A reflection on privacy and data confidentiality in Official Statistics

F. Ricciato, A. Bujnowska, A. Wirthmann, M. Hahn, E. Barredo-Capelot

EUROSTAT – Directorate B: Methodology; Dissemination; Cooperation in the ESS

Abstract:

The availability of new digital data sources represents an opportunity for Statistical Offices (SO) to complement traditional statistics and/or deliver novel statistics with improved timeliness and relevance. Nowadays SOs are part of a larger "data ecosystem" where different organizations, including public institutions and private companies, engage in the collection and processing of different kinds of (new) data about citizens, companies, goods etc. In this multi-actors scenario it is often desirable to let one organization extract some output statistics (i.e., aggregate information) from input data that are held by other organization(s) in different administrative domain(s). We refer to this problem as cross-domain statistical processing. To achieve this goal, the most intuitive approach—but not the only one—is to exchange raw input data across administrative domains (organizations). However, this strategy is not always viable when personal input data are involved, due to a combination of regulatory constraints (including lack of explicit legal basis for data sharing), business confidentiality, privacy requirements, or a combination of the above. Furthermore, new data sources often embed a much more pervasive view about individuals than traditional survey and/or administrative data, an aspect that amplifies the potential risks of data concentration. In such cases, performing cross-domain statistical processing requires technologies to elicit only the agreed-upon output information (exactly or approximately) without revealing the input data. This entails addressing two distinct but complementary problems. First, we need to compute the desired output statistics without seeing the raw input data. Second, we need to control the amount of information that might be inferred about individual data subjects in the input dataset from the output. In the field of privacy engineering the notions of "input privacy" and "output privacy" are used to refer respectively to these two problems. We remark that these problems are separable, i.e., they can be addressed with distinct tools and methods that get combined together, overlaid or juxtaposed. In this contribution we review recent advances in both fields and briefly discuss their complementary roles. As for input privacy, we provide a brief introduction to the fundamental principles of Secure Multi-Party Computation (SMPC). As for output privacy, we review recent advances in the field of Statistical Disclosure Control (SDC). Finally, we discuss possible scenarios for SMPC and SDC integration in the future "*confidentiality engineering*" setup of modern official statistics.

Keywords:
Privacy, Confidentiality, Security, Statistical Disclosure Control, Secure Multiparty Computation,

1. Introduction and motivations:

The modern society is undergoing a process of massive datafication [1]. The availability of new digital data sources represents an opportunity for Statistical Offices (SO) to complement traditional statistics and/or deliver novel statistical products with improved timeliness and relevance, so as to meet the increasing demands by users. However, such opportunities come with important challenges in almost every aspect – methodological, business models, data governance, regulatory, organizational and others. The new scenario calls for an evolution of the *modus operandi* adopted by SO also with respect to privacy and data confidentiality. We propose here a discussion framework focused on the prospective combination of advanced (dynamic) Statistical Disclosure Control (SDC) methods with Secure Multi-Party Computation (SMC) techniques.

For decades, the data business has been a natural monopoly centered around SO: no other entity had the technical and legal capability to collect and process large scale data across individuals and organizations. In the traditional operation model, illustrated in Fig. 1, the SO ingests internally all source (micro-)data that were collected either directly from the data subjects, via surveys and censuses, or indirectly through administrative registers. The input source data collected in the back-end are then processed centrally to deliver two types of front-end data in output: (i) official statistics for the general public; and (ii) more detailed data for further processing by expert users and researchers downstream the data flow.

The legal mandate of SO includes two important obligations that can be summarized as *'closed input and open output'*. On the input side (back-end) SO must preserve the confidentiality of the (micro-)data in order to protect the privacy of data subjects. On the output side (front-end) SO are committed to publish openly the processed statistics (and in general any output data), so as to ensure that all potential users get the same information and do so at the same time. The motivations and implications of both obligations are intimately connected to the democratic role of official statistics in modern society. However, in terms of real world applications, there is an unavoidable conflict between these two goals, since by definition the output data carry non-zero information about the input data (otherwise they would be useless), i.e., they always reveal *something* about the input. On the front-end, SO must determine whether what can be inferred from the output about the input can be tolerated or not, i.e., whether it is acceptable or not for the privacy of the individual data subjects. Such determination must be done case-by-case and this is the goal of the so-called Statistical Disclosure Control (SDC) function (ref. Fig. 1). When critical cases are detected, SDC methods seek to strike a reasonable compromise between the two conflicting goals of preserving accuracy and completeness of the output along with confidentiality of the input. In practice, this involves limiting and/or degrading the output data in a controlled manner. Traditionally, this was done statistically by suppression of selected elements in the output table. More recently, following the increasing demand by expert users to go beyond static tables predefined by SO and make their own statistics, be it tables or other forms of output, SDC is evolving towards dynamic models based on data perturbation, as discussed later in the paper. In general, with SDC (both static and dynamic) a trade-off is in place between accuracy and utility of the output on one hand, and confidentiality of the input on the other [2, 3], and SDC methods strive to address this problem.

