

**STATISTICAL COMMISSION and
ECONOMIC COMMISSION FOR EUROPE**

**COMMISSION OF THE
EUROPEAN COMMUNITIES**

CONFERENCE OF EUROPEAN STATISTICIANS

EUROSTAT

**Joint ECE/Eurostat Work Session on
Statistical Data Confidentiality**

(Skopje, The former Yugoslav Republic of Macedonia,
14-16 March 2001)

Working Paper No. 2
English only

Topic I: Application of statistical disclosure control methodology and software in business statistics and social and demographic statistics

**NEW TOOLS FOR CELL SUPPRESSION IN TAU-ARGUS:
ONE PIECE OF THE CASC PROJECT WORK DRAFT**

Invited paper

Submitted by the Federal Statistical Office of Germany¹

Abstract: In the course of the EU funded project CASC, the software τ -ARGUS shall be extended to become a generally applicable standard tool for tabular data protection. The required modifications will affect the facilities provided for (residual) disclosure risk statement, the data structure and the user interface. Methods will have to be implemented for protection of complex hierarchical tables and for table-to-table protection, especially in the context of public use data-base query systems. Extensions will be made concerning secondary cell suppression methodology provided by the package. The package will interface in particular with the GHQUAR hypercube algorithm. Finally, table perturbation tools will be added.

Keywords: CASC project, τ -ARGUS, tabular data protection, hierarchical tables, table-to-table protection, secondary cell suppression

I. INTRODUCTION

1. One of the projects funded by the European Union within the European Plan for Research in Official Statistics (EPROS) as part of the 5th Framework programme is the project CASC (Computational Aspects of Statistical Confidentiality). The overall management of the project is the responsibility of Anco Hundepool of Statistics Netherlands.

2. Work on this project will be carried out between January 2001 and December 2003. The project is meant to be a follow up for the SCD project of the 4th Framework, in the sense that it will build further on the achievements of that project, and take over the results and products emerging from the SDC-project. One of the main tasks will be to further develop the ARGUS-software, e.g. the software : -ARGUS for creation of safe micro-data files and the software τ -ARGUS for tabular data protection. It will be the objective of this paper to describe the project work draft concerning methodology for τ -ARGUS. The author is member of the CASC steering committee and as such she will monitor research and development of supplementary methodology for tabular data protection in τ -ARGUS.

3. Concerning tabular data protection, work within the CASC project will be carried out by the Statistical Institutes of the Netherlands, Germany and Nordrhein-Westfalen/Germany (CBS, StBa, LDS NRW), and teams at the Universitat La Laguna (ULL), Technische Universität Ilmenau (TUI), and Universitat Politècnica de Catalunya (UPC) . CBS will perform the implementation in τ -ARGUS, ULL will provide optimal search algorithms, StBa will suggest methodological strategies, LDS NRW will

¹ Prepared by Sarah Giessing (e-mail: sarah.giessing@statistik-bund.de).

supply the GHQUAR hypercube algorithm, UPC will provide an alternative optimisation algorithm and it will be the main task of TUI to develop algorithms for particular table perturbation techniques. StBa has a co-ordinating role in the research and CBS for the software development and research regarding linear programming.

II. OBJECTIVES OF THE CASC PROJECT CONCERNING METHODOLOGY FOR TABULAR DATA PROTECTION

4. Concerning tabular data protection, it is the objective of the project to develop a software package suitable to be established as a standard tool for disclosure control of aggregated data. This implies that the software package must be able to deal with tables of any size and complexity of structure, facilities must be offered to deal with specific problems of particular situations in a flexible, user-friendly and comfortable way. The software must be easily accessible and usable, and most of all, the package must be able to strike a good balance between quality and quantity. That is, the package should be able to offer, depending on the particular situation (e.g. size and complexity of the particular application), the best suppression patterns (in terms of information loss due to suppression) efficiently achievable (in terms of computing resource requirements).

