Knowledge and Information Systems manuscript No.
(will be inserted by the editor)

Outsourcing Analyses on Privacy-Protected Multivariate Categorical Data Stored in Untrusted Clouds

Josep Domingo-Ferrer · David Sánchez · Sara Ricci · Mónica Muñoz-Batista

Received: date / Accepted: date

Abstract Outsourcing data storage and computation to the cloud is appealing due to the cost savings it entails. However, when the data to be outsourced contain private information, appropriate protection mechanisms should be implemented by the data controller. Data splitting, which consists of fragmenting the data and storing them in separate clouds for the sake of privacy preservation, is an interesting alternative to encryption in terms of flexibility and efficiency. However, multivariate analyses on data split among various clouds are challenging, and they are even harder when data are nominal categorical (i.e., textual, non-ordinal), because the standard arithmetic operators cannot be used. In this article, we tackle the problem of outsourcing multivariate analyses on nominal data split over several honest-but-curious clouds. Specifically, we propose several secure protocols to outsource to multiple clouds the computation of a variety of multivariate analyses on nominal categorical data (frequency-based and semantic-based). Our protocols have been designed to outsource as much workload as possible to the clouds, in order to retain the cost-saving benefits of cloud computing while ensuring that the outsourced stay split and hence privacy-protected versus the clouds. The experiments we report on the Amazon cloud service show that by using our protocols the controller can save nearly all the runtime because it can integrate partial results received from the clouds with very little computation.

Keywords Cloud computing · Data privacy · Data splitting · Nominal data

J. Domingo-Ferrer · D. Sánchez · M. Muñoz-Batista
Universitat Rovira i Virgili, Dept. of Computer Science and Mathematics, UNESCO Chair in Data Privacy, CYBERCAT-Center for Cybersecurity Research of Catalonia, Av. Països Catalans 26, 43007 Tarragona, Catalonia.
E-mail: {josep.domingo@urv.cat,david.sanchez}@urv.cat, monicuba@gmail.com

S. Ricci
Brno University of Technology, Dept. of Telecommunications, Technická 3058/10, 61600 Brno, Czech Republic
E-mail: ricci@feec.vutbr.cz

1 Introduction

Statistical analyses involve collecting and investigating (potentially large) data samples. In turn, collecting and investigating data requires storing them and computing on them. Among the usual analyses, measuring the dependence between attributes in multivariate data sets is one of the costliest operations. For instance, measuring the correlation between (just) two categorical attributes in a data set containing one million records may require computing and storing matrices of size one million times one million [36]. If matrix values are as short as 4-byte integers (real numbers would take more), then storing one matrix alone takes nearly 4 terabytes.

Coping with such huge data amounts is often infeasible for data controllers. In this scenario, outsourcing storage and computation to the cloud is an attractive alternative because of the large, cheap and highly scalable resources it offers. Nevertheless, when the data to be outsourced contain sensitive information (e.g., personal information, clinical outcomes, etc.), controllers may refrain from embracing the cloud due to privacy concerns [4] or privacy regulations [20]. The problem is not only that the cloud service providers (CSPs) may read, use or even sell the data outsourced by their customers, but also that CSPs may suffer attacks or leakages that can compromise data confidentiality.

To mitigate the above concerns, privacy-preserving methods for storing and processing the data outsourced

to the cloud should be designed. This has been the main goal of the European project CLARUS [12] in which the current work is framed. CLARUS consists in a proxy located in a domain trusted by the controller that implements security and privacy-enabling features towards the CSP. Among other protection techniques, CLARUS relies on *vertical data splitting*: fragments consisting in projections of the data set on subsets of attributes are distributed to different CSPs. Specifically, data are partitioned into several fragments (so that leakage of any single fragment does not entail a significant privacy risk), and each fragment is stored in *clear* form in a different CSP so that attributes in different fragments cannot be linked, see [1] and [11]. Since fragments are stored in the clear, splitting allows efficient computations while protecting privacy. For this reason, splitting is an interesting alternative to encryption-based methods (e.g., searchable or homomorphic encryption) in terms of flexibility and efficiency, because computing on encrypted data is extremely limited and costly [34].

However, performing many statistical analyses, such as data dependence or correlation assessment, requires using the whole data set or at any rate more than a single fragment. With split data, the problem is for the controller to manage as effortlessly as possible the fragments stored at different untrusted CSPs to conduct such computations while ensuring that attributes in different fragments cannot be linked by the CSPs [9, 47]. The problem is even more challenging when dealing with nominal categorical data, i.e., data whose attribute values are noun-phrases corresponding to jobs, interests, conditions, etc. These data, that are textual and non-ordinal and on which the standard arithmetic operators cannot be used, account for most of the personal information currently being collected (e.g., in social networks, B2C transactions, etc.) [48]. In particular, accurately measuring the dependence or correlation between nominal attributes requires semantically grounded techniques [37] that are costly, both in computational power and storage.

Contribution and plan of this article

In [9,14], we evaluated several non-cryptographic proposals for statistical computation (basically correlations) on *numerical* split data. We explored application to *categorical nominal* data in the conference paper [36]. The present article describes a full approach to privacy-protected outsourcing of storage and computation on nominal categorical split data. We also include detailed performance analyses.

The novel contributions of this paper can be summarized as follows:

- We propose efficient protocols to securely compute statistical dependence analyses on split outsourced data for a variety of methods, encompassing frequency-based and semantic-based tests. In all cases, the goal of the protocols is to outsource as much workload as possible to the cloud, while ensuring that confidential data are not leaked to the CSPs.
- We show how frequency-based tests (χ^2-test, ANOVA or Cramér's V) can be efficiently derived from contingency table computation. Our protocols compute the contingency table on the cloud side and then transfer it to the local side, i.e. the controller's side. After that, the frequency-based tests can be easily computed locally, because they are not computationally demanding.
- Semantic-based tests, which were partially treated in [36], are analyzed in greater detail. Specifically, the computational costs of several semantic measures to quantify distances between nominal values are assessed, both theoretically and empirically. Note that semantic-based tests are the costliest ones and, hence, those that can benefit the most from outsourcing the computation to the cloud.
- Following the approach used in [36], we show how statistical dependence analyses can be performed on split data outsourced to separate clouds by relying on secure scalar product protocols.
- We also report on empirical work on Amazon Web Services cloud instances. Performance figures show that our protocols are able to outsource most of the workload to the CSPs, thereby reconciling data privacy with the cost-saving benefits of the cloud.

The rest of this article is organized as follows. Section 2 reviews related work on outsourcing data to the cloud. Section 3 surveys multivariate analyses on nominal data and reviews frequency-based tests and semantic-based tests. Section 4 presents the CLARUS architecture and the security assumptions considered in the design of our protocols. Section 5 discusses the advantages of vertical splitting for privacy-aware outsourcing of storage and computation, and then recalls two secure scalar product protocols used in the rest of the paper. After that, we propose protocols to outsource in a privacy-protected manner the computation of multivariate frequency-based tests (Section 6) and semantic tests based on the semantic-distance covariance (Section 7). These protocols decompose the multivariate analyses being considered into several secure scalar products that can be securely computed on the cloud side. Section 8 reports the experimental results obtained when implementing our protocols for the costliest analysis (the semantic-based test) and compares the workload savings against a local computation by the controller.

Section 9 contains the conclusions and lists some future research lines. Finally, there are two appendices. Appendix A details the calculation of semantic distances in semantic-based tests with three types of measures: edge-counting, feature-based and information content-based. Appendix B justifies the security of the two secure scalar product protocols used.

2 Related work

There is a sizeable body of literature devoted to outsourcing matrix and polynomial computations, depicted in Table 1. However, most of it deals with numerical data.

Many contributions require that the server(s) only see encrypted versions or shares of the actual data. In [5], a client securely outsources algebraic computations to one or several remote servers, in such a way that the server learns nothing about the client's private input or the result of the computation. This scheme is based on multiparty secure computation via secret sharing. In [26] and [27], a client outsources a matrix inversion and a large matrix multiplication, respectively, to an untrusted cloud, so that the cloud does not learn either the original or the resulting matrices. In this case, the original data are multiplied by two permutation matrices as encryption procedure. The more recent contribution [44] follows the same line (outsourcing polynomials and matrix computations), but it focuses on public verifiability of the computation (any third party can verify its correctness, and not just the client as in the previous proposals). This comes at the price of using more complex cryptographic schemes.

Hybrid schemes mixing vertical splitting and cryptography are an alternative. For the sake of privacy, in [29] symmetric homomorphic encryption is carried out on vertically partitioned databases. However, the use of homomorphic encryption limits the possible computations that can be performed on the encrypted data. In [52], a medical data set is vertically split into three tables. The contents of each table stay in the clear and can be encrypted or anonymized depending on their sensitivity level. The original data set can be recovered partially or entirely depending on the level of authorization of the user. The cloud is used as a secure repository and to generate the different tables; statistical analyses can be performed by the users in their local computing facilities.

Other proposals use the cloud as a repository of encrypted files. In particular, the data are encrypted and labeled with keywords, which allow different users to query on encrypted data. In [51], a logarithmic-time ranked search over encrypted documents outsourced to a cloud is presented. In [28], the data are encrypted using a symmetric cryptosystem, and then a variant of the k-nearest-neighbors algorithm, which can compute the Euclidean distance between two encrypted vectors, is used to perform searchable encryption. In [19], a content-based search scheme is designed that can find the semantic relationship between concepts in the encrypted datasets. It considers the similarity among concepts belonging to one attribute.

However, searchable encryption normally works on a collection of documents rather than on a numerical/categorical data set. The use of this technique to compute on a categorical data set would require labeling any single value in the data set and, therefore, the set of labels would be as large as the data set. In fact, if the whole data set is encrypted and labeled with only one keyword or only some parts of it are labelled, multivariate statistical analyses of its contents are not supported.

