 = 182$ (the first two terms are available from the table, the third term is his own contribution to cell $R_2 \times C_1$, and the fourth term is a lower bound on the other contributions to cells $R_1 \times C_1$ and $R_2 \times C_1$, which

can be computed using the q parameter). Since, 182 is within 20% of 155, the combination of cells $R_1 \times C_1$ and $R_2 \times C_1$ is considered unsafe according to the applied sensitivity measure.

This phenomenon where several suppressed cell values can be combined into an unsafe combined cell has been referred to as an "ad-hoc roll up" (see [28]).

2.4 The alternative criterion for the auditing problem

[5] notes that conceptually separate cells and aggregations of cells for which the total value is known are basically the same and should hence be treated the same way. Namely, in both cases, we have some underlying microdata from individual persons or enterprises summing up to a known total. From this point of view it is logical that separate cells and aggregations of cells should be subjected to the same sensitivity measure. [5] therefore considers a table safe if and only if all aggregations of suppressed cells are safe, using the same sensitivity measure (or a slightly extended version thereof) as for separate cells.

We will illustrate the concept of aggregations by means of an example. Let us suppose that we aim to protect Table 8 against disclosure. To each cell in this table, one or more contributors contribute. We suppose that cells $R_1 \times C_1$ and $R_5 \times C_3$ are the sensitive cells in this table.

Table 8: An unprotected table (example 3)

	C_1	C_2	C_3	C_4	C_5	Total
R_1	16	44	48	35	18	161
R_2	47	82	51	80	29	289
R_3	59	88	16	28	86	277
R_4	61	78	59	94	84	376
R_5	3	93	82	41	5	224
Total	186	385	256	278	222	1327

Let us suppose that Table 9 is suggested - usually by a software package such as τ-ARGUS - as a protected version of Table 8. We aim to check whether Table 9 is indeed safe.

Table 9: The "protected" version of Table 8 (example 3)

	C_1	C_2	C_3	C_4	C_5	Total
R_1	$x_{1,1}$	44	48	35	$x_{1,5}$	161
R_2	47	82	$x_{2,3}$	80	$x_{2,5}$	289
R_3	59	88	16	28	86	277
R_4	61	78	59	94	84	376
R_5	$x_{5,1}$	93	$x_{5,3}$	41	5	224
Total	186	385	256	278	222	1327

The values of some aggregations in Table 9 can be deduced immediately, simply by looking at the rows and columns of the table. Such aggregations are called *explicit aggregations*. The values of aggregations that are linear combinations of the explicit ones are known too. Such aggregations are called *implicit aggregations*.

An explicit aggregation always is a sum of suppressed cells in one dimension. That is, in a two-dimensional table an explicit aggregation is always a sum of suppressed cells in

a certain row or a certain column. The explicit aggregations of Table 9 together with their known values are:

$$x_{1,1} + x_{1,5} = 34 \tag{5}$$
$$x_{1,5} + x_{2,5} = 47 \tag{6}$$
$$x_{2,5} + x_{2,3} = 80$$
$$x_{2,3} + x_{5,3} = 133$$
$$x_{5,3} + x_{5,1} = 85$$
$$x_{5,1} + x_{1,1} = 19$$

An example of an implicit aggregation with its known value is

$$x_{1,1} - x_{2,5} = -13 \tag{7}$$

This implicit aggregation is obtained by subtracting the second explicit aggregation (6) from the first (5).

Any (explicit or implicit) aggregation for Table 9 can be written as

$$\begin{aligned}\mu_1(x_{1,1} + x_{1,5}) + \mu_2(x_{1,5} + x_{2,5}) + \mu_3(x_{2,5} + x_{2,3}) + \\ \mu_4(x_{2,3} + x_{5,3}) + \mu_5(x_{5,3} + x_{5,1}) + \mu_6(x_{5,1} + x_{1,1})\end{aligned} \tag{8}$$

where μ_1 to μ_6 are parameters that can, in principle, take any value between $-\infty$ and ∞. However, since

$$\begin{aligned}\alpha\mu_1(x_{1,1} + x_{1,5}) + \alpha\mu_2(x_{1,5} + x_{2,5}) + \alpha\mu_3(x_{2,5} + x_{2,3}) + \\ \alpha\mu_4(x_{2,3} + x_{5,3}) + \alpha\mu_5(x_{5,3} + x_{5,1}) + \alpha\mu_6(x_{5,1} + x_{1,1})\end{aligned}$$

is essentially the same aggregation as (8) for any $\alpha \neq 0$, we may rescale the μ_k so $-1 \leq \mu_k \leq 1$ (for $k = 1, \ldots, 6$). Note that (7) can be obtained from (8) by setting $\mu_1 = 1$, $\mu_2 = -1$, and $\mu_3 = \mu_4 = \mu_5 = \mu_6 = 0$.

The prior/posterior rule as presented in Section 2.1 needs to be reformulated slightly for an aggregation with a known total. We will write such an aggregation X_j as

$$X_j = \sum_{i=1}^{S_C} \lambda_i^j x_i \tag{9}$$

where λ_i^j $(i = 1, \ldots, S_C)$ are coefficients, the subscript i refers to a certain cell and the superscript j is used to identify the aggregation. For example, in the case of (7) we have $\lambda_{1,1}^1 = 1$ and $\lambda_{2,5}^1 = -1$ where we have numbered this aggregation as the first one and the superscript of the λ's therefore equals 1. In the general case (8) we have $\lambda_{1,1} = \mu_1 + \mu_6$, $\lambda_{1,5} = \mu_1 + \mu_2$, $\lambda_{2,5} = \mu_2 + \mu_3$, $\lambda_{2,3} = \mu_3 + \mu_4$, $\lambda_{5,3} = \mu_4 + \mu_5$ and $\lambda_{5,1} = \mu_5 + \mu_6$, where we did not number the aggregation because this is the general expression for an aggregation for Table 9 rather than for a specific aggregation. In general, the coefficients λ_i^j defining an aggregation given by (9) may differ from -1 or +1.

Note that an aggregation X_j is defined by the λ_i^j only, i.e. without its known total. However, in some cases we will also give the known total of such an aggregation.

The contribution of contributor r $(r = 1, \ldots, R)$ to aggregation X_j is given by $\sum_{i=1}^{S_C} \lambda_i^j x_i^r$. Since we need the absolute contributions in an aggregation (see also Section 2.1), we define two operators. The first operation is the Φ_{ABS}^r operator, which returns the *absolute contribution* of contributor r $(r = 1, \ldots, R)$ to aggregation X_j. It is defined by

$$\Phi_{ABS}^r(X_j) = \sum_{i=1}^{S_C} |\lambda_i^j x_i^r|$$

The Φ_{ABS}^r operator can also be applied to a single suppressed cell x_i $(i = 1, \ldots, S_C)$: $\Phi_{ABS}^r(x_i) = |x_i^r|$. Later in this paper we will also use the Φ_{TOT} operator. We need this operator for a single suppressed cell x_i $(i = 1, \ldots, S_C)$ only, and it is defined by

$$\Phi_{TOT}(x_i) = \sum_{r=1}^{R} |x_i^r|$$

$\Phi_{ABS}^r(X_j)$, $\Phi_{ABS}^r(x_i)$ and $\Phi_{TOT}(x_i)$ can be calculated from the data underlying the table for $(i = 1, \ldots, S_C)$ and all aggregations j.