5. Specific actions to address the overall goal of the work-package are the following:

- (i) Refine and support the integration of desirable qualities and facilities of existing software systems for tabular data protection into τ -ARGUS.
- (ii) Integration of the most recent version of the GHQUAR software, which will ensure wide applicability of the package to (linked) tables of any size and complexity of structure.
- (iii) Significant improvement of the cell suppression algorithms based on linear programming as already supplied along with the package, development of alternative techniques based on network flow methodology, and supply of supplementary heuristic methodology.
- (iv) Provide information on the performance of the various algorithms for secondary cell suppression to be included in the final package. This information will support the transfer of the package and will be useful as well for guiding internal decisions to be made during the project.
- (v) Methods and tools will be provided so as to maximise the information content of tables with suppressed entries, e.g. after a suppression procedure has been carried out.
- (vi) Gain expertise with any newly implemented facilities for control of the selection of secondary suppressions and pass on this expertise to potential users. In particular the 'European dimension' of the secondary cell suppression problem will be addressed, e.g. how to ease and sustain approaches of co-ordinating suppression patterns within Europe, as suggested e.g. by Eurostat for application to data of the structural business survey, c.f. [(Doc. Eurostat/D2/SBS-T/NOV99/03)].

III. NEW FACILITIES ADDRESSING DISCLOSURE RISK SPECIFICATION

6. Unless one does not disseminate any data at all, there may always remain some residual disclosure risk, even in protected data. In advance of running a disclosure control procedure, a user will have to state to the system how much of certain disclosure risks would be acceptable to him.

III.1 Primary Disclosure Risk Assessment

7. The first step in a disclosure control procedure for a table is always to assess the disclosure risk that would be connected to the release of each cell within the table. This test is usually done by applying certain sensitivity rules to the data. A cell, that is sensitive according to the sensitivity rule employed, would not be published, e.g. 'suppressed'. Other cells (so-called 'secondary' or 'complementary' suppressions) must be suppressed along with these so-called 'primary suppressions', in order to prevent the possibility, that users of the published table would be able to recalculate the primary suppressions exactly, or to derive too narrow an estimate by making use of the linear relations between published and suppressed cells of the table.

8. Currently, τ -ARGUS offers to employ a concentration rule (e.g. (n,k)-dominance rule) in combination with a minimum number of respondents rule, with the parameters n and k of the dominance-rule and the minimum number of respondents left to be chosen by the user. These rules are special cases only (although the best-known and most commonly applied ones) of the more general class of ‘upper linear sensitivity measures’ (c.f. e.g. [1], or [11]). The new version of τ -ARGUS will allow to combine several upper linear sensitivity measures. It will be offered particularly to apply a p%-rule or a (p,q)-rule.

III.2 The Protection Interval

9. By making use of the linear relations between published and suppressed cells, users of a published table are able to derive upper and lower bounds for the true value of any suppressed entry. The interval given by these bounds is usually called the ‘*suppression interval*’. For proper selection of complementary suppressions, the disseminator should determine safety bounds for any primary suppression. We call the interval between the upper and lower safety bounds ‘*protection interval*’. The suppression procedure will ensure, that no suppression pattern will be considered feasible, unless the corresponding suppression interval encloses the protection interval for any sensitive cell.

10. In the current version of τ -ARGUS, the user is requested to specify bounds for the protection interval. However, it is quite essential for a suppression procedure that these bounds are determined properly. Otherwise, it may cause either a risk of disclosure, or over-suppression.

11. When for primary confidentiality a concentration rule, such as an (n,k)-dominance-rule or p%-rule has been employed, there would be an unacceptable risk of disclosure according to this rule, when the upper bound of the suppression interval does not exceed the true value of the sensitive cell sufficiently. If the distance between upper bound and true value is below a certain minimum size, then this upper bound could be used to derive an estimate for single contributions to the sensitive cell, which is too close according to the sensitivity criterion employed. Formulas for (upper) bounds for the protection interval meeting this criterion can easily be obtained making use of the results of [1] and are given in the appendix for the most common sensitivity rules. It is intended for the new version of τ -ARGUS, to offer a default option computing the protection interval according to these formulas.

12. Unfortunately, in certain cases, even when the protection interval has been determined properly as described above, certain risks of residual disclosure would still remain unattended to. When complementary suppressions are sensitive as well, or when cells, which are components of the same (sub)total cell, share common respondents, as will be illustrated in III.3 below, then the suppression pattern should also ensure, that the upper bound for the true value of the (suppressed) combined cell would not disclose any respondents (combined) contribution. This cannot be achieved through a proper definition of a protection interval for the sensitive cell alone, by making e.g. sure, that a complementary suppression satisfies a certain minimum size condition. The problem will be addressed and solutions shall be implemented, such as e.g. suggested in [7] or [10].

III.3 The problem of common respondents

13. Statisticians often construct the same table for different response variables. Sometimes there is an additive relation present in a set of response variables, e.g. one of the response variables will actually be the sum of the others in the set. An example of this kind of interrelationship is a relation such as “total investment = investment in building + investment in ground + investment in technical equipment + other investment”.