Outsourcing matrix computations where the server computes on additively split matrices rather than encrypted matrices is considered in [31]. Even though no encryption is used, the split versions of the matrices seen by the server do not preserve any of the statistical features of the original data (they look gibberish), so that no direct exploratory analyses can be performed on them.

A substantial difference between our proposals in this paper and the previous literature is that the outsourced data preserve the utility of the original data. Therefore, our approach allows performing direct exploratory analyses.

3 Multivariate analyses on categorical data

When data are numerical, multivariate statistical analyses such as correlations, covariances, regressions and classifications are easy to perform and can be computed using standard arithmetic operators. In contrast, analyzing categorical data is more difficult. Especially challenging are nominal categorical attributes, whose values are noun-phrases describing jobs, interests or conditions, etc., because they are textual and non-ordinal and, therefore, require specific analytical methods. The difficulty of computing on nominal data has been addressed in the literature; examples can be found in [21, 46, 53] which focus on nominal and mixed data.

3.1 Frequency-based tests

The simplest methods rely on the frequencies of attribute values. Well-known frequency-based procedures

Table 1 Main features of related work on outsourcing computation: reference, data type, cloud security model, outsourcing method, outsourced computation, experimental results, pros and cons. "N" stands for numerical, "C" for categorical, "file" for document, "H-but-C" for honest-but-curious.

Reference	Data	Clouds	Methods	Computation	Exp.	Pros & Cons
Atallah et al [5]	N	H-but-C, (t,n)-collision resistant	data split by secret sharing	matrix product	NO	hidden data, no flexibility
Lei et al [26]	N	malicious	data permuted and masked	matrix inversion	YES	hidden data, cheating resistant, no flexibility
Lei et al [27]	N	malicious	data permuted and masked	matrix product	YES	hidden data, cheating resistant, no flexibility
Sun et al [44]	N	malicious	data encrypted	verifiability	NO	encrypted data, cheating resistant, no flexibility
Li et al [29]	N+C	H-but-C	data vertically split and encrypted	secure comparison problem	YES	encrypted data, no flexibility
Yang et al [52]	N+C	H-but-C	data vert. split, plain or encrypted or masked	query on plain/ masked data	YES	computation on original data on user side
Xia et al [51]	file	H-but-C	data encrypted	searchable encryption	YES	encrypted data, no flexibility
Li et al [28]	file	H-but-C	data encrypted	searchable encryption	YES	encrypted data, no flexibility
Fu et al [19]	C	H-but-C	data encrypted	searchable encryption	YES	encrypted data, no flexibility
Nassar et al [31]	N	H-but-C	data additively split	matrix product and inversion	NO	hidden data, no flexibility

to measure the statistical dependence between two categorical attributes are the χ^2-test of independence [2], ANOVA [22] and Cramér's V [2]. These tests use the contingency tables associated with categorical attributes as input for a linear regression analysis.

In particular, a *contingency table* (or cross-classification table) is a table containing the (multivariate) frequency distributions of the nominal attributes. Let $\mathbf{a}$ and $\mathbf{b}$ denote two nominal attributes, $\mathbf{a}$ with h categories $c_1(\mathbf{a}), \ldots, c_h(\mathbf{a})$ and $\mathbf{b}$ with k categories $c_1(\mathbf{b}), \ldots, c_k(\mathbf{b})$. The contingency table has h rows and k columns displaying the sample frequency counts of the $h \times k$ category combinations.

3.2 Semantic-based tests

Even if frequency-based methods can measure some degree of statistical dependence, they only consider the similarities between the distributions of categorical labels; therefore, they fail to capture the *semantic* similarity among the categories themselves, which is the means by which human beings create, understand and manage nominal data.

To tackle this issue, semantically grounded methods have been recently proposed. In [15], nominal values are assigned numbers that capture both their semantic and distributional features; from these, semantically coherent variances [42] and correlations can be computed based on standard numerical methods. In [45], a more accurate way to measure the dependence between categorical attributes is proposed. Specifically, the *distance covariance* and the *distance correlation* measures are proposed as alternatives to numerical covariance and correlation, respectively. Numerical covariance requires the values of attributes to be totally ordered, and it measures dependence by checking whether greater values of one attribute correspond to greater values of the other attribute, and smaller values to smaller values. This assumption does not work for non-ordinal (i.e., nominal) categorical attributes, which lack total order. In contrast, the distance covariance quantifies to what extent the two attributes are independently dispersed, where dispersion is measured according to the pairwise distances between all pairs of values of each attribute. Unlike frequency-based approaches, pairwise distances can capture the *semantics* inherent to categorical values. To do so, the pairwise distance can be calculated using similarity/distance measures [37], that quantify how similar the meanings of the concepts associated with the categorical values are, based on the semantic evidences gathered from one or several knowledge sources (e.g., ontologies, corpora).

Following the approach presented in [37], let $\mathbf{x}^1 = (x_1^1, \ldots, x_n^1)^T$ and $\mathbf{x}^2 = (x_1^2, \ldots, x_n^2)^T$ be vectors of values of two nominal attributes. The calculation of the distance covariance requires measuring the pairwise semantic distance between the nominal values of each attribute. The *semantic-distance matrix* of $\mathbf{x}^1$ is given by

$$\mathbf{X}^1 = [x_{ij}^1]_{i,j \leq n}, \tag{1}$$

where $x_{ij}^1 = |x_i^1 - x_j^1|$ are the semantic distances between two nominal values of the same attribute $\mathbf{x}^j$ (see Appendix A for more details). Similarly, we define $\mathbf{X}^2 = [x_{ij}^2]_{i,j \leq n}$, where $x_{ij}^2 = |x_i^2 - x_j^2|$. Then, the *double-centered matrix* $\hat{\mathbf{X}}^1$ is computed, whose elements are obtained as

$$\hat{X}^1_{kl} = x^1_{kl} - \bar{x}^1_{k\cdot} - \bar{x}^1_{\cdot l} + \bar{x}^1_{\cdot\cdot} \quad \text{for } k,l = 1,\ldots,n, \tag{2}$$

and where

$$\bar{x}^1_{k\cdot} = \frac{1}{n}\sum_{l=1}^{n} x^1_{kl}, \quad \bar{x}^1_{\cdot l} = \frac{1}{n}\sum_{k=1}^{n} x^1_{kl}, \quad \bar{x}^1_{\cdot\cdot} = \frac{1}{n^2}\sum_{k,l=1}^{n} x^1_{kl}. \tag{3}$$

Similarly, let us define $\hat{X}^2_{kl} = x^2_{kl} - \bar{x}^2_{k\cdot} - \bar{x}^2_{\cdot l} + \bar{x}^2_{\cdot\cdot}$ for $k,l = 1,\ldots,n$.

Definition 1 The squared *semantic-distance covariance* is obtained as the arithmetic average of the products $X^1_{kl}X^2_{kl}$, that is,

$$d\mathcal{V}^2_n(\mathbf{x}^1,\mathbf{x}^2) = \frac{1}{n^2}\sum_{k,l=1}^{n} X^1_{kl}X^2_{kl}, \tag{4}$$

and the squared *semantic-distance variance* is obtained as

$$d\mathcal{V}^2_n(\mathbf{x}^1) = d\mathcal{V}^2_n(\mathbf{x}^1,\mathbf{x}^1) = \frac{1}{n^2}\sum_{k,l=1}^{n} X^1_{kl}X^1_{kl}. \tag{5}$$

See [45] and [37] for details and justification of the above definition.

In general, if $\mathbf{X} = (\mathbf{x}^1,\ldots,\mathbf{x}^m)$ is a data set with m attributes $\mathbf{x}^j$, $j = 1,\ldots,m$, the distance covariance matrix $\hat{\boldsymbol{\Sigma}}$ of $\mathbf{X}$ is

$$\hat{\boldsymbol{\Sigma}} = \left(\begin{array}{c|c|c|c} d\mathcal{V}_n(\mathbf{x}^1) & d\mathcal{V}_n(\mathbf{x}^1,\mathbf{x}^2) & \cdots & d\mathcal{V}_n(\mathbf{x}^1,\mathbf{x}^m) \\ \hline d\mathcal{V}_n(\mathbf{x}^2,\mathbf{x}^1) & d\mathcal{V}_n(\mathbf{x}^2) & \cdots & d\mathcal{V}_n(\mathbf{x}^2,\mathbf{x}^m) \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline d\mathcal{V}_n(\mathbf{x}^m,\mathbf{x}^1) & d\mathcal{V}_n(\mathbf{x}^m,\mathbf{x}^2) & \cdots & d\mathcal{V}_n(\mathbf{x}^m) \end{array}\right).$$

The method we have just recalled accurately captures the (semantic) dependence between nominal attributes; however, its main drawback is cost. Due to the need to compute pairwise distance matrices, the calculation of the distance covariance between two nominal attributes has quadratic cost, both in time and storage. Moreover, as we discuss in Appendix A, evaluating the semantic distance between each pair of nominal attribute values adds a significant burden. Specifically, we consider three types of ontology-based semantic distances:

1. Edge-counting measures, whose cost is $O(D)$, where D is the depth of the ontology.
2. Feature-based measures, whose cost is $O(S)$, where S is the maximum number of ancestors of a concept in the ontology (equal to its depth D if there is no multiple inheritance).
3. Measures based on information content, whose cost is $O(C+D)$, where C is the total number of concepts in the ontology. These are the costliest measures, but as argued in Appendix A they are also the most accurate.

Thus, whatever the semantic distance considered, the cost of assessing semantic dependences makes it attractive for data practitioners to outsource the calculation to the cloud. However, when the data are sensitive (which is often the case because most of the personal attributes gathered on individuals are nominal [48]), outsourcing storage and calculation should be performed in a privacy-preserving way. This is precisely the main goal of the present article.