In a similar way as in Section 2.1 we can now derive that aggregation X_j is safe if and only if

$$\frac{q}{100} \sum_{r \neq 1,2} \Phi_{ABS}^{[r]}(X_j) \geq \frac{p}{100} \Phi_{ABS}^{[1]}(X_j)$$

where we again use the superscript $[r]$ to denote decreasingly ordered absolute contributions to aggregation X_j, i.e. $\Phi_{ABS}^{[1]}(X_j) \geq \Phi_{ABS}^{[2]}(X_j) \geq \ldots \geq \Phi_{ABS}^{[R]}(X_j) \geq 0$. That is, aggregation X_j is safe if and only if $(p+q)\Phi_{ABS}^{[1]}(X_j) + q\Phi_{ABS}^{[2]}(X_j) - q\sum_{r=1}^{R} \Phi_{ABS}^{[r]}(X_j) \leq 0$.

Equivalently, aggregation X_j is unsafe if and only if

$$(p+q)\Phi_{ABS}^{[1]}(X_j) + q\Phi_{ABS}^{[2]}(X_j) - q\sum_{r=1}^{R} \Phi_{ABS}^{[r]}(X_j) > 0 \tag{10}$$

For details of the derivation of (10), we refer to [5]. As already noted in Section 2.1, the prior/posterior rule with parameters p and q is equivalent to a $p^*\%$-rule with $p^* = \frac{p}{q}$.

Similar to the sensitivity function (4) for individual cells, we can define the sensitivity function

$$S_{p,q}^{agg}(X_j) = (p+q)\Phi_{ABS}^{[1]}(X_j) + q\Phi_{ABS}^{[2]}(X_j) - q\sum_{r=1}^{R} \Phi_{ABS}^{[r]}(X_j) \tag{11}$$

for aggregations.

One way to look at sensitivity function (11) for an aggregation X_j is that all absolute contributions $\Phi_{ABS}^r(X_j) = \sum_{i=1}^{S_C} |\lambda_i^j x_i^r|$ $(r = 1, \ldots, R)$ are first combined into one ad-hoc cell and next these absolute contributions are seen as contributions of individual contributors to that cell. That is,

$$S_{p,q}^{agg}(X_j) = S_{p,q}(\{\Phi_{ABS}^1(X_j), \Phi_{ABS}^2(X_j), \ldots, \Phi_{ABS}^R(X_j)\})$$

where $\{\Phi_{ABS}^1(X_j), \Phi_{ABS}^2(X_j), \ldots, \Phi_{ABS}^R(X_j)\}$ is an ad-hoc cell with contributions $\Phi_{ABS}^1(X_j)$ to $\Phi_{ABS}^R(X_j)$.

3 A Linear Programming Formulation for the Auditing Problem

To check whether a table with cell suppressions is safe according to the criterion proposed by [5], that article proposes to determine the most sensitive aggregation of suppressed cells. If even the most sensitive aggregation is safe, the table is safe. Finding the most sensitive aggregation is not trivial, however. [5] formulates a complicated mixed-integer programming problem that needs to be solved in order to find the most sensitive aggregation of suppressed cells. For more information on that formulation we refer to Section 6 of [5].

Mixed-integer programming problems can be notoriously hard to solve. In particular, the computing time can become very large for large or even medium-sized problem instances. In principle, the computing time of a mixed-integer programming problem can be exponential in terms of its unknowns; in any case this is the general assumption amongst operations research experts.

In order to overcome this problem, we propose to solve a sequence of LP problems, where we - for each of these LP problems - fix the attacker and attacked contributor beforehand. These LP problems can generally be solved quite quickly. In essence, a mixed-integer programming problem is often also solved by cleverly solving a sequence of LP problems. Examples of such approaches include Branch-and-Bound algorithms, Branch-and-Cut algorithms and Branch-and-Price algorithms (see, e.g., [15], [16], [18], [19] and [31]). What makes our approach different is that for each LP problem we can fix the attacker and attacked contributor beforehand.

The underlying idea of our LP approach is that, for each sensitive cell, we will check how accurate any other contributor to the table can calculate the value of the largest (absolute) contribution to that sensitive cell. We only have to check this for the largest (absolute) contributions to the other suppressed cells and for the second largest (absolute) contribution to the sensitive cell itself. Each of these checks can be expressed as an LP problem. This kind of audit check is also available in τ-ARGUS, but only for contributors within the same suppressed cell, not for contributors within aggregations of suppressed cells.

We will illustrate the LP approach by means of Tables 8 and 9. We recall that Table 8 has two unsafe cells, denoted as $x_{1,1}$ and $x_{5,3}$ in Table 9. As already noted in Section 2.4, all linear aggregations for Table 9 can be written as (8) with $-1 \leq \mu_k \leq 1$ (for $k = 1, \ldots, 6$). For $x_{1,1}$, we then check whether, in a certain (explicit or implicit) aggregation, the contributors with the largest (absolute) contributions to cells $x_{1,5}$, $x_{2,5}$, $x_{2,3}$, $x_{5,3}$, respectively $x_{5,1}$ can re-calculate the largest (absolute) contribution to $x_{1,1}$ too accurately in any aggregation. We also check whether the contributor with the second largest (absolute) contribution to $x_{1,1}$ can re-calculate the largest (absolute) contribution to $x_{1,1}$ too accurately in any aggregation.

Say, we first check whether the largest (absolute) contribution to cell $x_{1,5}$ can re-calculate the largest (absolute) contribution to cell $x_{1,1}$ too accurately. This largest (absolute) contribution to cell $x_{1,5}$ can, in principle, have 2 different signs for the corresponding λ_i^j in (9), namely a minus sign or a plus sign, in an aggregation involving $x_{1,1}$. For each of the two possible signs, we perform a check. We perform similar checks for the largest (absolute) contributons to $x_{2,5}$, $x_{2,3}$, $x_{5,3}$ and $x_{5,1}$, respectively. For the second largest (absolute) contribution to $x_{1,1}$ we have to perform only one check, because the second largest (absolute) contribution to $x_{1,1}$ can only have the same sign for the corresponding λ_i^j in (9) as the largest (absolute) contribution to $x_{1,1}$ in any aggregation.