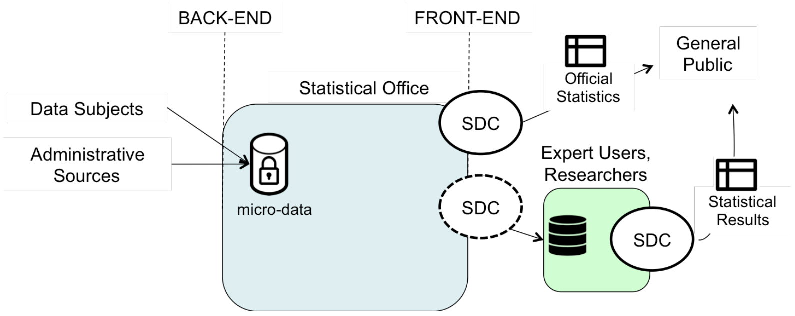

Figure 1 – Traditional scenario: the SO at the center of a data monopoly

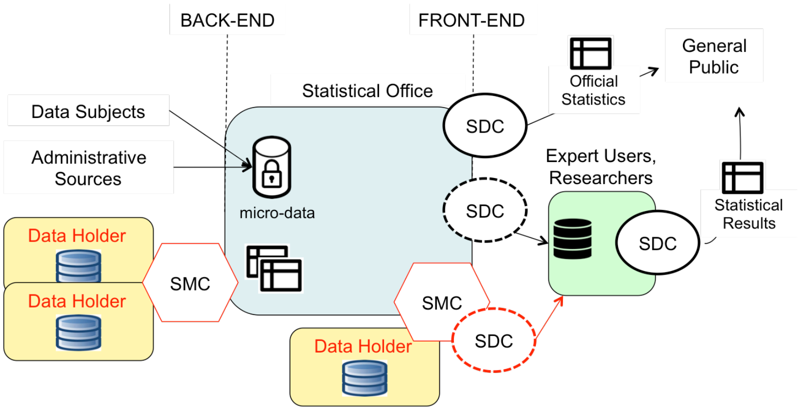

Figure 2 – New scenario: the SO as part of data ecosystem

The new scenario illustrated in Fig. 2 yields several elements of novelty. Instead of dominating the 'data monopoly' as in the past, SO are now one species of a larger 'data ecosystem' where different organizations, including public institutions and private companies, engage in the collection and processing of different kinds of data about citizens, companies, goods etc. For SO this change has implications on both sides. In the back-end, there are new potential data sources to be accessed, in addition to traditional survey/census and administrative data, but the peculiarities of such new sources might require alternative access models other than direct ingestion of raw input data [4]. On the front-end, the expert users downstream the processing flow have now increased possibilities to combine the data obtained by SO with other external data, an aspect that exacerbates the challenge for SDC.

2. Input privacy vs. Output privacy

Hereafter we provide an abstract view about the relations between input data and output data and then present the notions of 'input privacy' and 'output privacy' as introduced in the literature (see e.g. [5,10]). Finally we elaborate about how these categories map to the new scenario described above