4 System architecture and assumptions

The scenario we consider involves the three entities in Figure 1: the data controller, the CLARUS proxy and the CSPs. The controller owns the data that need to be outsourced to the CSP. CLARUS is a proxy located in a domain trusted by the controller that implements security and privacy-enabling features towards the cloud service provider so that i) the CSP only receives privacy-protected versions of the controller's (or the controller's users') data, ii) CLARUS makes access to such data transparent to the controller's users (by adapting their queries and reconstructing the results retrieved from the cloud) and iii) it remains possible for the users to leverage the cloud to perform accurate computations on the outsourced data without downloading them.

CLARUS may outsource data either in separate cloud accounts within the same CSP (see left side of Figure 1) or to different clouds, each one run by a different CSP (see right side of Figure 1). The CSPs that receive privacy-protected versions of the controller's data are not trusted and, hence, they should not be given access to the entire original data set. CSPs are considered *honest-but-curious*: they will properly fill their role in the communication and computation protocols but they may gather and analyze the data they store and the messages they receive. We also assume that CSPs do not share information between them, i.e., they do not collude to aggregate their data.

The *raison d'être* of this architecture is to outsource as much storage and computation as possible to the cloud in a privacy-preserving manner, while keeping the workload of the CLARUS proxy (which sits in the controller's premises) as low as possible. The privacy-preserving calculation protocols implemented by CLARUS should, thus, follow this principle. The underlying

assumption is that computing in the cloud is cheaper and/or more convenient than computing in the controller's local facilities.

5 Vertical splitting for privacy-protected storage and computation outsourcing

A privacy breach occurs when individuals are re-identified in a data set containing confidential attributes. Re-identification may be enabled by single attributes (like SS numbers) or by sets of attributes each of which does not uniquely identify the individual to whom the record corresponds but whose combination may (e.g., job+age+place of birth). The former type of attributes are known as *identifiers*, whereas the latter are called *quasi-identifiers*.

If a data set is to be released without splitting, identifiers should be removed or at the very least encrypted or pseudonymized, whereas quasi-identifiers should be masked (by either perturbing them or reducing their detail); see [24] for details. Thus, information is lost in this process (removed identifiers, reduction of detail or perturbation of quasi-identifiers).

However, if the data set is protected by splitting it among several untrusted clouds, no information loss needs to be incurred. Under *vertical data splitting*, identifiers may be fragmented (e.g. locations can be split into longitude and latitude, SS numbers may be fragmented into several subgroups of digits, etc.) and the set of quasi-identifier attributes may also be fragmented into several disjoint subsets; this fragmentation of identifiers and quasi-identifiers should be such that no re-identification is feasible from a single fragment [39]. Then fragments (either split values or attribute subsets) are stored in separate locations (i.e., different CSPs or cloud accounts).

Since re-identification is not possible from single fragments, these can be stored in clear form. Thanks to this feature, data splitting allows fast additions and updates of the outsourced data, provided that the local proxy in charge of orchestrating the splitting process keeps track of the criteria employed to fragment the data and the locations at which fragments were stored [39].

In contrast, when protecting data by masking instead of splitting, adding or updating a value typically requires re-encrypting or re-masking the entire data set or in any case larger chunks than just the value that has been added/updated. For example, if data are protected according to the k-anonymity privacy model by generalizing quasi-identifiers [38] or microaggregating them [16], adding or updating an original record may require k-anonymizing again the entire data set. Furthermore, the CSP storing the anonymized data set might be able to infer the value of some original records by comparing the successive anonymized versions of the data set; thus, splitting may afford more protection than masking as long as the various CSPs holding fragments do not collude.

Similar issues arise if protecting data using functionality-preserving encryption: to update a single value, the data controller has to (1) retrieve the entire data set from the cloud, (2) decrypt, update and re-encrypt it and, (3) send back the entire encrypted data to the cloud. Therefore, data splitting is significantly more efficient for additions and updates than encryption methods, such as searchable or homomorphic encryption [39]. Regarding data processing on the cloud, even though searchable and homomorphic encryption allow performing some operations on ciphertext [18], computing on encrypted data is extremely limited and costly [34], and it requires careful management of encryption keys. Outsourced data processing can be performed much more efficiently on split data: although each CSP only holds partial data, these are in the clear. Admittedly, both the orchestration of the split calculations to be done and the aggregation of the partial results retrieved from each CSPs should be done by the local proxy; therefore, computation protocols should be designed to minimize both the data that need to be locally stored and the amount of local calculations needed to obtain the final result.

In vertical splitting, analyses that involve single attributes (e.g., mean, variance) or attributes stored within a single data fragment are fast and straightforward: the cloud storing the fragment can independently compute and send the output of the analysis to the local proxy. However, analyses assessing the relationship (e.g., correlation) between attributes may involve fragments stored in different locations, and thus, communication between several clouds. As shown in [9,14], performing calculations on data split among multiple clouds can be decomposed into several secure scalar products to be conducted between pairs of clouds. Scalar products can be made secure with or without cryptography. Cryptographic approaches use a variety of techniques; for instance, the protocol in [23] involves homomorphic encryption. Non-cryptographic approaches are rather based on modifying the data before sharing them, in such a way that the original data cannot be inferred from the shared data but the final results are preserved (e.g., [13], [25]).

We next recall two protocols that will be used in the remainder of this paper to compute secure scalar products with honest-but-curious clouds. The security of the

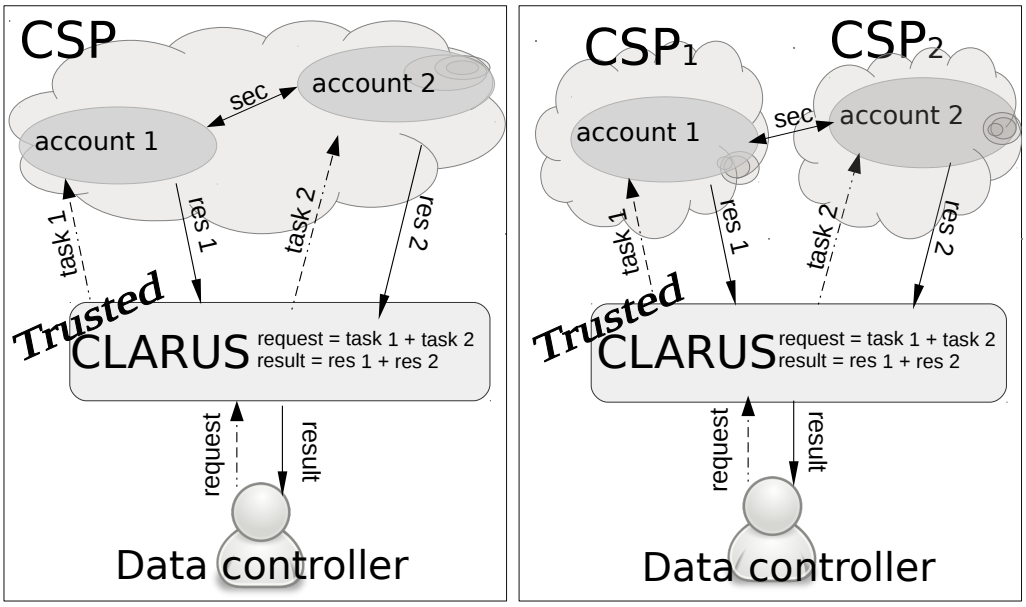


Fig. 1 System architecture: CLARUS sits in the controller's trusted premises while the CSPs are untrusted. Left, data and computation splitting among two accounts at the same CSP; right, splitting among two different CSPs. When the data controller sends a computation "request" to CLARUS, CLARUS reformulates this computation in two tasks "task 1" and "task 2", which are sent to the involved CSPs. The CSPs use (if needed) a secure protocol "sec" to exchange data (e.g. a secure scalar product), and then they send to CLARUS their partial results "res 1" and "res 2". CLARUS combines the partial results and sends the final result to the data controller.

first protocol does not rely on cryptography, whereas the security of the second does. Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two vectors with n components owned by Alice and Bob (e.g., two CSPs), respectively. The goal is to securely compute the product $\boldsymbol{x}^T\boldsymbol{y}$. Note that the desired result is obtained by the local proxy (CLARUS) and, therefore, any disclosure by Alice or Bob to CLARUS during the protocol does not entail any privacy leakage, because (unlike Alice and Bob) CLARUS is trusted by the data controller.

5.1 Non-cryptographic secure scalar product

In [17], the authors propose a protocol based on what they call a commodity server. Let Alice and Bob be as previously defined and Charlie be a third cloud who plays the role of the commodity server. This protocol was identified as the most efficient one in the comparison of [49]. Its privacy relies on the fact that the original vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are not shared at any time by the respective CSPs owning them; only linear transformations of them are, such that the number of unknowns (randomness) added by the transformations is greater than or equal to the number of private unknowns.