Similarly, for $x_{5,3}$, we check whether the contributors with the largest (absolute) contributions to cells $x_{5,1}$, $x_{1,1}$, $x_{1,5}$, $x_{2,5}$, respectively $x_{2,3}$, and the second largest (absolute) contri-

bution to cell $x_{5,3}$, can re-calculate the largest (absolute) contribution to $x_{5,3}$ too accurately.

We have to check $22 = 2 \times (5 \times 2 + 1)$ relatively small LP problems instead of one large mixed-integer programming problem. Here the first 2 refers to the number of sensitive cells, the 5 to the number of other suppressed cells besides the unsafe cell under consideration, the second 2 to the possible signs in an aggregation of an attacker from another cell than the sensitive cell under consideration, and the 1 to the second largest (absolute) contribution in the unsafe cell under consideration.

We will describe two of those 22 LP problems, namely the LP problems to check whether the contributor with the largest (absolute) contribution to cell $R_1 \times C_5$ can re-calculate the largest (absolute) contribution to cell $R_1 \times C_1$ too accurately. The contribution of cell $R_1 \times C_1$ to an aggregation given by (8) is $(\mu_6 + \mu_1)x_{1,1}$. Similarly, we can find the contribution of the other suppressed cells to aggregation (8).

According to criterion (10), we should look at the absolute contributions to an aggregation given by (8). In order to find the absolute contributions to an aggregation we introduce variables $y^+_{i,j}$ and $y^-_{i,j}$ for the suppressed cells and some constraints for these variables. For instance, for $y^+_{1,1}$ and $y^-_{1,1}$ we demand that

$$y^+_{1,1} \geq (\mu_6 + \mu_1)\Phi_{TOT}(x_{1,1})$$

and

$$y^-_{1,1} \geq -(\mu_6 + \mu_1)\Phi_{TOT}(x_{1,1})$$

When $y^+_{1,1} + y^-_{1,1}$ is minimized, this sum will become equal to $|(\mu_6 + \mu_1)\Phi_{TOT}(x_{1,1})| = \sum_{r=1}^{R} |\mu_6 + \mu_1||x^r_{1,1}|$, i.e. the sum of the absolute contributions to cell $R_1 \times C_1$ in aggregation (8).

The LP problem assuming that the largest (absolute) contribution to $x_{1,1}$ is attacked by the largest (absolute) contribution to $x_{1,5}$ is given in (12) to (30) below.

$$\text{Maximize } (p+q)S + qA - qT \tag{12}$$

subject to

$$T = y^+_{1,1} + y^+_{1,5} + y^+_{2,5} + y^+_{2,3} + y^+_{5,3} + y^+_{5,1} + y^-_{1,1} + y^-_{1,5} + y^-_{2,5} + y^-_{2,3} + y^-_{5,3} + y^-_{5,1} \tag{13}$$

$$y^+_{1,1} \geq (\mu_6 + \mu_1)\Phi_{TOT}(x_{1,1}) \tag{14}$$

$$y^+_{1,5} \geq (\mu_1 + \mu_2)\Phi_{TOT}(x_{1,5}) \tag{15}$$

$$y^+_{2,5} \geq (\mu_2 + \mu_3)\Phi_{TOT}(x_{2,5}) \tag{16}$$

$$y^+_{2,3} \geq (\mu_3 + \mu_4)\Phi_{TOT}(x_{2,3}) \tag{17}$$

$$y^+_{5,3} \geq (\mu_4 + \mu_5)\Phi_{TOT}(x_{5,3}) \tag{18}$$

$$y^+_{5,1} \geq (\mu_5 + \mu_6)\Phi_{TOT}(x_{5,1}) \tag{19}$$

$$y_{11}^- \geq -(\mu_6 + \mu_1)\Phi_{TOT}(x_{1,1}) \tag{20}$$

$$y_{15}^- \geq -(\mu_1 + \mu_2)\Phi_{TOT}(x_{1,5}) \tag{21}$$

$$y_{25}^- \geq -(\mu_2 + \mu_3)\Phi_{TOT}(x_{2,5}) \tag{22}$$

$$y_{23}^- \geq -(\mu_3 + \mu_4)\Phi_{TOT}(x_{2,3}) \tag{23}$$

$$y_{53}^- \geq -(\mu_4 + \mu_5)\Phi_{TOT}(x_{5,3}) \tag{24}$$

$$y_{51}^- \geq -(\mu_5 + \mu_6)\Phi_{TOT}(x_{5,1}) \tag{25}$$

$$y_{1,1}^+ \geq 0, y_{1,5}^+ \geq 0, y_{2,5}^+ \geq 0, y_{2,3}^+ \geq 0, y_{5,3}^+ \geq 0, y_{5,1}^+ \geq 0 \tag{26}$$

$$y_{1,1}^- \geq 0, y_{1,5}^- \geq 0, y_{2,5}^- \geq 0, y_{2,3}^- \geq 0, y_{5,3}^- \geq 0, y_{5,1}^- \geq 0 \tag{27}$$

$$-1 \leq \mu_k \leq 1 \text{ (for } k = 1, \ldots, 6) \tag{28}$$

$$S \leq (\mu_6 + \mu_1)|x_{1,1}^{\max}| \tag{29}$$

$$A \leq (\mu_1 + \mu_2)|x_{1,5}^{\max}| \tag{30}$$

Here $x_{1,1}^{\max}$ and $x_{1,5}^{\max}$ are the maximum contributions to cells $R_1 \times C_1$ and $R_5 \times C_3$, respectively. The μ_k $(k = 1, \ldots, 6)$ refer to the (explicit and implicit) aggregations (see (8)). The unknowns in the above LP problem are the μ_k $(k = 1, \ldots, 6)$, S, A, T and the $y_{i,j}^+$ and $y_{i,j}^-$.

Note that, when we maximize (12), T is in fact

$$T = |y_{1,1}| + |y_{1,5}| + |y_{2,5}| + |y_{2,3}| + |y_{5,3}| + |y_{5,1}| = \sum_{r=1}^{R} \Phi_{ABS}^{[r]}(X^*)$$

where aggregation X^* is the optimal aggregation for the above LP problem, since maximizing (12) will make T as small as possible (and A and S as large as possible).

The upper and lower bounds on the μ_k $(k = 1, \ldots, 6)$ given by (28) ensure that the outcome of (12) is finite. By maximizing $(p + q)S + qA - qT$ (see (12)) we try to construct an unsafe aggregation. This is the case when the optimal value of $(p + q)S + qA - qT$ is larger than zero (see (10)). As already explained, equations (13) to (27) are needed to ensure that $T = \sum_{r=1}^{R} \Phi_{ABS}^{[r]}(X^*)$. Equation (29) ensures that $S = |(\mu_6 + \mu_1)\Phi_{TOT}(x_{1,1}^{\max})|$ for the optimal aggregation X^*.