14. A common table protection technique used often in cases such as this, is to do the table protection only for one of the response variables (the “total investment” for instance), and then suppress in this example those entries in the “building”, “ground”, “technical equipment” and “other investment” tables corresponding to suppressed entries in the “total investment” table. Though this approach is certainly valuable in reducing the effort for data protection, it is preferable not to use it when the information given by any of the other response variables has to be considered both sensitive and identifying. Identifying

means that it can be assumed, that attackers are not only able to identify those respondents with extraordinarily large responses to the ‘overall’ variable (e.g. the total investment), but might as well be able to guess large respondents to the other variables. For our investment example, this might happen to be the case. If for example the table presents low-level aggregate data, and one of the respondents has had an expensive new building, while the other respondents did not have considerable costs for building, then many of those respondents may know, who has a very large share in their common published value for building investment, and be able to disclose this single contribution. So, this cell should be suppressed, even if the ‘overall’ investment may turn out to be safe. In this case, all the “investment” tables should be protected together as a single table, with the relation between the different investment categories as one dimension of this table.

15. Tables constructed in this fashion have a typical property: cells, which are components of the same (sub)total cell, share common respondents. If the combined contribution of a respondent to a union of these cells has to be considered as confidential information, such as, say, the “investment in building and ground”, then, as stated in III.2 above, the suppression pattern should also ensure, that the upper bound for the true value of the (suppressed) combined cell would not disclose any respondents combined contribution.

IV. NEW DATA STRUCTURES

16. In the current version, τ -ARGUS cannot handle tables with hierarchical substructure, nor tables with a decomposition structure of the response variable, as described in section III.3 above. There is no option for table-to-table protection of linked tables, and if, due to a decentralised organisation structure like within in the European or the German statistical system, a potential user of the software is unable of providing the micro-data set on which the table he wants to protect is based, he cannot use the system. All these facilities will be offered by new versions, which will of course require modification and new concepts in the data structures.

17. New data structures need to be designed for tabular data input, modifications will be made in the structure for the microdata input, and the design of files containing meta-information, codelists, and other structural information on the tables has to be modified or newly invented.

IV.1 The users perspective: what to expect regarding new interface options of t-ARGUS

18. This section will briefly illustrate alternatives for the kind of user input to be supplied for specification of hierarchical tables, and control of table-to-table protection procedures.

IV.1.1 Definition of hierarchical tables

19. For hierarchical tables, the user has to supply information on the structure of the table to be protected. Naturally, a variety of options can be imagined of how to provide this structural information. For users of the tabular data input option, e.g. users supplying tabular data, one possibility would be to extract this information from the data-file. In that case, the tabular-data input file for a hierarchical table should specify cells using hierarchical codes for any hierarchically structured variable. The software would then for each dimension of the table create a list of codes as appearing in the data file. This would of course be not an option in case the user supplies micro-data. A simple option here would be to ask the user to supply a list of hierarchical codes for each dimension of the table. Of course this list must properly suit the codes for this dimension as supplied by the micro-data file. The software would check whether certain conditions in this respect are fulfilled.

20. Many users may be quite happy with this kind of input information. Hierarchical codelists will be readily available for the most detailed variables, such as for instance NACE or NUTS for the variables ‘industry’ or ‘region’. When no such codelist is already available for a variable (typically variables such as ‘size class’, ‘year of foundation’, etc., the effort of creating the list might be considered rather low, as these variables normally tend to be given simple hierarchical structures (if any), with rarely more than 20 categories. So, designed along these lines, the software would be usable, although not very flexible, still

leaving a rather high organisational burden regarding data preparation, especially for non-standard applications, or when the input-data in some way may not instantly meet the requirements.

21. Considering, that we claim to design a standard tool to be used in so many different environments as there are within the European Statistical System at least, we might want to build a more flexible, user-friendly software. Such a software would offer tools to assist and to guide a user in the design of a hierarchical structure for a variable, and the modification of a given structure (for instance, if the table should not display data for all NACE codes, but only part of them), while preventing him from misspecifications, that might cause disclosure risk, or breakdown of the secondary cell suppression procedure. Tools of this kind will be useful essentially for users attempting to optimise the design of tables by ‘playing’ with the data, redesigning the table over and over again, dropping sub-totals or introducing new ones, and so forth, until finally ending up with a favourable table and an acceptable suppression pattern. This may exceed the needs of ‘new’ users with little experience in automated tabular data protection, however with growing expertise, especially when the task is not simply to protect one single table but multiple tables, maybe to the degree even of providing to some extent safe input for public use statistical data base query systems, it is expected that users will learn to appreciate this flexibility.