In [14], we modified the above protocol to make it suitable for the CLARUS scenario and we increased its security by adding permutation operations by Alice and Bob (see security discussion below). The resulting protocol was:

Protocol 1

1. *Charlie sends to Alice the seed for a common random generator of a random n-vector $\boldsymbol{r}_x$, and sends to Bob the seed for a common random generator of a random n-vector $\boldsymbol{r}_y$ (or equivalently generates and sends the vectors if doing so is faster than Alice and Bob generating them).*
2. *Alice computes $\hat{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{r}_x$ and randomly permutes the values in $\hat{\boldsymbol{x}}$ to obtain $\hat{\boldsymbol{x}}' = \mathcal{P}_x(\hat{\boldsymbol{x}})$.*
3. *Alice sends $\hat{\boldsymbol{x}}'$ to Bob and $\boldsymbol{r}'_x = \hat{\boldsymbol{x}}' - \boldsymbol{x}$ to Charlie.*
4. *Bob computes $\hat{\boldsymbol{y}} = \boldsymbol{y} + \boldsymbol{r}_y$ and randomly permutes the values in $\hat{\boldsymbol{y}}$ to obtain $\hat{\boldsymbol{y}}' = \mathcal{P}_y(\hat{\boldsymbol{y}})$.*
5. *Bob sends $\hat{\boldsymbol{y}}'$ to Alice and $\boldsymbol{r}'_y = \hat{\boldsymbol{y}}' - \boldsymbol{y}$ to Charlie.*

6. *Charlie sends* $p = (\boldsymbol{r}'_x)^T \boldsymbol{r}'_y$ *(note that* p *is a number) to CLARUS.*
7. *Bob sends* $t = (\hat{\boldsymbol{x}}')^T \boldsymbol{y}$ *to CLARUS.*
8. *Alice sends* $s_x = (\boldsymbol{r}'_x)^T \hat{\boldsymbol{y}}'$ *to CLARUS.*
9. *CLARUS computes*

$$t - s_x + p \ \ [= (\boldsymbol{x}+\boldsymbol{r}'_x)^T \boldsymbol{y} - (\boldsymbol{r}'_x)^T (\boldsymbol{y}+\boldsymbol{r}'_y) + (\boldsymbol{r}'_x)^T (\boldsymbol{r}'_y)] = \boldsymbol{x}^T \boldsymbol{y}.$$

The following proposition characterizes the security of Protocol 1 and is proven in Appendix B:

Proposition 1 *Protocol 1 does not allow Charlie to learn* $\boldsymbol{x}$ *or* $\boldsymbol{y}$*, it does not allow Alice to learn* $\boldsymbol{y}$*, and it does not allow Bob to learn* $\boldsymbol{x}$*.*

Note that in Protocol 1, vectors $\boldsymbol{r}'_x$ and $\boldsymbol{r}'_y$ should be reused in successive instances of the protocol with the same original data vectors $\boldsymbol{x}$ and/or $\boldsymbol{y}$, in order to avoid leaking new equations that would facilitate the reconstruction of the original data vectors by an unauthorized part. Vectors $\boldsymbol{r}'_x$ and $\boldsymbol{r}'_y$ are kept in memory respectively by Alice and Bob, instead of being regenerated every time the protocol is run. The underlying assumption is that the CSPs have unlimited storage and, therefore, they can store any random matrices or vectors that may need to be reused.

5.2 Cryptographic secure scalar product

In [23], the authors proposed a cryptographic protocol based on Paillier's homomorphic cryptosystem [32]. The use of cryptography increases the computational complexity with respect to non-cryptographic protocols. However, it is attractive in terms of security since the difficulty for Alice to learn $\boldsymbol{y}$ or Bob to learn $\boldsymbol{x}$ amounts to breaking a cryptosystem proven to be secure. Let $\boldsymbol{x} = (x_1, \ldots, x_n)^T$ and $\boldsymbol{y} = (y_1, \ldots, y_n)^T$ be the private vectors of Alice and Bob, respectively.

In [36], we suggested a variant that consists of the following steps:

Protocol 2

Set-up phase:

1. *Alice generates a private and public key pair* (s_k, p_k) *and sends* p_k *to Bob.*

Scalar product:

2. *Alice generates the ciphertexts* $c_i = Enc_{p_k}(x_i; r_i)$*, where* r_i *is a random number in* $\mathbb{F}_N$*, for every* $i = 1, \ldots, n$*, and sends them to Bob.*
3. *Bob computes* $\omega = \prod_{i=1}^{n} c_i^{y_i}$*.*
4. *Bob generates a random plaintext* s_B*, a random number* r' *and sends* $\omega' = \omega Enc_{p_k}(-s_B; r')$ *to Alice and sends* s_B *to CLARUS.*
5. *Alice sends* $s_A = Dec_{s_k}(\omega') = \boldsymbol{x}^T \boldsymbol{y} - s_B$ *to CLARUS.*
6. *CLARUS computes* $s_A + s_B = \boldsymbol{x}^T \boldsymbol{y}$*.*

For completeness, we justify in Appendix B that Protocol 2 still offers the same security versus Alice and Bob as the basic protocol [23].

Protocol 2 works in a finite field $\mathbb{F}_N$, where the order N is the product of two primes p and q of the same length and such that $\gcd(pq, (p-1)(q-1)) = 1$. In case Alice and Bob need to execute this protocol several times, they can reuse the public and private keys; therefore, the set-up step (first step) needs to be executed only once. The complexity of all these operations depends on N: the larger N, the more computationally demanding they are. Since we are computing $\mathbf{x}^T\mathbf{y} \mod N$, if we do not want the result to be modified by the modulus, it must hold that $N > \mathbf{x}^T\mathbf{y}$. Let $M_{\mathbf{x}} = \max_{x_i \in \mathbf{x}} x_i$ and $M_{\mathbf{y}} = \max_{y_i \in \mathbf{y}} y_i$. It is sufficient to choose $N > nM_{\mathbf{x}}M_{\mathbf{y}}$.

6 Privacy-preserving frequency-based analyses on the cloud

In this section, we show how the frequency-based multivariate analyses introduced in Section 3.1 can be performed on split data outsourced to separate clouds by relying on secure scalar products. For each analysis, we give a protocol describing how the CLARUS proxy decomposes and orchestrates calculations and aggregates partial results. To avoid overloading the local system in which the CLARUS proxy runs, the protocols are designed to keep the workload of the CLARUS proxy as low as possible by outsourcing as much storage and computation as possible to the CSPs in a privacy-preserving way.

To calculate the χ^2-test, ANOVA or Cramér's V, CLARUS orchestrates the calculation of the contingency table of the split attributes stored in separated CSPs. This table is the input of the aforementioned tests.

To obtain the contingency table from data vertically split among several clouds, one just needs to compute the table cells. Let $(a_1, \ldots, a_n)^T$ and $(b_1, \ldots, b_n)^T$ be the vectors of values from the attributes $\mathbf{a}$ and $\mathbf{b}$, owned by the CSPs Alice and Bob, respectively. A cell C_{ij} (for every $i = 1, \ldots, h$ and $j = 1, \ldots, k$) is computed by counting the number of records in the original data set containing both the categories $c_i(\mathbf{a})$ and $c_j(\mathbf{b})$. Alice creates a new vector $\mathbf{x} = (x_1, \ldots, x_n)^T$ such that

$$x_l = \begin{cases} 1 & \text{if } a_l = c_i(\mathbf{a}) \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } l = 1, \ldots, n. \tag{6}$$

Bob creates $\mathbf{y} = (y_1, \ldots, y_n)^T$ such that

$$y_l = \begin{cases} 1 & \text{if } b_l = c_j(\mathbf{b}) \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } l = 1, \ldots, n. \tag{7}$$

The scalar product $\mathbf{x}^T\mathbf{y}$ (computed by means of Protocol 1 or Protocol 2 above) gives the number C_{ij} of records in the original data set containing both the categories $c_i(\mathbf{a})$ and $c_j(\mathbf{b})$.

Specifically, Alice and Bob can use Protocol 1 or Protocol 2 above to securely compute C_{ij} by just adding two preliminary steps to the scalar product computation part: one step by Alice to generate $\mathbf{x}$ from $(a_1, \ldots, a_n)^T$ using Equation (6), and another step by Bob to generate $\mathbf{y}$ from $(b_1, \ldots, b_n)^T$ using Equation (7).

6.1 Security

The only modification with respect to Protocols 1 or 2 is that Alice and Bob compute $\mathbf{x}$ and $\mathbf{y}$, respectively. These computations are done by the clouds in isolation, i.e., without exchanging information; hence, the security of the underlying secure scalar product protocol is preserved.

6.2 Cost

Once the contingency table is obtained, frequency-based tests, which have a low computational cost, can be run locally by CLARUS. In fact, for the χ^2-test, ANOVA and Cramér's V the most demanding computation is a linear regression; therefore, given a $h \times k$ contingency table with $h < k$, the linear regression has complexity $O(h^2k + h^3)$ (notice that h and k are much smaller than the number n of records of the original data set). For CSPs Alice and Bob the computation of one cell has $O(n)$ cost in both Protocol 1 and Protocol 2. In particular, in Protocol 1 Alice and Bob have to perform, respectively, n products and n reads as the most demanding computations; Charlie (the third cloud needed in the protocol) generates two random n-vectors and CLARUS just performs two sums. In Protocol 2 Alice performs n encryptions, n reads and n random number generations. Bob performs n reads and n products as demanding computations. CLARUS computes one sum. Since the contingency table has $h \times k$ cells, Alice's and Bob's calculation has $O(n \times h \times k)$ cost. CLARUS just needs to compute 2 sums in Protocol 1 or 1 sum in Protocol 2 for each table cell, that is, constant cost. Therefore, the CLARUS computation has complexity $O(h \times k)$.

Another reason to conduct the calculation of the frequency-based tests locally is that sharing the contingency table with a CSP can lead to privacy issues, because the table may contain cells with values one or zero that may allow re-identifying some subjects. For instance, if a cell representing the number of Asian people with HIV that answered a specific survey has value equal to one, just knowing that only one participant of the survey was Asian discloses that he is sick. Moreover, this information can be enough to recognize a subject if, for example, the survey was carried out in an area with only few Asian families.

Observe that, if one CSP stores in its own data fragment all the attributes required for the contingency table computation, all the calculations are done by that CSP in isolation, and CLARUS just receives the result of the required frequency-based test.

7 Privacy-preserving semantic-based analyses on the cloud

In this section, we deal with the semantic-based multivariate analyses mentioned in Section 3.2. We show how they can be performed on split data outsourced to separate clouds.

As introduced in Section 3, the calculation of the distance covariance requires measuring the pairwise semantic distance between the nominal values of each attribute. The pairwise distances as well as the double-centered matrices are computed among the values of one attribute at once and, therefore, the CSP owning the attribute performs the calculation in isolation. Each CSP can also compute the distance variance of its attribute in isolation. Then the CSPs use a secure scalar product like those in Sections 5.1 or 5.2 to securely compute the distance covariances in view of completing the distance covariance matrix $\hat{\mathbf{\Sigma}}$.