However, it is not guaranteed that for this choice of the μ_k $(k = 1, \ldots, 6)$, we also have that $A = |(\mu_1 + \mu_2)\Phi_{TOT}(x_{1,5}^{\max})|$, since $\mu_1 + \mu_2$ is not necessarily nonnegative in the optimal aggregation X^*. Therefore, we also solve a second problem given by: maximize (12) subject to (13) to (29) and

$$A \leq -(\mu_1 + \mu_2)\Phi_{TOT}(x_{1,5}^{\max}) \tag{31}$$

In either of these two LP problems we will now have that $\mu_1 + \mu_2$ or $-(\mu_1 + \mu_2)$ is non-negative, and hence that either the LP problem with (30) or the one with (31) ensures that $A = |(\mu_1 + \mu_2)\Phi_{TOT}(x_{1,5}^{\max})|$ for the optimal aggregation X^*. That is, one of the two LP problems solves the problem we were aiming to solve, and the most sensitive aggregation (in terms of target function (12)) will be constructed. If the optimal value of (12) is positive, the thus constructed aggregation is unsafe. Otherwise, even the most sensitive aggregation where the largest (absolute) contribution to cell $R_1 \times C_1$ is attacked by the largest (absolute) contribution to cell $R_1 \times C_5$ is still safe.

4 Finding Secondary Cell Suppressions

The LP formulation for checking whether a table with suppressed cell values is safe for publication can also be used as a basis for a method for actually finding secondarily suppressed cells, i.e. for solving the secondary suppression problem. The simplest way of doing this is to combine the LP approach described in Section 3 with the so-called hypercube approach for cell suppression (see, e.g., [13], [21] and [22]).

The hypercube approach is essentially a sequential approach, where for each unsafe table a hypercube of suppressed cells is constructed in order to protect the selected unsafe cell. In a two-dimensional table without any hierarchical structure such a hypercube is simply a rectangle. In a three-dimensional table without any hierarchical structure, such a hypercube is a cuboid.

In the basic version of the hypercube approach, a hypercube is constructed by first testing all possible hypercubes with the selected unsafe cell as one of its corner points. When a hypercube sufficiently protects the selected unsafe cell, i.e. if the largest (absolute) contribution of the unsafe cell under consideration is sufficiently protected in any aggregation that can be constructed from this hypercube, the hypercube is a candidate for suppression. Of all candidate hypercubes, the one that leads to the least information loss is selected, and all corner points of the selected hypercube are suppressed. This is done for all unsafe cells. The final suppression pattern is the union of the hypercubes for the individual unsafe cells.

In order to combine our LP approach for the auditing problem with the hypercube approach, all we need to do is to invoke our LP approach whenever we need to test whether a hypercube sufficiently protects the selected unsafe cell. That is, we need to check if any of the contributors to a corner point of the hypercube can estimate the value of the largest (absolute) contribution to the unsafe cell too accurately. We have implemented this basic idea in prototype software in order to test whether this approach is indeed feasible in practice.

That a table with suppressed cells obtained by the hypercube method in combination with our LP approach for the auditing problem is safe is non-trivial and requires some proof. Table 10 illustrates why it is not trivial that a combination of safe hypercubes - one for each unsafe cell - leads to a safe table.

Let us assume that cells $x_{2,2}$ and $x_{4,5}$ are unsafe, that the (safe) hypercube for $x_{2,2}$ is given by $x_{1,1}$, $x_{1,2}$, $x_{2,1}$ and $x_{2,2}$ itself, and that the (safe) hypercube for $x_{4,5}$ is given by $x_{2,2}$, $x_{2,5}$, $x_{4,2}$ and $x_{4,5}$ itself. One could imagine that it is possible for $x_{2,2}$ or $x_{4,5}$ to be involved in an unsafe aggregation that involves both hypercubes, for instance the aggregation $x_{1,1} - x_{2,2} + x_{4,5} = 18$, which is obtained by adding $x_{1,1} + x_{1,2} = 60$ to $x_{4,2} + x_{4,5} = 162$, and then subtracting the aggregation $x_{1,2} + x_{2,2} + x_{4,2} = 204$. We will, however, show that it is not possible for $x_{2,2}$ or $x_{4,5}$ to be involved in an unsafe aggregation that involves both hypercubes.

Table 10: A table with suppressions

	C_1	C_2	C_3	C_4	C_5	Total
R_1	$x_{1,1}$	$x_{1,2}$	48	35	18	161
R_2	$x_{2,1}$	$x_{2,2}$	51	80	$x_{2,5}$	289
R_3	59	88	16	28	86	277
R_4	61	$x_{4,2}$	59	94	$x_{4,5}$	376
R_5	3	93	82	41	5	224
Total	186	385	256	278	222	1327

We start by proving a lemma.

Lemma: Suppose that an unsafe cell x_i is sufficiently protected in a certain aggregation of suppressed cells X_j, i.e. none of other contributors to aggregation X_j can derive the value of the largest (absolute) contribution $x_i^{[1]}$ to that cell to within $p\%$. Aggregation X_j could, for instance, have been constructed from a hypercube of suppressed cells. Suppose furthermore that we have a larger aggregation X_k that contains X_j, i.e. if $X_j = \sum_{i \in S} \lambda_i^j x_i$ for some set of suppressed cells S then $X_k = \sum_{i \in S} \lambda_i^j x_i + \sum_{t \in T} \lambda_t^k x_t$ where T is another set of suppressed cells ($T \cap S = \emptyset$). Then unsafe cell x_i is sufficiently protected from an attack from any contributor to X_k.

Proof: Let us denote the largest (absolute) contribution to cell x_i by $x_i^{[1]}$. Since cell $x_i^{[1]}$ is sufficiently protected in aggregation X_j we have $(p+q)|\lambda_i^j x_i^{[1]}| + q\Phi^{[2]}_{ABS} - q\sum_{r=1}^{R} \Phi^{[r]}_{ABS}(X_j) \leq 0$. For aggregation X_k we have two options: the largest (absolute) contribution to X_k besides $\lambda_i^j x_i^{[1]}$ is the second largest (absolute) contribution to X_j, i.e. $\Phi^{[2]}_{ABS}(X_k) = \Phi^{[2]}_{ABS}(X_j)$, or the largest (absolute) contribution to X_k besides $\lambda_i^j x_i^{[1]}$ is not the second largest (absolute) contribution to X_j, i.e. the largest (absolute) contribution to X_k besides $\lambda_i^j x_i^{[1]}$ is not involved in X_j.

In the first case we have for X_k:

$$
\begin{aligned}
(p+q)|\lambda_i^j x_i^{[1]}| + q\Phi^{[2]}_{ABS}(X_k) - q\sum_{r=1}^{R} \Phi^{[r]}_{ABS}(X_k) = \\
(p+q)|\lambda_i^j x_i^{[1]}| + q\Phi^{[2]}_{ABS}(X_j) - q\sum_{r=1}^{R} \Phi^{[r]}_{ABS}(X_k) \leq \\
(p+q)|\lambda_i^j x_i^{[1]}| + q\Phi^{[2]}_{ABS}(X_j) - q\sum_{r=1}^{R} \Phi^{[r]}_{ABS}(X_j) \leq 0
\end{aligned}
$$

The first inequality follows from the assumption that X_k contains X_j.