22. Challenges for us will be to meet existing table specification standards and to find a good balance between user-friendliness and not wasting software development capacity by ‘reinventing the wheel’, as of course there is already software available to handle the management of variable codes.

IV.1.2 Specification of table-to-table and data base protection procedures

23. In comparison with the effort required to create comfortable tools for specification of hierarchical tables as illustrated above, the effort connected to software development needed to create user input facilities to specify a table-to-table protection procedure is rather low. (This remark does however ignore the fact that comfortable tools for specification of hierarchical tables, such as discussed above, with a high software development burden connected to them, become truly useful chiefly in the context of table-to-table protection.) To perform a table-to-table protection run, the user will have to create his tables, and to define them to the system just as when protecting single tables only. Then he will list the tables for the run, and is done.

24. Similar comments regarding the effort required for their creation can be made as well in the context of database protection facilities. Options will be created in order to specify a set of (protected or unprotected) tables to be pooled in a single file. This pool file will contain one record only for each cell within the set of tables, particularly for cells appearing in more than one table. There will be pool meta files, giving historical information, e.g. showing which tables are already contained in the pool, which of them have already been protected, and will for protected tables provide information on the run-parameters used and log-files created when protecting them. Finally some simple options shall be provided to extract data from the pool file into ready-made tables.

25. Though this sounds quite simple, it should again be noted that, especially in the context of data base protection, other facilities, which otherwise might not really be needed, will be required for support, to improve the performance of such a procedure. We will denote them as:

IV.1.3 Preference facilities

26. ‘Preference facilities’ are supposed to make the system prefer or even force certain cells to remain unsuppressed, or on the contrary, to make certain cells be used as complements first. There will be options provided to define ‘preferences’ in automated fashion, like a switch to be turned on or off for certain classes of cells, such as sub-total cells, particular cells of overlapping parts of tables, cells used as secondary suppressions in a previous period for tables presenting results of periodical surveys, etc. On the other hand, there may options offered to define ‘preferences’ for user-defined sets of cells, where in the extreme a set may even contain a single cell only. Those cells may be specified as (sub)tables, to be stated as cross-combinations of subsets of the sets of codes used to define the original tables. There should of course be facilities provided then to select subsets from a code list.

27. We certainly will have to supply a lot of user guidance and default options for ‘beginners’ and less experienced users. If it should turn out in the course of the CASC project, that the project capacity (chiefly: the capacity available for software development) does not suffice to implement all the user-friendly facilities that may have proven to be useful, we will then at least try to make the design of the system open enough as to allow for the experienced user by supplying his own procedures (outside the system) or by manual intervention, to run the suppression procedure according to his individual needs.

IV.2 The developers perspective: methodological strategies for the extension of t-ARGUS

28. This section will outline strategies on how to apply a suppression algorithm for unstructured tables to hierarchical tables, how to implement table-to-table protection, and how to extend table-to-table protection efficiently in the context of data base query systems.

IV.2.1 Strategies for the protection of hierarchical tables

Single table approach

29. From a purely methodological point of view, it is actually the best strategy for protection of hierarchical tables, to not treat it at all as hierarchical table, but to turn it into one single(!) non-hierarchical table in advance. This is always possible and would be relatively simple to implement. In fact, this would be the only method to protect such a table properly, avoiding a particular kind of disclosure risk, which will be illustrated farther below. The challenge of this strategy would be to speed up the suppression algorithm for single, unstructured tables sufficiently, as to be powerful enough for application to real-life sized tables. It should be noted in this context, that in a non-hierarchical representation, a hierarchical table is much, much larger than in the hierarchical representation. Considering that the input for the suppression algorithm based on linear programming methodology is mainly a set of equations, that might (for simplicity) be regarded as describing linear relations between (potentially) suppressed and unsuppressed cells of the table, one will of course at a certain stage of the cell-suppression process drop from the set of equations resulting from the non-hierarchical representation all the non-valid ones, removing e.g. identity equations and equations appearing more than once (to the extent of not even creating them at all).