Formally, let $\mathbf{x}^1 = (x_1^1, \ldots, x_n^1)^T$ and $\mathbf{x}^2 = (x_1^2, \ldots, x_n^2)^T$ be vectors of values of two nominal attributes owned by CSPs Alice and Bob, respectively. Alice computes $\mathbf{X}^1$ and $\hat{\mathbf{X}}^1$ and Bob computes $\mathbf{X}^2$ and $\hat{\mathbf{X}}^2$. In this case, the distance covariance matrix $\hat{\mathbf{\Sigma}}$ of $\mathbf{X} = (\mathbf{x}^1, \mathbf{x}^2)$ is given by

$$\hat{\mathbf{\Sigma}} = \left(\begin{array}{c|c} d\mathcal{V}_n(\mathbf{x}^1) & d\mathcal{V}_n(\mathbf{x}^1, \mathbf{x}^2) \\ \hline d\mathcal{V}_n(\mathbf{x}^2, \mathbf{x}^1) & d\mathcal{V}_n(\mathbf{x}^2) \end{array} \right).$$

Note that $d\mathcal{V}_n(\mathbf{x}^i, \mathbf{x}^j)$ is the square root of $d\mathcal{V}_n^2(\mathbf{x}^i, \mathbf{x}^j)$, for $i, j = 1, \ldots, m$, and that X^j, X_{kl}^j, $d\mathcal{V}_n(\mathbf{x}^j)$, for $j = 1, \ldots, m$, are separately computed by the CSP storing the respective attribute. The most challenging task is, therefore, calculating the squared sample distance covariance, i.e., Equation (4), which requires performing n secure scalar products of vector pairs, where the

two vectors in each pair are respectively held by two different CSPs.

In fact, by calling $\mathbf{X}_k^1 = (X_{k1}^1, \ldots, X_{kn}^1)^T$ and $\mathbf{X}_k^2 = (X_{k1}^2, \ldots, X_{kn}^2)^T$ for $k = 1, \ldots, n$, we can rewrite Equation (4) as

$$d\mathcal{V}_n^2(\mathbf{x}^1, \mathbf{x}^2) = \frac{1}{n^2} \sum_{k=1}^{n} \left(\sum_{l=1}^{n} X_{kl}^1 X_{kl}^2 \right) = \frac{1}{n^2} \sum_{k=1}^{n} (\mathbf{X}_k^1)^T \mathbf{X}_k^2, \tag{8}$$

where the n scalar products are $(\mathbf{X}_k^1)^T \mathbf{X}_k^2$ for $k = 1, \ldots, n$.

Therefore, once the double-centered matrices are obtained, the distance covariance matrix computation with data split among different CSPs can be decomposed into several secure scalar products to be conducted between pairs of clouds. To compute $(\mathbf{X}_k^1)^T \mathbf{X}_k^2$, for $k = 1, \ldots, n$, use one of the two protocols in Sections 5.1 and 5.2 by adding two preliminary steps: one step for Alice to compute $\mathbf{x} = \mathbf{X}_k^1$ from $\mathbf{x}^1$, and another step for Bob to compute $\mathbf{y} = \mathbf{X}_k^2$ from $\mathbf{x}^2$.

7.1 Security

The two preliminary steps added before the secure scalar product are separately performed by Alice and Bob, so there is no additional exchange of information between the clouds. Hence, the security of the underlying secure scalar product protocol is preserved.

7.2 Cost

Calculating the distance covariance matrix between two nominal attributes has a quadratic cost, both in time and storage. Moreover, generating the semantic-distance matrices (Eq. (1)) requires using the semantic measures. Let h_1 be the number of categories of $\mathbf{x}^1$ and h_2 be the number of categories of $\mathbf{x}^2$, where $h_1,\ h_2 \leq n$; then $h_1^2/2$ and $h_2^2/2$ semantic distances are computed for each attribute by Alice and Bob, respectively. Recalling the costs mentioned in Section 3.2 and justified in Appendix A for each type of semantic measure, we have that the cost of generating the semantic-distance matrix is $O(h_j^2 \times D)$ for the edge-counting measure, $O(h_j^2 \times S)$ for the feature-based measure and $O(h_j^2 \times (C+D))$ for the information content-based measure, for $j = 1, 2$ and where D is the depth of the taxonomy, S is the maximum number of subsumers of any concept and C is the total number of concepts in the ontology (which can be in the order of thousands or hundreds of thousands in large ontologies). On the other hand, the double-centered matrix (Eq. (2)), which is also computed by each CSP independently, has $O(n^2)$ computational cost.

Finally, the distance covariance matrix computation is decomposed into several scalar products, where the total number of scalar products performed by the CSPs is $3n$. Each scalar product has $O(n)$ computational cost for both protocols in Sections 5.1 and 5.2. In particular, in Protocol 1 Alice and Bob have to perform, respectively, n products as the most demanding computations, Charlie generates two random n-vectors, and CLARUS just performs two sums. In Protocol 2, Alice performs n encryptions and n random number generations, Bob performs n products as the most demanding computations, and CLARUS computes one sum. Consequently, the CSPs' computation has $O(n^2)$ cost and CLARUS's computation has $O(1)$ cost. The storage needs at the CSPs are also quadratic due to the need to create several $n \times n$ matrices, i.e., 1 semantic-distance matrix and 1 double-centered matrix per attribute.

One can notice that the calculation of the semantic-distance covariance is significantly costlier than the frequency-based method (both in time and storage); yet, the protocol we propose is able to outsource the cost to the CSPs, thus keeping the CLARUS workload low even with large data sets.

8 Experimental results

Most of the literature on outsourcing matrix computations to CSPs focuses on numerical data, see for example [5, 26, 31]. Computing on nominal data is harder [21] and contributions like [19] show how basic search can be performed on nominal data in CSPs. However, the combined problem of outsourcing statistical multivariate analyses on categorical data to the cloud has not been tackled yet by previous works. Therefore, our tests cannot compare with previous similar approaches. For this reason, we show how to make the analyses on categorical data over several CSPs by decomposing them into several scalar products, and we focus on comparing the performance with the two best secure scalar product protocols in [14]. We also try different semantic distances for the computation of the distance covariance matrix.

In the rest of this section, we report the results of the implementation of our protocols in a real setting. As a use case, we employed the most computationally demanding analysis: the semantic-distance covariance. As evaluation metrics, we report the workload of each entity (CLARUS and the CSPs) and quantify the percentage of workload that our protocols were able to securely outsource to the CSPs w.r.t. a local implementation of the analysis.

The tests were run in a free-tier CSP provided by Amazon Web Services (AWS). It is important to note that the computing power and storage of such a free-of-charge service are substantially limited and, therefore, significant improvements can be expected when moving to payment services. On the client side, a local computer was configured to act as the CLARUS proxy, which is in charge of orchestrating the storage and the calculations on the outsourced data. The specifications of AWS and CLARUS are summarized in Table 2.

The experiments were conducted on a sample of 1,000 records with two nominal attributes extracted from a patient discharge database provided by the California Office of Statewide Health Planning and Development [8]. The two nominal attributes represent the diagnosis ($\mathbf{x}$) and medical procedure ($\mathbf{y}$) of each patient. Notice that the size of the sample is deliberately small because of the limited resources of the CSPs instances we used. To cope with larger data sets, one just needs to hire more powerful CSP instances, e.g., see those offered by AWS [3] in Table 3. Given the computational cost figures we discussed in the previous section, scaling the obtained results for larger data sets and more attributes is straightforward.

SNOMED-CT was used as the ontology for the semantic distance calculation in the semantic-based test. SNOMED-CT models 321,901 clinical concepts and constitutes the largest and most detailed medical knowledge base [43].

In all the experiments, two AWS CSPs (Alice and Bob) separately store the two attributes (diagnosis ($\mathbf{x}$) and procedures ($\mathbf{y}$), respectively), whereas a third CSP (Charlie) is used as commodity server. For each analysis, we report the storage requirements and workload of each CSP and CLARUS for the protocols we propose, and compare them against a local implementation in which CLARUS should store the whole data and perform all the computations.

As detailed in Section 7, first each CSP computes in isolation the semantic-distance matrix (Eq. (1)) and the double-centered matrix (Eq. (2)) of the attribute it stores; then, Alice and Bob jointly work on the calculation of the distance covariance. CLARUS performs a small part in this latter calculation, and its workload depends on the secure scalar protocol in use: Protocol 1 or Protocol 2. In particular, the total number of scalar products performed by the CSPs is $(m(m-1)/2) * n + m * n$, out of which $(m(m-1)/2) * n$ are secure scalar products, being n the number of records and m the number of attributes.

In the local implementation, CLARUS plays the part of the data controller and is required to perform all the computation by itself: semantic-distance matrices, double-centered matrices and the distance covariance. However, since CLARUS owns the whole data and runs in a trusted environment, no secure scalar products are needed.

Table 4 shows the storage requirements of the calculations for the cloud-based and local scenarios. The storage is broken down into long-term and temporary: the former corresponds to the storage of the split data, whereas the latter is the storage required to conduct the calculation at some point, which can be discarded once the calculation is finished. In terms of temporary storage, the CSPs (or CLARUS in the local solution) need to load into RAM the SNOMED-CT ontology, which requires 242.5 MB. The semantic-distance and the double-centered matrices of the attributes are stored in the long-term storage for them to be re-used in further calculations. The local solution, which requires storing the matrices of all the attributes, can be considerably heavy for CLARUS when the number of records and/or attributes is large. In contrast, in the cloud-based solution only the semantic-distance covariances are stored by CLARUS. The use of secure scalar products imperceptibly increases the required storage (for 1,000 records, the storage increases by around 0.032 MB for Protocol 1 and by 0.065 MB for Protocol 2).