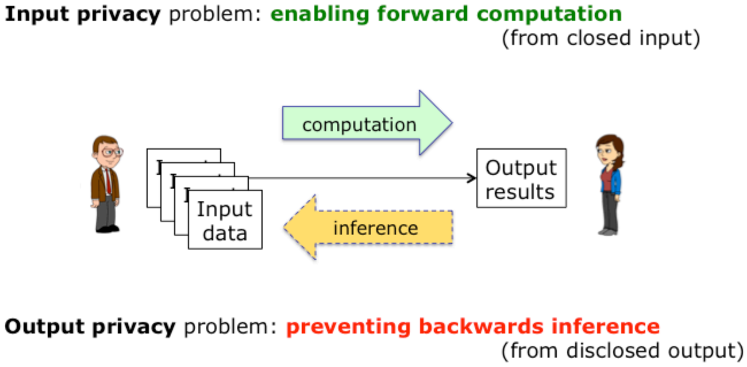
Input privacy problem: **enabling forward computation**
(from closed input)
computation
Output results
Input data
inference
Output privacy problem: **preventing backwards inference**
(from disclosed output)

Figure 3 – Input Privacy vs. Output Privacy problems.

We call *computation* the task of extracting the desired output information (or results) from a set of input data. We call *inference* the task of extracting some (partial) information about one of the input components based on knowledge of the output (and possibly other external data). It is clear from Figure 3 that computation and inference flow logically in opposite direction. In this discussion the output can take any arbitrary form, including for example a summary indicator, a set of regression coefficients or a whole frequency table, to name some concrete examples. We focus here on the case where the computation function (that may be called algorithm, methodology, procedure, program etc. in different context) is *well defined before execution*. In other words, our focus is on the stage of statistics *production*, not on the (logically antecedent) phase of data exploration and methodological development. What is relevant for our discussion is the multi-party scenario where the entity or entities (organizations, institutions, individuals, etc.) holding the input data differ from the entity/entities interested to get the output results. We shall use the terms '***input party***' (**IP** for short) and '***output party***' (**OP for short**) to refer, respectively, to the entities holding the input data and those interested to learn the output result. For example, when processing confidential data held by private business companies for official statistics, such companies take the role of IP, while the OP role is taken by SO. In another *citizen statistics* scenario, each individual respondent (data subject) can be an independent IP, and again the OP maps to SO.

Given this abstract setting, with IP and OP role, we identify two distinct confidentiality challenges:

- **Output privacy** problem: Given that the computation results will be made available in some way to the OP how to prevent OP from inferring *too much* about the input data held by IP?
- **Input privacy** problem: Given that the input data are confidential and cannot be disclosed outside their respective IP, how to enable the OP to learn the computation results?

In the particular case where a single IP holds all input data, the input privacy problem admits a very simple solution: the whole computation can be executed internally to the IP, and only the final output is passed to the OP. This has been indeed the case of official statistics for decades, with the statistical office playing the role of IP on the front-end, where the external users (including researchers and the general public) play the role of OP. In this setting, exemplified in Figure 1, the input privacy problem is inherently solved and only the output privacy problem had to be addressed.

Instead, in the new scenario depicted in Figure 2, we foresee the possibility for the statistical office to compute statistics based on confidential data held by other entities (e.g., private companies, other public institutions, or individuals) that we cannot or do not want move into the statistical office domain. In this case, external data holders play the role of IP in the back-end, where the statistical office plays the role of OP. Furthermore, on the front-end, we may want to let our users compute statistics based on the fusion of confidential data held by the statistical office with other external input data. In this case, the input privacy and output privacy problems occur jointly on the front-end.

In the new scenario, *we must cope with the input privacy problem in addition to (not in place of) the output privacy problem*. Again, if the desired statistics can be computed from the input data held by a single data holder (as IP) in isolation from other data holders, the most natural approach is to let the IP execute the computation and then pass the (final or intermediate) non-confidential data to the statistical office (as OP). Standard technical and non-technical means can be adopted to ensure that the program that is executed by the IP premises does not deviate from what was approved (or developed) by the OP. This is particular relevant on the back-end, where OP maps to the statistical office: we highlight that outsourcing the mere execution of a computation program to the IP does not imply loss of control by the OP over *what program* is executed.

The input privacy problem is more challenging when the required output results are based on the contribution of several input data sets held by multiple IPs that cannot disclose their data. In some cases, the computation program can be factorized into separate computation instances that are run independently by the multiple IPs, either sequentially or in parallel. However, very often the desired output results do not allow for computation factorization. For example, this is the case when output results must be computed on the intersection records between different IP data sets, or when a regression must be run over variables that are held by different IPs. In these cases, we may resort to Secure Multi-Party Computation (SMC).