30. The secondary cell suppression problem is known to be computationally hard. The number of computations required for solving the secondary cell suppression problem stated as Integer Linear Programming (ILP) problem grows exponentially with the size of the table. It is therefore clear in advance, that it will not be an option to protect extremely large tables of several hundred thousand cells or more using linear programming methodology within such a single table approach.

‘Backtracking’ strategies

31. A common approach is to split the table into sub-tables and protect the sub-tables separately. In doing so, one must of course take into account that these sub-tables of the same table do have cells in common. Otherwise it might happen that the same cell is suppressed in one sub-table because it is used as a secondary suppression, while within another table it remains unsuppressed. A user comparing the two sub-tables would then be able to disclose confidential cells in the first table. A common method of complying with this is to note any complementary suppression belonging also to one of the other sub-tables, suppress it in this sub-table as well, and repeat the cell suppression procedure for this table. This approach is sometimes called a ‘*backtracking procedure*’, a denotation which we will follow here. Though within a backtracking procedure the cell-suppression procedure will usually be repeated several times for each sub-table, the number of computations required for the protection procedure will be much smaller than when the entire table is protected all at once. It must, however, be stressed that a backtracking procedure is not fail-safe. For example, even though each suppressed cell in the table may be protected properly in each subtable, it may still happen that cells can be disclosed exactly when all the linear relations between the cells of the entire table are considered. The problem will be illustrated considering a simple 2-dimensional table without substructure. We may consider this table as a set of interrelated 1-dimensional tables, e.g. the set of all the rows and columns of the table. We might then protect each row and each column separately, and only make sure that finally each row and each column

has been protected properly, which means each row and each column will contain at least two, or no suppressions at all. This is, however, not a sufficient criterion for a safe suppression pattern in a 2-dimensional table (c.f. [3] for counter example). The problem extends analogously to the n-dimensional case. (c.f. [8]).

32. The speed of the backtracking procedure can be increased when cells appearing in more than one sub-table are given a low probability to be selected as secondary suppression, which can be performed for instance using dynamical weighting schemes (c.f. IV.2.2 "Implementing preferences" below).

33. A natural way of splitting a hierarchical table into sub-tables would be to split the table into the set of sub-tables without substructure, e.g. in a set of tables constructed in the following way: For any explanatory variable we pick one particular non-bottom-level category. Then we construct a 'sub-variable'. This sub-variable consists only of the category picked in the first step and those categories of the hierarchical level below, belonging to this category. The table specified through this set of explanatory (sub)variables is free from substructure then, and is a sub-table of the original one, e.g. any cell within the sub-table does also belong to the original table. When we repeat this procedure for any combination of non-bottom-level categories in each dimension, we will have divided the original table into a set of sub-tables without substructure.

34. Instead of constructing sub-tables without any substructure, we can also construct tables with a less complex substructure. In that case we will not construct the above described 'sub-variables' for each explanatory variable, but only for a part of them. The table specified through original (hierarchical) variables in some dimensions and sub-variables in the other dimensions will then have a less complex structure as compared to the original table.

IV.2.2 Strategies for table-to-table and database protection

Table to Table protection

36. Usually some of the tables in the set of multiple tables published from the same source (e.g. response data from a survey) will be overlapping. Let a table T1 for instance present "turnover by enterprise employee size class", a table T1.1 present "turnover by NACE and enterprise employee size class", and a table T1.2 present "turnover by enterprise employee size class and enterprise legal form". Then T1 is a sub-table of T1.1, as well as of T1.2, if all the categories of "employee size class" are identical for both tables T1.1 and T1.2. A cell of the overlap-table T1 will be a sensitive cell of T1.1 if, and only if, this cell is as well a sensitive cell of T1.2.

37. When secondary cell suppression is carried out for T1.1 and T1.2 individually, then it is not unlikely, that there will be T1 cells unsuppressed in T1.1, while in T1.2 they are complementary suppressions, and vice versa. Any user given access to both tables T1.1 and T1.2 will be able to disclose these values, and may hence be able to recalculate sensitive cells.