Table 5 shows the computation and communication runtimes of the distance covariance calculation with the three ontology-based semantic measures. Notice that in the local scenario there is no exchange of information between separate entities and, therefore, there is no communication cost. In the cloud-based solution, Protocols 1 and 2 were used for the computation of the secure scalar products. Observe that Protocol 2 results in higher costs in terms of computation due to the use of cryptographic primitives. Moreover, the runtime of Alice is significantly greater than Bob's, although the attributes have the same length. The reason is that the SNOMED-CT taxonomy for the attribute stored by Alice (diagnosis, i.e., clinical finding) is much larger than that of Bob's attribute (procedure), as shown in Table 6. Furthermore, within the 1,000-record data set, Alice's attribute has 434 categories, whereas Bob's attribute has only 342; hence, Alice needs to perform a greater number of semantic-distance assessments.

The reported runtime figures are consistent with the cost of the semantic measures we detail in Appendix A: whereas the edge-counting and the feature-based measures have similar costs, because both analyze the set of ancestors of the concepts to be compared, the measure based on information content is significantly costlier (around 8 times slower in the local solution) due to the need to iterate through all the hyponyms of each concept. In fact, the calculation of

Table 2 Specifications of CLARUS (the trusted proxy running on a local computer) and the AWS CSPs (free-of-charge t2.micro Amazon EC2 instances)

Machine	Operating System	Width(bits)	CPU(GHz)	RAM(GB)	HDD(GB)	Instances
CLARUS	Windows 7	64	2.5	8	500	1
AWS CSP (t2.micro instance)	Ubuntu Server 16.04 LTS	64	2.4	1	30	3

Table 3 AWS instance types. The t2.micro free-of-charge instance was used in our experiments.

AWS instance type	Name	CPU Cores	RAM(GB)	Clock Speed(GHz)
General purpose	t2.micro	1	1	Up to 3.3
General purpose	t2.2xlarge	8	32	Up to 3.0
General purpose	m4.16xlarge	64	256	2.3
Compute optimized	c4.8xlarge	36	60	2.9
Accelerated computing	f1.16xlarge	64	976	2.3
Memory optimized	r4.16xlarge	64	488	2.3
Memory optimized	x1.32xlarge	128	1,952	2.3

Table 4 Long-term and temporary storage for the semantic-distance covariance calculation with two attributes and 1,000 records

Storage requirements (MB)									
LOCAL			CLOUD						
Long-term	Temporary		Long-term				Temporary			
CLARUS	CLARUS		Alice	Bob	CLARUS	Charlie	Alice	Bob	CLARUS	Charlie
40	242.5		16	16	8	0	242.5	242.5	0	0

the semantic-distance matrix takes around 13 hours for Alice's attribute (whose domain is significantly larger than Bob's). Even if the runtime of the cloud-based scenario is around 5 times greater that the local one, we should consider the very limited resources of the free t2.micro instance we use. With more powerful instances, runtimes will be decreased to reasonable figures, e.g., a general-purpose t2.2xlarge instance should be around 8 times faster than the free instance (see Table 3), which would make the cloud-based calculation faster than the local one.

Since absolute runtime figures depend on the amount of resources of the CSPs, in Table 7 we report a more general metric stating the percentage of runtime saved by CLARUS (which runs on local premises) when outsourcing local calculations to the cloud. The runtime saved by CLARUS was computed with the formula

$$100 * \frac{\text{CLARUS}_l - \text{CLARUS}_c}{\text{CLARUS}_l}, \tag{9}$$

where CLARUS_l represents the computation runtime of CLARUS in the local scenario and CLARUS_c represents the computation runtime of CLARUS in the cloud-based scenario.

Table 6 Number of concepts in some taxonomies of SNOMED-CT

Taxonomy	Number of concepts
Body structure	31,206
Clinical findings	104,737
Pharmaceutical/biologic product	17,425
Procedure	55,880
Substance	25,911

Since the calculation of the semantic-distance matrices is, by far, the costliest operation (especially for the measure based on information content), outsourcing this calculation results in very large savings (i.e., very low workload and also very low storage requirements) for CLARUS.

9 Conclusions and future work

Data splitting is an alternative to encryption that is more flexible and efficient for securing sensitive data outsourced to the cloud. With data splitting, CSPs do not only store data, but they can efficiently conduct computations on the data they store in a privacy-preserving manner. Multivariate analyses are, however, challenging; the reason is not just their potentially large computational cost, but also the difficulty of performing the calculations involving data fragments stored in different clouds.

In this paper, we have presented protocols to securely outsource the computation of several multivariate statistical analyses on nominal data split among a number of honest-but-curious clouds. Our protocols are designed to outsource as much workload as possible to the CSPs, which is especially interesting for computationally demanding calculations that may not be affordable locally. In this way, we retain the cost-saving benefits of the cloud while ensuring that the outsourced data do not incur privacy risks.

Empirical tests conducted on AWS free tier cloud instances confirm our theoretical assumptions. Experimental results clearly show that outsourcing the calcu-

Table 5 Computation and communication runtimes for the distance-covariance calculation with the three semantic-distance measures and the two secure scalar product protocols. The computation runtime (Comp. runtime) represents the time each entity spent following the protocol. The communication runtime (Comm. runtime) is an approximation of the time the CSPs and CLARUS spent sending and receiving the data. The times are given in minutes (m.).

LOCAL	CLOUD						
Comp. runtime (m.)	Comp. runtime (m.)						Comm. runtime (m.)
Edge-counting measure (Eq. (10))							
CLARUS		Alice	Bob	Charlie	CLARUS	Total comp.	Total comm.
21.3	Prot.1	9.9	2.4	3.5×10^{-4}	2.8×10^{-5}	12.3	16.5
	Prot.2	34.2	2.4	–	8.7×10^{-5}	36.7	6.3
Feature-based measure (Eq. (11))							
CLARUS		Alice	Bob	Charlie	CLARUS	Total comp.	Total comm.
22.8	Prot.1	9.6	2.5	3.5×10^{-4}	2.7×10^{-5}	12.1	16.5
	Prot.2	33.8	2.4	–	9.1×10^{-5}	36.2	6.6
Information content-based measure (Eq. (12))							
CLARUS		Alice	Bob	Charlie	CLARUS	Total comp.	Total comm.
183.8	Prot.1	836.5	49.2	3.9×10^{-4}	2.6×10^{-5}	885.7	16.5
	Prot.2	863.2	49.3	–	9.7×10^{-5}	912.5	6.6

Table 7 Percentage of computation runtime saved by CLARUS when moving from the local scenario to the cloud-based scenario for the different semantic measures and scalar product protocols

Computation runtime saved by CLARUS (%)							
Edge-counting			Feature-based			Information content-based	
Prot. 1	99.99987		Prot. 1	99.99988		Prot. 1	99.99999
Prot. 2	99.99959		Prot. 2	99.99960		Prot. 2	99.99995

lations to the cloud considerably decreases the workload of the data controller, who can save more than 99.999 percent of the runtime for the most demanding test we considered.

As future research, we plan to combine the numerical methods presented in [9] with those meant for nominal data presented here to deal with data sets with heterogeneous attribute types. Other scenarios more challenging than the honest-but-curious CSPs assumption may also be considered, e.g., malicious or colluding clouds. Furthermore, we also intend to tackle outsourcing additional multivariate analyses for nominal data, such as multidimensional scaling, multiple correspondence analysis and non-linear principal component analysis. Finally, using non-free cloud services will allow experimenting on larger data sets, which can be expected to increase even more the proportional computational savings at the controller.

Acknowledgments and disclaimer

Partial support to this work has been received from the European Commission (projects H2020-700540 "CANVAS" and H2020-644024 "CLARUS"), from the Government of Catalonia (ICREA Acadèmia Prize to J. Domingo-Ferrer and grant 2017 SGR 705), and from the Spanish Government (projects RTI2018-095094-B-C21 "CONSENT" and TIN2016-80250-R "Sec-MCloud"). The authors are with the UNESCO Chair in Data Privacy, but the views in this paper are the authors' own and are not necessarily shared by UNESCO.

References

1. Aggarwal, G., Bawa, M., Ganesan, P., Garcia-Molina, H., Kenthapadi, K., Motwani, R., Srivastava, U., Thomas, D., Xu, Y.: Two can keep a secret: A distributed architecture for secure database services. In CIDR 2005, pp. 186–199 (2005).
2. Agresti, A., Kateri, M.: Categorical Data Analysis. Springer (2011).
3. Amazon EC2 Instance Types. `https://aws.amazon.com/ec2/instance-types/?nc1=h_ls`
4. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: A view of cloud computing. Communications of the ACM, vol. 53, no. 4, pp. 50–58 (2010).
5. Atallah, M. J., Frikken, K. B.: Securely outsourcing linear algebra computations. In 5th ACM Symposium on Information, Computer and Communications Security – ASIACCS 2010, pp. 48-59. ACM (2010).
6. Batet, M., Harispe, S., Ranwez, S., Sánchez, D., Ranwez, V.: An information theoretic approach to improve semantic similarity assessments across multiple ontologies. Information Sciences, vol. 283, pp. 197–2010 (2014).
7. Batet, M., Sánchez, D.: A review on semantic similarity. In Encyclopedia of Information Science and Technology, Third Edition, pp. 7575–7583. IGI Global (2015).
8. California patient discharge data: California Office of Statewide Health Planning and Development (OSHPD), 2009. `http://www.oshpd.ca.gov/HID/DataFlow/index.html`

