Privacy-Preserving Data Analysis Workflows for eScience

Khalid Belhajjame
PSL, Université Paris-Dauphine,
LAMSADE
Paris, France
khalid.belhajjame@dauphine.fr

Noura Faci
Claude Bernard University
Lyon, France
noura.faci@univ-lyon1.fr

Zakaria Maamar
Zayed University
Dubai, United Arab Emirates
zakaria.maamar@zu.ac.ae

Vanilson Burégio
Federal Rural University of
Pernambuco
Recife, Brazil
vanilson.buregio@ufrpe.br

Edvan Soares
Federal Rural University of
Pernambuco
Recife, Brazil
edvan.soares@ufrpe.br

Mahmoud Barhamgi
Claude Bernard University
Lyon, France
mahmoud.barhamgi@univ-lyon1.fr

ABSTRACT

Computing-intensive experiences in modern sciences have become increasingly data-driven illustrating perfectly the Big-Data era's challenges. These experiences are usually specified and enacted in the form of workflows that would need to manage (i.e., read, write, store, and retrieve) sensitive data like persons' past diseases and treatments. While there is an active research body on how to protect sensitive data by, for instance, anonymizing datasets, there is a limited number of approaches that would assist scientists identifying the datasets, generated by the workflows, that need to be anonymized along with setting the anonymization degree that must be met. We present in this paper a preliminary for setting and inferring anonymization requirements of datasets used and generated by a workflow execution. The approach was implemented and showcased using a concrete example, and its efficiency assessed through validation exercises.

1 INTRODUCTION

Data-driven transformation and analysis (e.g., re-formatting data and computing statistics) are omnipresent in science and have become attractive for verifying scientists' hypotheses. This verification is dependent on dataset availability that third parties (e.g., government bodies and independent organizations) supply for re-formatting, combination, and scrutiny using what the community refers to as complex Data analysis Workflow (DWf) [9]. A DWf is a process that has an objective (e.g., discover prognostic molecular biomarkers) and a set of operations packaged (at design time) into stages (e.g., pre-process and analyze) and orchestrated (at run-time) according to data and other dependencies that the workflow designer specifies. Despite the availability of free datasets for the scientific community (e.g., Figshare[1], Dataverse[2], OpenAire[3], and DataOne[4]), data providers, in certain disciplines, are still reluctant to sharing their datasets with the community. Indeed, there is a serious concern about dataset inappropriate manipulation/misuse during experiences that could lead to sensitive-data leak and/or misuse. Although this could happen inadvertently, the consequences remain the same. As a result, some scientists/DWfs are deprived of valuable and necessary datasets due to some restrictions (e.g., access control policies) that the data providers impose. Moreover, data analysis may yield into sensitive and private data about individuals (e.g., health conditions) that were not expected during the experiment design.

Various research works (e.g., [4, 18, 26, 29–31]) have examined data outsourcing and/or sharing from a privacy perspective. We note, however, that in the context of data analysis workflows the techniques/tools that assist the designer in the specification and enforcement of data protection policies are limited. In particular, scientists need to identify the parameters in the workflows that carry sensitive datasets during their execution, and determine which anonymization method should be applied to those datasets prior to their publication. This task can be tedious, especially for large workflows.

In this preliminary work, we overcome the above issue by providing scientists with the means to automatically (i) identify the workflow parameters that are bound to sensitive data during the workflow execution, and (ii) infer the anonymity degree that needs to be applied to such datasets before releasing them publicly. We will define what we exactly mean by anonymity degree later on in Section 3.1 when introducing k-anonymity [23].

Our contributions are as follows: (*i*) an architecture of a privacy preserving workflow system that preserves the privacy of the dataset used and generated when enacting workflows, (*ii*) a method for automatically detecting sensitive dataset and setting their anonymity degree, and (*iii*) a system that implements the proposed method and experiments that showcase its efficiency using real-world scientific workflows.

The paper is organized as follows. Section 2 presents a scientific workflow from the health-care domain that we use as a running example. Section 3 presents an architecture for a privacy-preserving workflow environment, and then discusses certain necessary requirements that this environment should satisfy. Section 4 presents a new method for automatically detecting sensitive workflow parameters, and for inferring the anonymity degree that should be enforced when publishing the datasets used or generated by such parameters as a result of the workflow execution. This method is implemented and validated in Section 5 and Section 6, respectively. Section 7 presents a literature review. Conclusions are drawn in Section 8.