In the second case we have for X_k:

$$(p+q)|\lambda_i^j x_i^{[1]}| + q\Phi^{[2]}_{ABS}(X_k) - q\sum_{r=1}^{R}\Phi^{[r]}_{ABS}(X_k) =$$

$$(p+q)|\lambda_i^j x_i^{[1]}| - q\sum_{r=1,r\neq 2}^{R}\Phi^{[r]}_{ABS}(X_k) \leq$$

$$(p+q)|\lambda_i^j x_i^{[1]}| - q\sum_{r=1}^{R}\Phi^{[r]}_{ABS}(X_j) \leq$$

$$(p+q)|\lambda_i^j x_i^{[1]}| + q\Phi^{[2]}_{ABS}(X_j) - q\sum_{r=1}^{R}\Phi^{[r]}_{ABS}(X_j) \leq 0$$

The first inequality follows from $-q\sum_{r=1,r\neq 2}^{R}\Phi^{[r]}_{ABS}(X_k) \leq -q\sum_{r=1}^{R}\Phi^{[r]}_{ABS}(X_j)$ since the largest (absolute) contribution to X_k besides $\lambda_i^j x_i^{[1]}$ is not involved in X_j and X_j is contained in X_k.

In both cases we conclude that the largest (absolute) contribution to cell x_i is sufficiently protected in X_k, which concludes the proof of the lemma. ■

Note that aggregation X_k itself need not be safe since it may involve other unsafe cells besides cell x_i. Those other unsafe cells may be insufficiently protected in aggregation X_k.

Table 11: Illustration of part of the proof of the lemma

	C_1	C_2	C_3	C_4	C_5	Total
R_1	$x_{1,1}$	$x_{1,2}$	300	200	100	800
R_2	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	200	300	750
R_3	250	$x_{3,2}$	$x_{3,3}$	200	350	1000
Total	450	250	500	600	750	2550

We provide a short illustration of the second case mentioned in the proof of the above lemma using Table 11. Suppose $x_{2,1}$ and $x_{2,3}$ are the unsafe cells in Table 11. We assume that both of these cells have only one contributor with contribution 100. The hypercube constructed to protect $x_{2,1}$ consists of $x_{1,1}$, $x_{1,2}$, $x_{2,2}$ and $x_{2,1}$ itself. The hypercube constructed to protect $x_{2,3}$ consists of $x_{3,3}$, $x_{3,2}$, $x_{2,2}$ and $x_{2,3}$ itself. Suppose cell $x_{2,2}$ consists four contributors, three with a contribution of 10 and one with a contribution of 20. We assume that a (p,q)-rule with $p = 20$ and $q = 100$ is used. One aggregation in the hypercube for $x_{2,1}$ is $x_{2,1} + x_{2,2} = 150$. Based on this aggregation, the largest (absolute) contribution to cell $x_{2,2}$ can derive an upper bound of $150 - 20 - 0 = 130$ (the total for the aggregation minus the attacker's own contribution minus lower bounds for the other contributors to cell $x_{2,2}$) for the (only) contribution to cell $x_{2,1}$. Since this deviates more than 20% from the true value, the only contribution to cell $x_{2,1}$ is indeed sufficiently protected in this aggregation. Similarly, the only contribution to cell $x_{2,3}$ is sufficiently protected in aggregation $x_{2,2} + x_{2,3} = 150$.

We examine aggregation $x_{2,1} + x_{2,2} + x_{2,3} = 250$ and check what the contributor to cell $x_{2,3}$ can derive about the contribution to cell $x_{2,1}$. The contributor to cell $x_{2,3}$ can derive the following upper bound on the contribution to cell $x_{2,1}$: $250 - 100 - 0 = 150$ (the total for the aggregation minus the attacker's own contribution minus lower bounds for the other

contributors to cell $x_{2,3}$). The only contribution to $x_{2,1}$ is better protected in aggregation $x_{2,1}+x_{2,2}+x_{2,3}=250$ than in aggregation $x_{2,1}+x_{2,2}=150$, which was checked when the hypercubes were constructed, since the contribution of the largest (absolute) contribution to $x_{2,2}$ is no longer subtracted (cf. the first inequality of the second case of the lemma). Similarly, the contribution to $x_{2,3}$ is better protected in aggregation $x_{2,1}+x_{2,2}+x_{2,3}=250$ than in aggregation $x_{2,2}+x_{2,3}=150$.

Below we will illustrate how the above lemma can be applied to Table 10. In this illustration we will use the term "restricting an aggregation to a hypercube". We will explain this term for a hypercube in a two-dimensional table. Suppose this hypercube is given by x_{i_1,j_1}, x_{i_1,j_2}, x_{i_2,j_1} and x_{i_2,j_2}, where $i_1 \neq i_2$ and $j_1 \neq j_2$. We assume that besides this hypercube more cells with indices given by a set U have been suppressed, where $U \neq \emptyset$ and $U \cap \{(i_1,j_1),(i_1,j_2),(i_2,j_1),(i_2,j_2)\} = \emptyset$. The explicit aggregations involving x_{i_1,j_1}, x_{i_1,j_2}, x_{i_2,j_1} or x_{i_2,j_2} are then given by

$$
\begin{aligned}
&x_{i_1,j_1}+x_{i_1,j_2}+\sum_{(i_1,j)\in U} x_{i_1,j}\\
&x_{i_1,j_1}+x_{i_2,j_1}+\sum_{(i,j_1)\in U} x_{i,j_1}\\
&x_{i_2,j_1}+x_{i_2,j_2}+\sum_{(i_2,j)\in U} x_{i_2,j}\\
&x_{i_1,j_2}+x_{i_2,j_2}+\sum_{(i,j_2)\in U} x_{i,j_2}
\end{aligned}
$$

Any (explicit or implicit) aggregation (possibly) involving x_{i_1,j_1}, x_{i_1,j_2}, x_{i_2,j_1} or x_{i_2,j_2} is hence given by

$$
\begin{aligned}
&(\mu_1+\mu_2)x_{i_1,j_1}+(\mu_1+\mu_4)x_{i_1,j_2}+(\mu_2+\mu_3)x_{i_2,j_1}+(\mu_3+\mu_4)x_{i_2,j_2}+\\
&\mu_1\sum_{(i_1,j)\in U} x_{i_1,j}+\mu_2\sum_{(i,j_1)\in U} x_{i,j_1}+\mu_3\sum_{(i_2,j)\in U} x_{i_2,j}+\mu_4\sum_{(i,j_2\in U} x_{i,j_2}+\\
&\text{terms involving suppressed cells with indices in } U
\end{aligned}
$$

for some μ_1, μ_2, μ_3 and μ_4.