9. Calviño, A., Ricci, S., Domingo-Ferrer, J.: Privacy-preserving distributed statistical computation to a semi-honest multi-cloud. In IEEE Conference on Communications and Network Security (CNS 2015), pp. 506–514. IEEE (2015).
10. Cimiano, P.: Ontology Learning and Population from Text: Algorithms, Evaluation and Applications. Springer Science & Business Media (2006).
11. Ciriani, V., De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S., Samarati, P.: Selective data outsourcing for enforcing privacy. Journal of Computer Security, vol. 19, no. 3, pp. 531–566 (2011).
12. CLARUS - A Framework for User Centred Privacy and Security in the Cloud, H2020 project (2015-2017). `http://www.clarussecure.eu`
13. Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X., Zhu, M.: Tools for privacy preserving distributed data mining. ACM SiGKDD Explorations Newsletter, vol. 4, no. 2, pp. 28–34 (2002).
14. Domingo-Ferrer, J., Ricci, S., Domingo-Enrich, C.: Outsourcing scalar products and matrix products on privacy-protected unencrypted data stored in untrusted clouds. Information Sciences, vol. 436-437, pp. 320–342 (2018).
15. Domingo-Ferrer, J., Sánchez, D., Rufian-Torrell, G.: Anonymization of nominal data based on semantic marginality. Information Sciences, vol. 242, pp. 35–48 (2013).
16. Domingo-Ferrer, J., Torra, V.: Ordinal, continuous and heterogeneous k-anonymity through microaggregation. Data Mining and Knowledge Discovery, vol. 11, no. 2, pp. 195–212 (2005).
17. Du, W., Han, Y., Chen, S.: Privacy-preserving multivariate statistical analysis: linear regression and classification. In SDM, vol. 4. SIAM, pp. 222–233 (2004).
18. Dubovitskaya, A., Urovi, V., Vasirani, M., Aberer, K., Schumacher, M.: A Cloud-Based eHealth Architecture for Privacy Preserving Data Integration. In ICT Systems Security and Privacy Protection, pp. 585–598. Springer (2015).
19. Fu, Z., Sun, X., Ji, S., Xie, G.: Towards efficient content-aware search over encrypted outsourced data in cloud. In Computer communications, IEEE INFOCOM 2016-the 35th annual IEEE international conference: pp. 1-9. IEEE (2016).
20. General Data Protection Regulation. European Union. `http://www.gdpr-info.eu`
21. Ghattas, B., Michel, P., Boyer, L.: Clustering nominal data using unsupervised binary decision trees: Comparisons with the state of the art methods. Pattern Recognition; 67:177-85 (2017).
22. Gelman A.: Analysis of variance—why it is more important than ever. The Annals of Statistics, vol. 33, no. 1, pp. 1–53 (2005).
23. Goethals, B., Laur, S. , Lipmaa, H., Mielikäinen, T.: On private scalar product computation for privacy-preserving data mining. In Information Security and Cryptology - ICISC 2004, LNCS, vol. 3506, pp. 104–120. Springer (2005).
24. Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Schulte Nordholt, E., Spicer, K., De Wolf, P.-P.: Statistical Disclosure Control. Wiley (2006).
25. Karr, A., Lin, X., Sanil, A., Reiter, J.: Privacy-preserving analysis of vertically partitioned data using secure matrix products Journal of Official Statistics, vol. 25, no. 1, p. 125-138 (2009).
26. Lei, X., Liao, X., Huang, T., Li, H., Hu, C.: Outsourcing large matrix inversion computation to a public cloud. IEEE Transactions on Cloud Computing 1(1):78-87 (2013).
27. Lei, X., Liao, X., Huang, T., Heriniaina, F.: Achieving security, robust cheating resistance, and high-efficiency for outsourcing large matrix multiplication computation to a malicious cloud. Information Sciences 280:205-217 (2014).
28. Li, H., Yang, Y., Luan, T.H., Liang, X., Zhou, L., Shen, X.S.: Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data. IEEE Transactions on Dependable and Secure Computing, 13(3):312-25 (2016).
29. Li, L., Lu, R., Choo, K.K., Datta, A., Shao, J.: Privacy-preserving-outsourced association rule mining on vertically partitioned databases. IEEE Transactions on Information Forensics and Security; 11(8):1847-61 (2016).
30. Lin, D.: An information-theoretic definition of similarity. In Proceedings of the 15th International Conference on Machine Learning, ICML 1998, pp. 296–304 (1998).
31. Nassar, M., Erradi, A., Sabry, F., Malluhi, Q. M.: Secure outsourcing of matrix operations as a service. In IEEE CLOUD 2013, pp. 918-925. IEEE (2014)
32. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In Advances in Cryptology — EUROCRYPT '99, LNCS, vol. 1592, pp. 223–238. Springer (1999).
33. Rada, R., Mili, H., Bichnell, E., Blettner, M.: Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, pp. 17—30 (1989).
34. Ren, K., Wang, C., Wang, Q.: Security challenges for the public cloud. IEEE Internet Computing, vol. 16, no. 1, pp. 69–73 (2012).
35. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI, vol. 1, pp. 448–453 (1995).
36. Ricci S., Domingo-Ferrer J., Sánchez D.: Privacy-preserving cloud-based statistical analyses on sensitive categorical data. In Modeling Decisions for Artificial Intelligence, pp. 227–238. Springer (2016).
37. Rodríguez-García, M., Batet, M., Sánchez, D.: A semantic framework for noise addition with nominal data. Knowledge-based Systems, vol. 112, pp. 103-118 (2017).
38. Samarati, P.: Protecting respondents' identities in microdata release. IEEE Transactions on Knowledge and Data Engineering, vol. 13, no. 6, pp. 1010-1027 (2001)
39. Sánchez, D., Batet, M.: Privacy-preserving data outsourcing in the cloud via semantic data splitting. Computer Communications, vol. 110, pp. 187–201 (2017).
40. Sánchez, D., Batet, M., Isern, D., Valls, A.: Ontology-based semantic similarity: a new feature-based approach. Expert Systems with Applications, vol. 39, no. 9, pp. 7718–7728 (2012).
41. Sánchez, D., Batet, M., Isern, D.: Ontology-based information content computation. Knowledge-based Systems, vol. 24, no. 2, pp. 297-303 (2011).
42. Sánchez, D., Batet, M., Martínez, S., Domingo-Ferrer, J.: Semantic variance: an intuitive measure for ontology accuracy evaluation. Engineering Applications of Artificial Intelligence, vol. 39, pp. 89–99 (2015).
43. SNOMED-CT Ontology. `https://en.wikipedia.org/wiki/SNOMED_CT`
44. Sun, Y., Yu, Y., Li, X., Zhang, K., Qian, H., Zhou, Y.: Batch verifiable computation with public verifiability for outsourcing polynomials and matrix computations. In Australasian Conference on Information Security and Privacy – ACISP 2016, Lecture Notes in Computer Science, vol. 9722, pp. 293–309. Springer (2016).

45. Székely, G.J., Rizzo, M.L.: Brownian distance covariance. The Annals of Applied Statistics, vol. 3, no. 4, pp. 1236–1265 (2009).
46. Taha, A., Hadi, A.S.: Pair-wise association measures for categorical and mixed data. Information Sciences; 346:73-89 (2016).
47. Tugrul, B., Polat, H.: Privacy-preserving kriging interpolation on partitioned data Knowledge-based Systems, vol. 62, pp. 38–46 (2014).
48. U.S. Federal Trade Commission: Data Brokers, A Call for Transparency and Accountability (2014).
49. Wang, I.-C., Shen, C.-H., Hsu, T.-S., Liao, C.-C., Wang, D. W., Zhan, J.: Towards empirical aspects of secure scalar product. IEEE Transactions on Systems, Man and Cybernetics, Part C, vol. 39, no. 4, pp. 440–447 (2009).
50. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. Proceedings of the Annual Meeting of the Association for Computational Linguistics, pp. 133—139 (1994).
51. Xia, Z., Wang, X., Sun, X., Wangm Q.: A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. IEEE transactions on parallel and distributed systems, 27(2):340-52 (2016).
52. Yang, J.J., Li, J.Q., Niu, Y.: A hybrid solution for privacy preserving medical data sharing in the cloud environment. Future Generation Computer Systems; 43:74-86 (2015).
53. Zhang X., Boscardin W.J., Belin T.R., Wan X., He Y., Zhang K.: A Bayesian method for analyzing combinations of continuous, ordinal, and nominal categorical data with missing values. Journal of Multivariate Analysis; 135:43-58 (2015).

A Semantic distance calculation

The *semantic distance* quantifies the difference between the meaning of two nominal values. Semantic similarity/distance measures rely on the semantic evidences gathered from knowledge bases, such as ontologies, which taxonomically structure the concepts of a domain of knowledge [7]. Formally, an *ontology* $\mathcal{O}$ is composed, at least, of a set of concepts or classes C organized in a directed acyclic graph (due to multiple inheritance) by means of is-a ($c_i < c_j$) relationships [10], as shown in Figure 2.

Measuring the semantic distance in large ontologies can be costly. In this section we discuss the computational cost of some well-known measures by relying on the concepts introduced in the following definition.

Definition 1 *Let $S(\mathbf{X^a})$ be the set of subsumers (i.e., taxonomic ancestors) of the nominal values of attribute $\mathbf{X^a}$ mapped in an ontology $\mathcal{O}$. The* least common subsumer *of $\mathbf{X^a}$, denoted by $LCS(\mathbf{X^a})$, is the most specific concept in $S(\mathbf{X^a})$. Formally,*

$$S(\mathbf{X^a}) = \{c_i \in \mathcal{O} | \forall c_j \in \mathbf{X^a} : c_j \leq c_i\};$$

$$LCS(\mathbf{X^a}) = \{c \in S(\mathbf{X^a}) | \forall c_i \in S(\mathbf{X^a}) : c \leq c_i\}.$$

The *semantic distance* is defined as a function $d_s : \mathcal{O} \times \mathcal{O} \rightarrow \mathbb{R}$ mapping a pair of concepts (corresponding to nominal values) to a real number that quantifies the difference between their meanings. According to the calculation principle employed, ontology-based measures can be divided in three families:

1. Edge-counting measures.
2. Feature-based measures.
3. Measures based on information content.

A.1 Edge-counting measures

They estimate the semantic distance between concept pairs as a function of the length of the *taxonomic path* connecting the two concepts in the ontology [33].