3. Solution approaches to Output Privacy problem

The output privacy approach is traditionally addressed by so-called Statistical Disclosure Control (SDC) techniques, possibly in combination with Access Control (AC). SDC aims at restricting *what* is disclosed, while AC imposes restrictions on *to whom* it is disclosed. Generally speaking, there is a trade-off between the two: the weakest SDC requires strongest AC, and vice-versa.
AC methods rely on combination of requirements referring to the nature of the potential users, their experience with holding confidential data and legitimate use of the data. Potential users must provide evidence of fulfilling AC requirements which is scrutinized by the data owners. The trustworthy users are confined with more detailed data and better access facilities. SDC methods rely on combination of suppression, perturbation, randomization and aggregation of data.

Historically, SDC was performed manually by dedicated experts, following practices and criteria that were developed through the years in the official statistics community. SO successfully managed the output control as the statistics going out were pre-defined and SO could consistently apply suppressions on primary and secondary confidential cells. The current trend is towards "on-the-fly SDC". Nowadays many users want to calculate their own tailor-made statistics on the basis of the detailed data sources. In response to these needs SO make available dynamic data querying systems that implement modern SDC approaches, addressing in particular the problem of differential disclosure. These new SDC approaches require that the output is always safe, also in combination with any other statistics based on the same source. The random noise protection method developed by the Australian Bureau of Statistics (ABS) is an example of modern SDC approach [11, 12]. The ABS method consists in applying small perturbations (controlled noise) to the data with the predefined probability distribution. A specific pseudo-random mechanism called "cell key method" is adopted to ensure that the injected perturbations are consistent across multiple queries. This approach is robust to differential attacks that, instead, represent the main limitation of pure randomized systems (where noise varies across queries). The cell key method is recommended for protection of European census 2021 round [13]. It is expected that it will ensure consistent protection of the census data in view of making them available via various channels and access systems.

4. Solution approaches to Input Privacy problem

When the input data are held by multiple IPs, and the computation cannot be factorized into independent (sequential or parallel) components, one possible solution approach to the input privacy problem is given by Secure Multi-Party Computation (SMC) methods based on the principle of *secret sharing*. In a nutshell, with SMC every individual input data element is transformed into a set of so-called *secret shares* that are passed to a set of (three or more) intermediate '*computing parties'* (CP). The CPs form collectively the SMC infrastructure. The secret shares are produced in a way that yields two important properties. First, under certain conditions, defined by the applicable attack model, secret shares do not reveal anything about the input source data to the individual CPs (non-invertibility). Second, they allow to compute *exactly* the correct output that would be obtained by a direct computation on the clear input (homomorphism). A general introduction to SMC and secret sharing can be found in [6] while examples of practical applications[1] are found in [7, 8].

To preserve confidentiality, each CP must not disclose the received secret shares to other CPs, i.e., CPs must not collude among themselves to break the confidentiality of IP data. SMC can be tuned to be robust against a subset of colluding CPs. In other words, the system preserves input confidentiality as far *as at least one CP does not collude with the others*. That means, the *CPs must be trusted collectively, not individually*. Then the problem of ensuring confidentiality moves up to an institutional level, and translates into the task of identifying a suitable set of CP. An important property of SMC plays in our favour: in practical deployment, the same institution can play multiple roles. For example, one data holder serving as IP can at the same time host one CP instance – obviously he would never collude with other CPs against himself. Also, one entity (e.g., the SO) can play contemporarily the roles of IP, OP and CP.

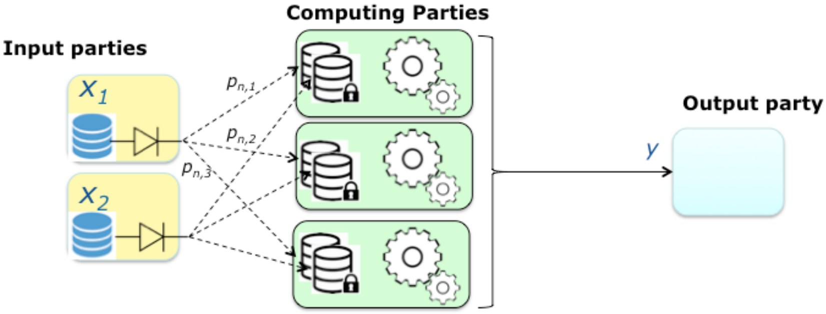

Figure 4 - Principle of operation of SMC (secret sharing)

In theory, any arbitrary function can be computed via SMC, at the cost of increased computation load and communication overhead between the CPs compared to a plain centralized computation. The cost of SMC translates into longer computation time and/or more hardware/bandwidth resources. The cost increase factor might be substantial, but still acceptable for most practical applications.