The restriction of such an aggregation to the hypercube given by x_{i_1,j_1}, x_{i_1,j_2}, x_{i_2,j_1} and x_{i_2,j_2} is then defined as

$$(\mu_1+\mu_2)x_{i_1,j_1}+(\mu_1+\mu_4)x_{i_1,j_2}+(\mu_2+\mu_3)x_{i_2,j_1}+(\mu_3+\mu_4)x_{i_2,j_2}$$

We can define the restriction of an aggregation to any set of suppressed cells - instead of a hypercube of supressed cells - that was constructed to protect a certain unsafe cell in a similar way.

Now, let us consider an aggregation in Table 10 in which an unsafe cell is involved, say $x_{1,1}-x_{2,2}+x_{4,5}=18$, which in fact involves both unsafe cells $x_{2,2}$ and $x_{4,5}$. We now restrict this aggregation to the hypercube for the first unsafe cell $x_{2,2}$. This gives the aggregation $x_{1,1}-x_{2,2}$. During the cell suppression process it was concluded that the largest (absolute) contribution to unsafe cell $x_{2,2}$ is sufficiently protected in this latter aggregation. By applying the lemma, we conclude that the largest (absolute) contribution to unsafe cell $x_{2,2}$ is also sufficiently protected in aggregation $x_{1,1}-x_{2,2}+x_{4,5}$.

Similarly, we restrict $x_{1,1} - x_{2,2} + x_{4,5} = 18$ to the hypercube for the second unsafe cell $x_{4,5}$. This gives the aggregation $x_{4,5} - x_{2,2}$. During the cell suppression process it was concluded that the largest (absolute) contribution to unsafe cell $x_{4,5}$ is sufficiently protected in this latter aggregation. By applying the lemma, we conclude that the largest (absolute) contribution to unsafe cell $x_{4,5}$ is also sufficiently protected in aggregation $x_{1,1} - x_{2,2} + x_{4,5}$. Since both unsafe cells involved in the aggregation are sufficiently protected, the aggregation itself is sufficiently protected.

The same idea, i.e. restricting an aggregation based on several hypercubes to a single hypercube that was constructed to protect a certain unsafe cell and then applying the lemma, can be used to show that other tables that are protected by means of the hypercube method in combination with our LP approach for the auditing problem are safe too. Namely, restricting an aggregation based on several hypercubes to a single hypercube that was constructed to protect a certain unsafe cell leads to an aggregation in that hypercube in which the largest (absolute) contribution to the unsafe cell is sufficiently protected. This holds true more generally: restricting any aggregation to any set of suppressed cells that was constructed to protect a certain unsafe cell leads to an aggregation in which the largest (absolute) contribution to that unsafe cell is sufficiently protected. The lemma shows that a cell is at least as well protected in a larger aggregation than in a smaller aggregation that is contained in the larger aggregation.

The above-described basic idea for constructing hypercubes can (and sometimes needs) to be extended, because in some cases suppressing a single hypercube may not suffice to protect a particular unsafe cell and more (hypercubes of) cells need to be suppressed. Note that the lemma applies to this situation as well. Another reason for extending the basic idea for constructing hypercubes is that after several hypercubes have been suppressed, some of the earlier suppressed cells may not need to be suppressed after all and may be released without disclosing any unsafe information. Whether earlier suppressed cells may be released after all can be checked by means of our LP approach for the auditing problem. We have not implemented such extensions in our prototype software, but have only implemented the basic version of the hypercube approach.

5 Test results

As already mentioned, we have developed prototype software for testing our LP approaches for the auditing problem and the cell suppression problem. For this, we have developed simple tailor-made software in Delphi. For solving linear programming problems, we have used the code from [20]. Much more efficient solvers for linear programming problems are nowadays available, not only commercially, but also freely. The development of efficient software is, however, out of scope of this paper. We have developed our prototype software only to show that our proposed LP approaches for the auditing and cell suppression problems are indeed feasible.

5.1 The auditing problem

Our LP approach for the auditing problem indeed shows that Table 4 (example 1) and Table 7 (example 2) described earlier are unsafe according to the criterion proposed by [5].

We give another example. Table 12 requires protection against disclosure of sensitive information. The parameters of the used protection rule again are $p = 20$, $q = 100$. Cell $R_1 \times C_1$ is unsafe, whereas the other cells in Table 12 are safe.

Table 12: An unprotected table (example 4), in brackets the largest two contributions to each cell are mentioned

	C_1	C_2	C_3	Total
R_1	100 (90 / 5)	1200 (600 / 360)	2100 (1050 / 630)	3400
R_2	1000 (500 / 300)	80 (75 / 3)	1600 (800 / 480)	2680
R_3	2200 (1100 / 660)	3100 (1550 / 930)	4800 (2400 / 1440)	10100
Total	3300	4380	8500	16180

According to traditional criterion for the auditing problem described in Section 2.2, the upper bound on the suppression interval for cell $R_1 \times C_1$ should be at least 113. Namely, when the upper bound on the suppression interval cell $R_1 \times C_1$ equals 113, the second largest contributor to cell $R_1 \times C_1$ can derive that the upper bound on the largest contribution is 108 $(= 113 - 5)$. This upper bound exceeds the actual value (i.e. 90) of the largest contribution to cell $R_1 \times C_1$ by exactly 20%.

Similarly, the lower bound on the suppression interval for cell $R_1 \times C_1$ should be at most 87, for then the second largest contributor to cell $R_1 \times C_1$ can derive that the lower bound on the largest contribution is 72 $(= 87 - 5 - 2 \times 5$, where the first 5 is the contribution of the second largest contributor to cell $R_1 \times C_1$ and 2×5 is the upper bound on the contributions of the other contributors to cell $R_1 \times C_1$). A lower bound of 72 on the largest contribution to cell $R_1 \times C_1$ is exactly 20% less than its actual value.

With the requirement that the upper bound on the suppression interval is at least 113 and the lower bound at most 87, τ-ARGUS gives Table 13 as output, if we measure information loss due to suppression by the sum of the suppressed cell values.

Table 13: A safe table version according to τ-ARGUS (example 4)

	C_1	C_2	C_3	Total
R_1	$\times$	$\times$	2100	3400
R_2	$\times$	$\times$	1600	2680
R_3	2200	3100	4800	10100
Total	3300	4380	8500	16180

We check whether the privacy of the largest contributor to $R_1 \times C_1$ is sufficiently protected against an attack from the largest contributor to cell $R_2 \times C_2$ according to the alternative criterion proposed by [5]. The result of our LP approach, and of our prototype software, is that Table 13 is unsafe according to that criterion. To check this result of our LP approach, we introduce the notation in Table 14.