A well-known edge-counting measure was proposed by Wu and Palmer [50]:

$$d_{\mathrm{WP}}(c_1, c_2) = 1 - \frac{2 \times \mathrm{depth}(LCS(c_1, c_2))}{denominator}, \tag{10}$$

where $denominator = 2 \times \mathrm{depth}(LCS(c_1, c_2))$ $+\mathrm{path}(c_1, LCS(c_1, c_2))$ $+\mathrm{path}(c_2, LCS(c_1, c_2))$; $LCS(c_1, c_2)$ is the most specific subsumer of c_1 and c_2 in the ontology; $\mathrm{depth}(LCS(c_1, c_2))$ is the number of nodes in the longest taxonomic path between the $LCS(c_1, c_2)$ and the node root of the taxonomy; and $\mathrm{path}(c_i, LCS(c_1, c_2))$ is the number of taxonomic edges in the shortest taxonomic path between the two concepts.

Simplicity is the main advantage of edge-counting measures. However, they present some shortcomings: 1) if they are applied to ontologies incorporating multiple taxonomical inheritance, several taxonomical paths are not taken into account, and 2) by considering only the paths (i.e., subsumers) between the concepts, much of the taxonomical knowledge explicitly modeled in the ontology is ignored.

Assuming that concepts in the ontology are linked with their ancestors through pointers, in the worst case (comparing the two most specific concepts in the ontology that have the root node as LCS), obtaining the $LCS(c_1, c_2)$ requires running through the longest path in the taxonomy, i.e., twice the taxonomy depth D. Therefore, it takes $O(D)$ cost to compute Expression (10).

A.2 Feature-based measures

They consider the degree of overlap between the sets of ontological features of the concepts to be compared. In [40], the authors suggested measuring the semantic distance as a function of taxonomic features, i.e., as the ratio between the number of non-common taxonomic ancestors and the total number of ancestors of the two concepts:

$$d_{log\mathrm{SC}}(c_1, c_2)$$

$$= \log_2 \left(1 + \frac{|S(c_1) \cup S(c_2)| - |S(c_1) \cap S(c_2)|}{|S(c_1) \cup S(c_2)|}\right), \tag{11}$$

where $S(c_i)$ is the set of taxonomic subsumers of the concept c_i, for $i = 1, 2$. Due to the additional knowledge feature-based measures take into account (i.e., multiple direct ancestors in case of multiple inheritance), they tend to be more accurate than edge-counting measures [40].

If S is the maximum number of ancestors that a concept can have in the ontology, computing Expression (11) takes $O(S)$ cost. Notice that, for ontologies without multiple inheritance, this cost is the same as the one of edge-counting measures.

A.3 Measures based on information content

They measure the semantic distance between two concepts as the inverse of the amount of information they share in the ontology, which is represented by their LCS [35]. In particular, Lin [30] proposed as a measure the inverse of the ratio

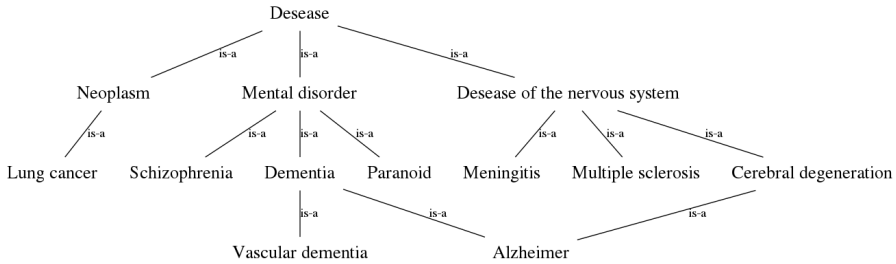


Fig. 2 Ontology extract for the "Diagnosis" concept

between the information content of the LCS of the concepts and the sum of the information content of each concept.

$$d_{\mathrm{lin}}(c_1, c_2) = 1 - \frac{IC(LCS(c_1, c_2))}{IC(c_1) + IC(c_2)}. \tag{12}$$

In [41], $IC(c)$ is intrinsically estimated within the ontology as the normalized ratio between the number of leaves (i.e., terminal hyponyms) under concept c in the taxonomy and the number of subsumers of c:

$$IC(c) = -\log\left(\frac{\frac{|\mathrm{leaves}(c)|}{|S(c)|} + 1}{|\max_\mathrm{leaves} + 1|}\right). \tag{13}$$

Thanks to IC-based measures exploiting the largest amount of ontological evidence (i.e., ancestors and leaves), they achieve better accuracy than edge-counting and feature-based measures [6].

Expression (12) requires computing the LCS of the two concepts, plus the ICs of the LCS and the concepts. Like in edge-counting measures, computing the LCS has a worst-case complexity $O(D)$. On the other hand, Expression (13) requires obtaining all the possible concepts connected to c, either subsumers of hyponyms; hence, in the worst case (i.e., when c is the root node, which subsumes all the concepts in the ontology), the IC computation takes $O(C)$ cost, where C is the total number of concepts in the taxonomy. In conclusion, Expression (12) has $O(C + D)$ computational cost. Thus, IC-based measures are not only the most accurate but also the costliest.

B Security of the scalar product protocols used

B.1 Proof of Proposition 1

Charlie receives $\boldsymbol{r}'_x$ from Alice. But $\boldsymbol{r}'_x$ can be obtained as the difference between $\hat{\boldsymbol{x}}' + \boldsymbol{k}$ and $\boldsymbol{x} + \boldsymbol{k}$, where $\boldsymbol{k}$ is an n-vector with all its components set to k and k is any real number. Hence, Charlie learns nothing about $\boldsymbol{x}$. A similar argument shows that Charlie learns nothing about $\boldsymbol{y}$.

Bob receives $\hat{\boldsymbol{x}}'$ from Alice and $\boldsymbol{r}_y$ from Charlie. Clearly, $\boldsymbol{r}_y$ contains no information on $\boldsymbol{x}$. On the other hand,

$$\hat{\boldsymbol{x}}' = \mathcal{P}_x(\hat{\boldsymbol{x}}) = \mathcal{P}_x(\boldsymbol{x} + \boldsymbol{r}_x).$$

Since $\mathbf{P}_x$ is a random permutation, the probability of Bob's learning $\hat{\boldsymbol{x}}$ from $\hat{\boldsymbol{x}}'$ is 1 over the number of permutations of $\hat{\boldsymbol{x}}$, that is

$$\frac{n_1^x! n_2^x! \ldots n_{d_x}^x!}{n!},$$

where d_x is the number of different values among the n values of $\hat{\boldsymbol{x}}$, and n_i^x is the number of repetitions of the i-th different value. Since $\hat{\boldsymbol{x}}$ is the result of adding a random vector to $\boldsymbol{x}$, it is highly unlikely that $\hat{\boldsymbol{x}}$ contains repeated values, so the probability of Bob's learning $\hat{\boldsymbol{x}}$ is very low. Furthermore, Bob does not know $\boldsymbol{r}_x$. Without knowledge of $\hat{\boldsymbol{x}}$ and $\boldsymbol{r}_x$, Bob cannot learn $\boldsymbol{x}$.

The argument on the inability of Alice to learn $\boldsymbol{y}$ is analogous.

B.2 On the security of Protocol 2

Protocol 2 is a variation of a protocol proposed in [23]. The latter protocol takes place only between Alice and Bob and there is no CLARUS proxy. Thus it differs from Protocol 2 in the last three steps, which are as follows:

4. Bob generates a random plaintext s_B, a random number r' and sends $\omega' = \omega Enc_{p_k}(-s_B; r')$ to Alice.
5. Alice computes $s_A = Dec_{s_k}(\omega') = \boldsymbol{x}^T\boldsymbol{y} - s_B$.
6. Alice and Bob simultaneously exchange the values s_A and s_B, respectively, so that both can compute $s_A + s_B = \boldsymbol{x}^T\boldsymbol{y}$.

The authors of [23] prove that, if Paillier's cryptosystem is secure, Alice cannot learn $\boldsymbol{y}$ and Bob cannot learn $\mathbf{x}$ in their protocol.

The only modification introduced by Protocol 2 is that Alice and Bob do not share their results s_A and s_B, but they send these values to CLARUS. Since neither Alice nor Bob have more information than in the protocol of [23], the security of the latter protocol is preserved in Protocol 2.



Josep Domingo-Ferrer (Fellow, IEEE) is a distinguished professor of computer science and an ICREA-Acadèmia researcher at Universitat Rovira i Virgili, Tarragona, Catalonia, where he holds the UNESCO Chair in Data Privacy and leads CYBERCAT. He received the MSc and PhD degrees in computer science from the Autonomous University of Barcelona in 1988 and 1991, respectively. He also holds an MSc degree in mathematics. His research interests are in data privacy, data security and cryptographic protocols.



David Sánchez is a Serra Hunter associate professor at Universitat Rovira i Virgili. He received his Ph.D. in Computer Science from the Technical University of Catalonia in 2008. He has participated in several national and European funded research projects and authored several papers and conference contributions. His research interests include data semantics, ontologies, data privacy and security.



Sara Ricci is a postdoctoral researcher at Brno University of Technology, Czech Republic. She received her MSc in Mathematics at University of Pisa, Italy in 2015 and her PhD in Computer Engineering and Mathematics of Security at Universitat Rovira i Virgili, Catalonia in 2018. Her research interests are in theoretical cryptography, in particular lattice-based and elliptic curve cryptography, data privacy and data security. She is also focused on the design of new privacy-preserving cryptographic protocols and their security analysis.



Mónica Muñoz-Batista completed her BS. in Computer Science at Havana University, Cuba, in 2006. She received her Master in Computer Security and Intelligent Systems from Universitat Rovira i Virgili, Tarragona, Catalonia, in 2017. Since then she has been working in the industry as a data engineer. She is interested in integrating security and privacy in the software development workflow using the Agile and DevOps methods.