38. Of course there are possibilities of preventing this situation. One could protect the 'full' table T1.3: "turnover by enterprise employee size class, NACE, and enterprise legal form" and suppress in T1.1 and T1.2 any cells which also were suppressed in T1.3. Assume now, maybe due to an exceedingly fine employee size class scheme, it is not possible to protect this table within a single run, because of huge computer resource requirement. In this situation, new versions of τ -ARGUS will offer to apply a table-to-table protection procedure. Like within a backtracking procedure as described above, the software will firstly apply secondary cell suppression to e.g. table T1.1, and then to table T1.2, keeping track of any secondary suppressions in the overlap table T1. Secondary suppressions in T1, as resulting from protecting T1.1 will be treated like primary suppressions when protecting T1.2, and vice versa. The procedure will be repeated over and over again, until a step of the iteration is reached where no new secondary suppressions have been selected in T1. After the table-to-table protection procedure is finished, any cell of the overlap table T1 is either suppressed in both T1.1 and T1.2, or unsuppressed in both T1.1 and T1.2. Moreover, none of the suppressions can be disclosed making use of the additive relationship between suppressed and unsuppressed cells in either of T1.1 or T1.2.

Table-to-table protection in the context of data base query systems

39. Ideally a table-to-table protection procedure should be applied to the full set of tables ever to be published from this data source. This, however, seems less and less a realistic option. Nowadays, the process of releasing data turns to be more and more user demand driven and less pre-planned – to the extent even of providing public use data base query systems. This does yet cause serious trouble with cell suppression. The situation can be improved to some extent, when data are ‘pooled’ to keep track of suppressions in those tables which have already been published, while still others get newly created. Upcoming versions of τ -ARGUS will be capable of setting up such a ‘data pool’. One might attempt to use a data pool created as will be illustrated below as data basis for public- or scientific use data base query systems. It should be stressed here, however, that it is not at all within the scope of the project, to implement such a data base query system. Nor do we claim that users or suppliers of data base query systems will be substantially happy with the level of detail or the proportion of unsuppressed low level cells in the data pool.

40. The data pool as corresponding to a particular micro data basis will contain one and only one record for each cell of any table already protected. This record will contain an entry regarding the suppression status of the cell. When a new table has to be protected, the software will for any cell of this table investigate the data pool. Assume now, the data pool does already contain an entry for this cell. If the cell has already been used as secondary suppression in one of the tables processed earlier, then, like within a backtracking procedure, in the new table it will be treated like a primary suppression. If on the contrary the suppression status for the cell is ‘unsuppressed’ according to the data pool entry, then the software will attempt to avoid to select this cell as complementary suppression in the new table to some extent. As this will not always be possible, sometimes the user may be forced to either abandon the new table, at least part of it, or to allow for an inconsistency between the suppressions patterns of a table already published and the new table, and hence put up with a risk of disclosure.

Implementing preferences

41. Strategies like that will be implemented by ‘freezing’ the previously published cells, e.g. making them uneligible for suppression. The weaker variant - instead of ‘freezing’ these cells completely - would be to give them a low probability to be selected as secondary suppression. Facilities of that kind can be implemented by manipulating (e.g. increasing) the ‘costs’ or ‘weights’ regularly assigned to the suppression of such cells.

42. In contrast, it may sometimes seem desirable to prefer certain cells as suppressions. When for instance cells had to be included into the table because secondary cell suppression requires a table to be complete, containing e.g. the entire set of linearly interrelated cells, although one actually does not intend to publish these cells. Or when a cell, which is part of an overlap-table of a set of linked tables, is likely or well suited to be selected as secondary suppression in one of the other tables to be processed later. This would be implemented by decreasing the regular ‘costs’ assigned to the cell.

43. As another application for these ‘preferences’, imagine the situation of a table published periodically, monthly, quarterly, or annually for instance. A part of the sensitive cells may be sensitive in every period. As a simple illustration, assume a table without substructure, containing some sensitive cells, which are ‘for ever’ sensitive. Assume further, that there is more than one feasible suppression pattern, and that the costs for each pattern differ only slightly. If nothing is done, it is then very likely, that the suppression pattern changes from period to period, which might be undesirable, and also cause a risk of disclosure, when the variation in the cell values of the secondary suppressions for different periods is only small. For this example, the problem would be solvable by preferential suppression of suppressions of the previous period.

Co-ordination of suppression patterns as a special application

44. Specific problems arise when data are published on different levels of a regional classification (e.g. on the national and on the super national (EU) level, or on the regional and national levels) but secondary suppressions are to be assigned by different agencies actually (e.g. NSI’s and Eurostat, or

regional and national statistical institutes). This problem, due to decentralised organisation of official statistics within Europe, will be tackled using the ‘preference facilities’ of the software as implemented so far. Feasibility of several approaches to improve the situation will be researched with a particular view on the practicability of any methods suggested. The methods will be applied to several real-life data sets, available on national as well as regional level. Methods turning out to be promising should be supported by the software package, e.g. special options may be included in the software.