When the computation cost and/or delay of SMC is too large, we may resort to an alternative solution, hinted hereafter. The key point of both solutions is to let the set of relevant stakeholders (any combination of IP, OP and/or external entities) to exert *shared control* over the computation process, so as to ensure jointly that no confidential data is disclosed except the agreed-upon final results. Such

[1] The relationship between SMC and personal data protection legislation presents some open issues that go beyond the scope of the present contribution, see e.g. the discussion in [15].

guaranteed can be delivered, in principle, by a special computation machine that is built (at both hardware and software levels) to execute exclusively code that is cryptographically authenticated by all and only the intended stakeholders, as depicted in Figure 5. Such ideal machine can be built by combining so-called Trusted Execution Environment (TEE) technology with cryptographic solutions for multi-party control (MPC). The TEE technology [14] was developed recently to address the emerging need in cloud computing applications to decouple, also at the hardware level, the physical operation of the computing machine (hosting, powering up, general maintenance) from the control of what is executed over that machine.

The MPC-TEE solution should be distinguished by the simplistic approach of relying on a Trusted Third Party (TTP). The trust models underlying the two settings are completely opposite to each other, as exemplified by the diagrams in Figure 6. The TTP represents an *independent* entity, *outside* the control of all stakeholders. On the contrary, MPC-TEE can be thought as a *dependent* entity that is *under direct control of all stakeholders jointly*. In other words, *full delegation* takes place with TTP, while *no delegation* take place with MPC-TEE.

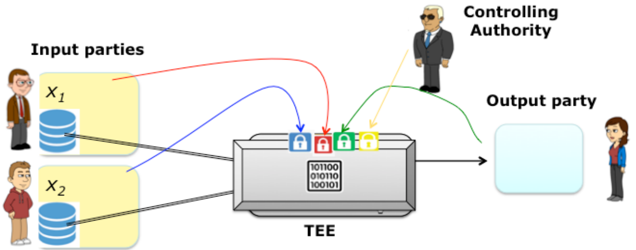

Figure 5 – Concept of Multi-Party Controlled Trusted Execution Environment (MPC-TEE)

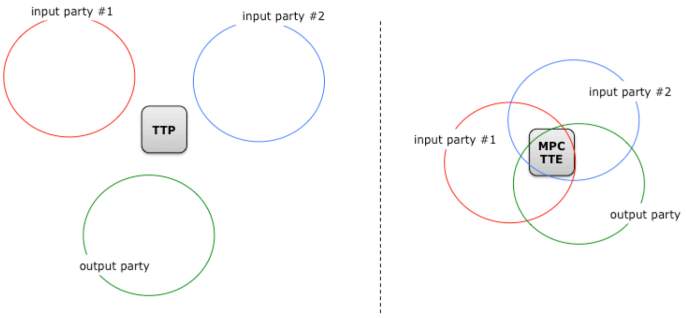

Figure 6 – Logic difference between the TTP model (independent entity, full delegation) and the MPC-TTE model (jointly dependent entity, no delegation). The latter model applies also to SMPC.

5. Conclusion and discussion

Since SDC and SMC are targeting different but complementary problems, it is natural to consider their combination. From the above definitions it should be clear that, in principle, both ‘input privacy’ and ‘output privacy’ problems might be encountered at both sides. In other words, in the new scenario we may consider adopting some combination of SMC and SDC in the back-end as well as on the front-end.

As to the back-end, SMC may play an important role when joint processing of multiple data sources from different parties is required but direct ingestion of raw input data by SO is not possible, e.g., due to legal restrictions or business considerations (as in [7]). This includes cases where the source data are held by the private business sector. Considering the special trust endowment of SO, who plays the role of OP in the back-end, it is reasonable to assume that non-disclosure agreements and legal provisions are sufficient to solve the ‘output privacy’ problem in the back-end, waiving the need to introduce SDC tools on this side.