Table 14: Checking the table protected by τ-ARGUS (example 4)

	C_1	C_2	C_3	Total
R_1	$x_{1,1}$	$x_{1,2}$	2100	3400
R_2	$x_{2,1}$	$x_{2,2}$	1600	2680
R_3	2200	3100	4800	10100
Total	3300	4380	8500	16180

One of the (implicit) aggregations that can be obtained from this table is $x_{1,1} - x_{2,2} = 20$. This aggregation, for instance, follows by subtracting the explicit aggregation $x_{1,2} + x_{2,2} =$

1280 from the explicit aggregation $x_{1,1} + x_{1,2} = 1300$.

The largest contributor to cell $R_2 \times C_2$ can hence derive that the upper bound on the largest contribution to cell $R_1 \times C_1$ is $20 + 75 + 10 - 0 = 105$ (the first term follows from the above implicit aggregation, the second term is his own contribution to cell $R_2 \times C_2$, the third term is an upper bound on the other contributions to cell $R_2 \times C_2$, and the fourth term is a lower bound on the other contributions to cell $R_1 \times C_1$. These latter upper and lower bounds can be computed by using the q parameter). That is, Table 13 is indeed unsafe according to the alternative criterion proposed by [5], because $105 \leq 90 + 20\% \times 90 = 108$.

Similarly, the largest contributor to cell $R_2 \times C_2$ can also derive that the lower bound on the largest contribution to cell $R_1 \times C_1$ is $20 + 75 + 0 - 20 = 75$ (the first term follows from the implicit aggregation, the second term is his own contribution to cell $R_2 \times C_2$, the third term is a lower bound on the other contributions to cell $R_2 \times C_2$, and the fourth term is an upper bound on the other contributions to cell $R_1 \times C_1$. These latter upper and lower bounds can again be computed by using the q parameter). This again confirms that Table 13 is unsafe according to the alternative criterion proposed by [5].

The LP approach and our prototype software construct the implicit aggregation $x_{1,1} - x_{2,2} = 20$, and conclude that too much information on the largest contribution to cell $R_1 \times C_1$ can be disclosed from this aggregation by the largest contributor to cell $R_2 \times C_2$.

5.2 Finding cell suppressions

We have also tested our prototype software for the cell suppression problem. We were in particular interested to see if our prototype software indeed avoids selecting the cell suppression patterns that are selected by τ-ARGUS, but that are unsafe according to the alternative criterion proposed by [5].

The objective of our prototype software is to minimize the sum of the suppressed cell values.

For Table 6 (example 2) our prototype software finds the suppression pattern in Table 15, and for Table 12 (example 4) it finds the suppression pattern in Table 16. Tables 15 and 16 are indeed safe according to the alternative criterion proposed by [5].

Table 15: A table protected by our prototype software (example 2)

	C_1	C_2	C_3	Total
R_1	×	380	×	880
R_2	50	80	60	190
R_3	×	800	×	1680
Total	820	1260	670	2750

Table 16: A table protected by our prototype software (example 4)

	C_1	C_2	C_3	Total
R_1	×	1200	×	3400
R_2	1000	80	1600	2680
R_3	×	3100	×	9100
Total	3300	3380	8500	15180

Given that Tables 15 and 16 are safe according to the alternative criterion proposed by [5], it is easy to verify that Tables 15 and 16 are indeed the protected tables that lead to the least

information loss. Namely, the cell suppression patterns with at least one secondarily suppressed cell in each row and each column of the unsafe cell $R_1 \times C_1$ that have the lowest information loss for Tables 6 and 12 are given in Tables 7 and 13, respectively. However, the cell suppression patterns in Tables 7 and 13 lead to unsafe tables according to the criterion proposed by [5]. Tables 15 and 16 give the suppression patterns with at least one secondarily suppressed cell in each row and each column of the unsafe cell $R_1 \times C_1$ that have the second lowest information loss for Tables 6 and 12, respectively. Since Tables 15 and 16 are safe according to the criterion proposed by [5], these tables are therefore the protected tables with the least information loss.

6 Discussion

In this paper, we have shown that it is feasible to apply the criterion proposed by [5] for the auditing problem in practice. This can be done by implementing this criterion by means of a series of LP problems rather than by means of a large mixed-integer programming problem as was originally proposed by [5]. This shows that the auditing problem based on the criterion proposed by [5] can be solved in polynomial time in its input parameters, rather than in non-polynomial time as the mixed-integer programming problem formulation of [5] suggests.

In principle, this approach based on using a series of LP problems can also be used for the development of a cell suppression method and accompanying software. One way to do this is by combining the LP approach for the auditing problem with the hypercube approach as sketched in Section 4 of the current paper. In order to show the feasibility of such an approach we developed some theory in Section 4.

We have not studied how often the alternative criterion for the auditing problem proposed by [5] leads to major differences with the traditional criterion. The examples where the traditional criterion gives undesirable results given in this paper were specially constructed for demonstrating the flaws of the traditional criterion. So, it is quite possible that the differences with the traditional criterion may be small in practice. From a theoretical point of view, we feel that the difference between the alternative criterion proposed by [5] and the traditional criterion is important as the traditional criterion can lead to inconsistencies with the applied sensitivity measure for individual cells. The alternative criterion proposed by [5] avoids this inconsistency. Our contribution in the current paper is a step towards implementability of this alternative criterion in practice.

We are aware of the fact that our LP approach is only a small step towards implementability of the alternative criterion. More efficient algorithms for the auditing problem and, especially, the cell suppression problem are highly desired. Undoubtedly, such more efficient algorithms can indeed be developed. We leave the development of such algorithms to specialists in operations research.

Related to the development of more efficient algorithms for the auditing and the cell suppression problems is the development of efficient and user-friendly software for solving these problems in practice. For this paper we have developed simple prototype software, demonstrating the feasibility of implementing the alternative criterion proposed by [5] as a series of LP problems. As already mentioned in the Introduction, we do not claim any superiority of our prototype software over any available software for the auditing problem or the cell suppression problem, nor over any software based on the mixed-integer programming problem formulation in [5]. For producing software based on our LP approach that can be used in the day-to-day routine at a statistical office, a substantial effort would

be required.

In this paper we have assumed that holdings do not occur in the data to be protected. Our approach should preferably be extended in order to take holdings into account. The basic idea of such an extension is quite straightforward: protect the largest contributing holding to an aggregation against an attack from the second largest contributing holding to that aggregation. The translation of this idea into mathematical machinery may, however, be quite complicated, especially if one wants to develop efficient software for this situation. We leave the extension of our approach to holdings as a potential topic for future research.

Even more important seems to be the development of an extension of our approach to hierarchical tables, where, for instance, cells sum up to subtotals, which in turn sum up to higher-level totals. Hierarchical tables frequently occur in practice at NSIs. The hierarchical structure of such tables has to be taken into account while solving the auditing problem and during the cell suppression process. Extending our approach to hierarchical tables would be an important step towards the use of this (extended) approach in practice.