V. EXTENSION OF THE KERNEL OF τ -ARGUS

45. In order to extend τ -ARGUS, it will be necessary to improve significantly the cell suppression algorithm of the current version and alternatively to include new algorithms. As a means of optimising the information content of protected data, algorithms will be supplied supporting the release of intervals for suppressed values, and of perturbed values to replace suppressed original ones.

V.1 Excellence versus Efficiency: suppression algorithms for upcoming versions

46. In the current version, the selection of secondary suppression within τ -ARGUS is carried out using (integer) linear programming methodology as described in [2]. The method, supplied by University La Laguna, will be further improved by the team of Prof. J.J. Salazar, in order to make it applicable to hierarchical tables at reasonable expense of computing time. Due to the enormous computational burden of the method, as stated in section IV.2.1. above, it will probably be impossible even with the improved version, to apply it to large multidimensional real-life tables. Therefore, a backtracking strategy may be one of the solutions to be offered by upcoming versions, splitting the table into sub-tables and applying the linear programming procedures to the sub-tables separately, within a repeated procedure. However, this may still turn out to perform too slow for application to large sets of large multidimensional, detailed hierarchical tables, as resulting e.g. from a major economic census.

47. In order to ensure tractability also of these big applications, the package will alternatively interface with the GHQUAR hypercube method of R. D. Repsilber of the Landesamt für Datenverarbeitung und Statistik in Nordrhein-Westfalen/Germany. The method has been described in [8] and [9], for a shorter description see [4], [5] or [11]. Making use of the fact that a suppressed cell in an n-dimensional table cannot be disclosed exactly if the suppressed cell is contained in a pattern of suppressed, nonzero cells, forming the corner points of a hypercube, the algorithm evaluates hypercube suppression patterns only. In the current version² the GHQUAR algorithm divides n-dimensional tables with hierarchical structure into the corresponding set of n-dimensional sub-tables without substructure (c.f. IV.2.1). These sub-tables are protected successively within an iterative procedure, starting from the highest level. Instead of explicitly checking the feasibility of a (potential) suppression pattern through solving a set of linear problems, merely a lower bound for the amount of protection is calculated. This does of course affect the performance of the method. Sometimes a ‘better’ suppression pattern will be rejected, because this bound is beyond the threshold, even though the actual amount of protection would be sufficient. This, together with the fact that the hypercube method will sometimes miss the ‘best solution’ (due to the ‘hypercube criterion’ being a sufficient, but not a necessary criterion for a ‘safe’ suppression pattern and it thus now and then the ‘best’ suppression pattern being not a set of hypercubes), results in some tendency for over-suppression connected to this method. Experimental results (c.f. [6], Tabelle 3, p. 125,) exhibit 30 % more suppressions as compared to the ARGUS linear programming approach.

48. Depending on the results of research to be carried out by the team of Prof. J. Castro at University Politecnica Catalunya, future versions of ARGUS may offer a network flow algorithm as another alternative technique to select secondary suppressions.

² A new version of the method is currently in development, which will be able to handle tables with hierarchical substructure as a single problem without having to break it into sub-tables. ARGUS will interface with this new version as well, if it becomes available in time.

V.2 Benchmarking

49. It is certainly essential for the success of the project, that the researchers involved have a good understanding of the problems the algorithms they develop are supposed to solve – otherwise they may run into perfectly solving simplified problems, offering solutions that do not work in complex real-life situations. The most important means of properly explaining the problems to them is to let them have real-life examples to study and to test their methods. These examples should be as realistic as possible. Our idea is to transform authentic, unprotected tabular data into tabular data matching any properties of the original data relevant regarding the selection of secondary suppressions, such as the structure of the table, number and location of primary suppressions in the table, amount of protection required by the primary suppressions, number and location of zero cells. So, on the one hand these data have to be seemingly unsafe, while at the same time it must not be possible to disclose data of the original table, as published (with suppressions), even if directly compared to it. The fact that it is not at all necessary for the test data to be valid for statistical analysis, will of course ease the task of creating such data. We may literally turn apples into oranges if we please.

50. Any of the algorithms for selection of secondary suppressions developed in the course of the project will be run on the set of tables from this test library. Performances with respect to certain key issues (information loss in terms of number and/or total value of suppressions, etc., computing time requirement) will be recorded. This information will be useful for guiding internal decisions to be made during the project. Later on, it shall support the transfer of the package, supplying potential users with information on the performance of the package or of particular algorithms included, in advance of procuring it.