Conversely, SDC solutions will remain crucial on the front-end. SMC can be used on the front-end to enable joint processing of confidential input data from SO and other data holders (ref. rightmost part of Fig. 1(b)). More in general, a wise combination of SMC and SDC might help to achieve a higher level of overall confidentiality in the new wilder scenario, where increased availability of external data sources amplifies the non-disclosure challenges. Some initial work in this direction is starting to appear in the field of Official Statistics [9], while commercial implementation of SMC already include some simple safeguards for disclosure control [8].

In conclusion, the new *datafied* scenario requires SOs to widen their traditional approach to privacy and data confidentiality. Purely regulatory means in the back-end and simple SDC methods in the front-end might not suffice any more. Embracing novel tools such as SMC, in combination with more advanced forms of dynamic SDC, seems to be a promising direction to move forward. More in general, SO need to develop a more systematic and articulated approach towards *confidentiality engineering* to face the new challenges posed by an increasingly complex data ecosystem.

References:

1. K. Cukier and V. Mayer-Schoenberger. The rise of big data. Foreign Affairs, May/June 2013. URL https://www.foreignaffairs.com/articles/2013-04-03/rise-big-data.
2. J. M. Abowd and I. M. Schmutte. An economic analysis of privacy protection and statistical accuracy as social choices. American Economic Review (forthcoming), August 2018. https://arxiv.org/abs/1808.06303.
3. J. Domingo-Ferrer, S. Ricci, and J. Soria-Comas. A methodology to compare anonymization methods regarding their risk-utility trade-off. Int. Conf. on Modeling Decisions for Artificial Intelligence (MDAI 2017), August 2017.
4. F. Ricciato et al. Towards a Reference Architecture for Trusted Smart Statistics. 104th DGINS conference, Bucharest, Romania, 10-11 October 2018. https://tinyurl.com/yco77y62.
5. T. Wang and L. Liu. Output privacy in data mining. ACM Transactions on Database Systems, 36(1), March 2011. DOI: 0.1145/1929934.1929935.
6. R. Cramer, I. Damgard, and J. Buus Nielsen. Secure Multiparty Computation and Secret Sharing. Cambridge University Press, 2015.
7. S. Anspal, M. Kaska, and I. Seppo. Using k-anonymization for registry data: pitfalls and alternatives. IEEE Trans. on Dependable and Secure Computing, 2017. http://dx.doi.org/10.12697/ACUTM.2017.21.05.
8. D. Bogdanov et al. Rmind: a tool for cryptographically secure statistical analysis. IEEE Trans. on Dependable and Secure Computing, 15(3), May/June 2018.
9. K. Shirakawa et al. A proposal of a simple and secure statistical processing system using secret sharing. UNECE Work Session on Statistical Data Confidentiality, Skopje, 20-22 Sept. 2017. www.unece.org/fileadmin/DAM/stats/documents/ece/ces/ge.46/2017/6_secret_sharing.pdf.
10. UN Handbook on Privacy-Preserving Computation Techniques, April 2019. Available online at http://tinyurl.com/y4do5he4.
11. L Leaver, Victoria and Marley, Jennifer K. (2011) A Method for Confidentialising User-Defined Tables: Statistical Properties and a Risk-Utility Analysis; ISI conference, Dublin 2011.
12. G. Thompson et al (2013) Methodology for the Automatic Confidentialisation of Statistical Outputs from Remote Servers at the Australian Bureau of Statistics, UNECE Works Session on Statistical Confidentiality, Ottawa, Canada, 20-23 October 2013.
13. P.P. de Wolf, Perturbative methods for ESS census tables, conference on New Techniques and Technologies for Statistics, Brussels, March 2019.
14. M. Sabt, M. Achemlal, A. Bouabdallah, Trusted Execution Environment: What It is, and What It is Not. 2015 IEEE Trustcom/BigDataSE/ISPA. DOI 10.1109/Trustcom.2015.357.
15. G. Spindler and P. Schmechel, Personal Data and Encryption in the European General Data Protection Regulation, (2016) JIPITEC 163 para 1. Online: https://www.jipitec.eu/issues/jipitec-7-2-2016/4440/spindler_schmechel_gdpr_encryption_jipitec_7_2_2016_163.pdf