References

[1] Cox, L.H. (1980), Suppression Methodology and Statistical Disclosure Control. *Journal of the American Statistical Association 75*, pp. 377-385.

[2] Cox, L.H. (1981), Linear Sensitivity Measures in Statistical Disclosure Control. *Journal of Statistical Planning and Inference 5*, pp. 153-164.

[3] Cox, L.H. (1995), Protecting Confidentiality in Business Surveys. In: *Business Survey Methods* (eds. B.G. Cox, D.A. Binder, B.N. Chinnappa, A. Christianson, M.J. Colledge and P.S. Kott), John Wiley & Sons, Inc., New York, pp. 443-473.

[4] Cox, L.H. (2001), Disclosure Risk for Tabular Economic Data. In: *Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies* (eds. P. Doyle, J.I. Lane, J.J.M. Theeuwes and L.V. Zayatz), North-Holland Elsevier, Amsterdam, pp. 167-183.

[5] Daalmans, J. and T. de Waal (2010), An Improved Formulation of the Disclosure Auditing Problem for Secondary Cell Suppression. *Transactions on Data Privacy 3*, pp. 217-251.

[6] Dellaert, N.P. and W.A. Luijten (1999), Statistical Disclosure in General Three-Dimensional Tables. *Statistics Neerlandica 53*, pp. 197-221.

[7] De Wolf, P.-P., A. Hundepool, S. Giessing, J.-J. Salazar and J. Castro (2014), *τ-ARGUS (version 4.1) - User's Manual*. Statistics Netherland.

[8] Duarte De Carvalho, F., N.P. Dellaert and M. De Sanches Osório (1994), Statistical Disclosure in Two-Dimensional Tables: General Tables. *Journal of the American Statistical Association 89*, pp. 1547-1557.

[9] Duncan, G.T., S.E. Fienberg, R. Krishnan, R. Padman and S.R. Roehrig (2001), Disclosure Limitation Methods and Information Loss for Tabular Data. In: *Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies* (eds. P. Doyle, J.I. Lane, J.J.M. Theeuwes and L.V. Zayatz), North-Holland Elsevier, Amsterdam, pp. 135-166.

[10] Fischetti, M. and J.J. Salazar-González (2000), Models and Algorithms for Optimizing Cell Suppression in Tabular Data with Linear Constraints. *Journal of the American Statistical Association 95*, pp. 916-928.

[11] Giessing, S. (2001), Nonperturbative Disclosure Control Methods for Tabular Data. In: *Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies* (eds. P. Doyle, J.I. Lane, J.J.M. Theeuwes and L.V. Zayatz), North-Holland Elsevier, Amsterdam, pp. 185-213.

[12] Giessing, S. (2004), Survey on Methods for Tabular Data Protection in ARGUS. In: *Privacy in Statistical Databases* (eds. J. Domingo-Ferrer and V. Torra), Springer-Verlag, Berlin, pp. 1-13.

[13] Giessing, S. and D. Repsilber (2002), Tools and Strategies to Protect Multiple Tables with the GHQUAR Cell Suppression Engine. In: *Inference Control in Statistical Databases*, From Theory to Practice (ed. J. Domingo-Ferrer), Springer Lecture Notes in Computer Science, Vol. 2316, pp. 181-192.

[14] Hundepool, A., J. Domingo-Ferrer, L. Franconi, S. Giessing, E. Schulte Nordholt, K. Spicer and P.-P. de Wolf (2012), *Statistical Disclosure Control*. Wiley & Sons, Chichester.

[15] Johnson, E.I., G.l. Nemhauser, and M.W.P. Savelsbergh (2000), Progress in Linear Programming-Based Algorithms for Integer Programming: An Exposition. *FORMS Journal on Computing, Vol. 12*, pp. 2-23.

[16] Jünger, M., G. Reinelt, and G. Rinaldi (1995), The Traveling Salesman Problem, In: *Handbooks in Operations Research and Management Science, Volume 7: Network Models* (eds. M.O. Ball, T.L. Magnanti, C. Monma, and G.L. Nemhauser), Elsevier, Amsterdam, pp. 225–330.

[17] Kelly, J.P., B.L. Golden and A.A. Assad (1992), Cell Suppression: Disclosure Protection for Sensitive Tabular Data. *Networks 22*, pp. 397-417.

[18] Ladányi L., T.K. Ralphs and L.E. Trotter (2001), Branch, Cut, and Price: Sequential and Parallel. In: *Computational Combinatorial Optimization. Lecture Notes in Computer Science, Vol 2241* (eds. M. Jünger and D. Naddef), Springer, Berlin, Heidelberg, pp. 223-260.

[19] Nemhauser, G.L. and L.A. Wolsey (1988), *Integer and Combinatorial Optimization*, Wiley, New York.

[20] Press, W.H., B.P. Flannery, S.A. Teukolsky, B.P. Flannery and W.T. Vetterling (1989), *Numerical Recipes in Pascal: The Art of Scientific Computing*. Cambridge University Press, Cambridge.

[21] Repsilber, D. (1994), *Preservation of Confidentiality in Aggregated Data*. Paper presented at the Second International Seminar on Statistical Confidentiality, Luxembourg.

[22] Repsilber, D. (2002), Sicherung Persönlicher Angaben in Tabellendaten. In: *Statistische Analysen und Studien Nordrhein-Westfalen*, Landesamt für Datenverarbeitung und Statistik NRW, Ausgabe 1/2002 (in German).

[23] Robertson, D. (2000), Improving Statistics Canada's Cell Suppression Software (CONFID). In: *Proceedings in Computational Statistics 2000* (eds. J.G.Bethlehem and P.G.M. Van der Heijden), Physica-Verlag, New York, pp. 403-408.

[24] Salazar-González, J.J. (2002), Extending Cell Suppression to Protect Tabular Data against Several Attackers. In: *Inference Control in Statistical Databases, From Theory to Practice* (ed. J. Domingo-Ferrer), Springer, pp. 34 - 58.

[25] Sande, G. (1977), *Towards Automated Disclosure Analysis for Establishment Based Statistics*. Report, Statistics Canada.

[26] Sande, G. (1978a), *A Theorem Concerning Elementary Aggregations*. Report, Statistics Canada.

[27] Sande, G. (1978b), *Confidentiality and Polyhedra - An Analysis of Suppressed Entries and Cross-Tabulations*. Report, Statistics Canada.

[28] Sande, G. (2000), *Blunders by Official Statistical Agencies while Protecting the Confidentiality of Business Statistics* (unpublished paper).

[29] Willenborg, L. and T. de Waal (1996), *Statistical Disclosure Control in Practice*. Springer-Verlag, New York.

[30] Willenborg, L. and T. de Waal (2001), *Elements of Statistical Disclosure Control*. Springer-Verlag, New York.

[31] Wolsey, L.A. (1998), *Integer Programming*, Wiley, New York.