V.3 Table perturbation techniques

51. As stated in section III.2 above, by making use of the linear relations between published and suppressed cells, users of a published table would be able in principle to derive upper and lower bounds for the true value of any suppressed entry. The suppression procedures of τ -ARGUS will therefore ensure that no suppression pattern will be considered to be feasible, unless disclosure of all these bounds does not cause any risk of disclosure for individual respondent data. Considering this, a data disseminator protecting tabular data with ARGUS might as well publish these bounds along with the protected data. It is therefore planned to include facilities to calculate these bounds and make them available. The package will be completed along these lines by adding facilities to derive perturbed values to replace suppressed original cell entries. The perturbed values will be located between the upper and lower bounds (s.a.), matching subtotals and totals of the protected table, thus implying that the original additive relations between the cells of the table will be maintained. Algorithms addressing these issues will be developed by the team of Prof. K. Luhn at the University of Ilmenau, who, in collaboration with StBA, will also be responsible for the tasks outlined in section V.2 .

VI. FINAL REMARKS

52. The paper has suggested and illustrated the methodology to be implemented into τ -ARGUS in the course of the CASC project, as to address the objectives listed in section II. The author would however like to stress, that at the stage of writing any details concerning software concept and design of τ -ARGUS are still absolutely preliminary. In the course of the CASC project, facilities will be proposed as illustrated in this paper, but these suggestions need to be agreed upon by the CASC partners concerned. Moreover, project capacities are of course limited, concerning particularly the capacities for software development. This may result in not all the methodology being fully implemented as proposed here, even if it were appreciated by all partners. We will then seek to offer relaxed approaches.

References

- [1] Cox, L. (1981), 'Linear Sensitivity Measures in Statistical Disclosure Control', *Journal of Planning and Inference*, 5, 153 - 164, 1981
- [2] Fischetti, M. and Salazar, J.J. (1998), 'Modelling and Solving the Cell Suppression Problem for Linearly-Constrained Tabular Data', In J. Domingo-Ferrer, ed., *Statistical Data Protection '98, Conference Proceedings 25-27 March 1998, Lisbon, Portugal*
- [3] Geurts, J. (1992), 'Heuristics for Cell Suppression in Tables', working paper, Netherlands Central Bureau of Statistics.
- [4] Gießing, S. (1998), 'Looking for efficient automated secondary cell suppression systems: a software comparison', *Research in Official Statistics Journal* 2/98
- [5] Gießing, S. (1999), 'A survey on packages for automated secondary cell suppression', Federal Statistical Office of Germany, proceedings of the Eurostat/UN-ECE Work Session on Statistical Data Confidentiality 1999
- [6] Gießing, S. (1999), 'Vergleich der Software zur maschinellen Durchführung der Sekundären Geheimhaltung', In: *Forum der Bundesstatistik, Band 31/1999: Methoden zur Sicherung der Statistischen Geheimhaltung* (in German)
- [7] Jewett, R. (1993), 'Disclosure Analysis for the 1992 Economic Census. Unpublished Manuscript. Economic Statistical Methods and Programming Division, Bureau of the Census, Washington, DC.
- [8] Repsilber, R. D. (1994), 'Preservation of Confidentiality in Aggregated data', paper presented at the Second International Seminar on Statistical Confidentiality, Luxemburg, 1994
- [9] Repsilber, D. (1999), 'Das Quaderverfahren' - in *Forum der Bundesstatistik, Band 31/1999: Methoden zur Sicherung der Statistischen Geheimhaltung*, (in German)
- [10] Robertson, D. (2000), 'Improving Statistics Canada's cell suppression software (CONFID)', *Proceedings of the Compstat 2000 conference 21.-25. August, Utrecht, Netherlands*
- [11] Willenborg L., de Waal, T. (2000) 'Elements of Statistical Disclosure Control', Springer, Lecture Notes in Statistics

Appendix

Feasible upper bounds for the protection interval

Sensitivity rule	Feasible upper bound for the protection interval
(1,k)-rule	$\frac{100}{k} x_1$
(2,k)-rule	$\frac{100}{k} (x_1 + x_2)$
(n,k)-rule	$\frac{100}{k} (x_1 + x_2 + \dots + x_n)$
p%-rule	$\frac{p}{100} x_1 + (x_1 + x_2)$
(p,q)-rule	$\frac{p}{q} x_1 + (x_1 + x_2)$