2 RUNNING SCENARIO

Fig. 1 exemplifies a DWf that consists of five operations ($\mathtt{op}_{i=1,5}$) connected through dataflow dependencies. Input/Output parameters are omitted for the sake of readability. This workflow's operations are as follows:

[1] figshare.com
[2] dataverse.org
[3] openaire.eu
[4] dataone.org

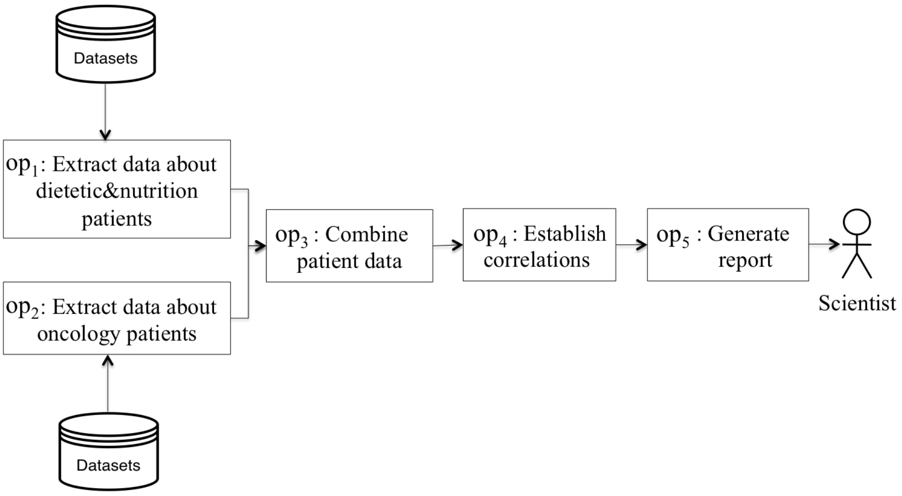
Datasets
op$_1$: Extract data about dietetic&nutrition patients
op$_2$: Extract data about oncology patients
Datasets
op$_3$: Combine patient data
op$_4$: Establish correlations
op$_5$: Generate report
Scientist

Figure 1: Example of data-analysis workflow

- op$_1$ query a dataset to get nutrition data. Table 1 is an example of this operation's output listing for each patient her average daily intake of fruits & vegetables, dairy products, meat, and dessert.
- op$_2$ retrieves oncology data about patients in terms of type of cancer and age (Table 2).
- op$_3$ combines Table 1 and Table 2's data. Specifically, it performs a natural join on nutrition and oncology information. The combination's outcome is presented in Table 3. Note that, in the general case, not all nutrition patients will be oncology patients, and *vice-versa*. We have the same patients in Tables 1 and 2 for the sake of illustration, only.
- op$_4$ implements a machine learning model that helps predict the likelihood of a patient to suffer from a particular type of cancer given his/her nutrition habits. Examples of models that can be produced are decision-based trees, neural networks, and Bayesian networks, to mention just a few.
- Finally, op$_5$ generates a final report that the scientist will examine. Such a report contains various information such as nutrition attributes that are prevalent in identifying the type of cancer the patients may suffer from, as well as information about the performance of the prediction model, e.g., accuracy, ROC curve, etc. [3].

We assume that dietetics&nutrition and oncology departments willing to share their datasets, should receive the necessary guarantees that safeguard private data from being leaked, misused, or tampered, for example. In particular, they should be able to state that their datasets are sensitive and set the anonymity degree that should be respected when anonymizing their datasets.

3 PRIVACY-PRESERVING WORKFLOW MANAGEMENT SYSTEM

This section presents the architecture of our privacy-preserving WfMS and defines the requirements that would preserve this privacy.

3.1 Overview

In Fig. 2, providers make their datasets available to a (trusted) workflow management system, that will be able to manipulate such datasets without them being anonymized. The datasets supplied can be sensitive or non-sensitive. Sensitive datasets carry personal details on individuals and therefore, should be anonymized before making them publicly available.

Initially, the datasets are transferred to a data repository that is private to the workflow system in preparation for their "cleansing" (Step 1). Once the DWf starts (Step 2), the execution engine loads the "cleansed" datasets from the private data repository (Step 3). The obtained intermediate and final datasets are stored again in this repository (Step 4). If the DWf execution reveals new insights at the scientist's discretion, she may choose to publish (some of) the datasets used and/or generated by the workflow in a public data repository (Step 6) for the benefit of the community who could explore, reuse, or even review such datasets. Prior to the release, these datasets are anonymized (Step 5).

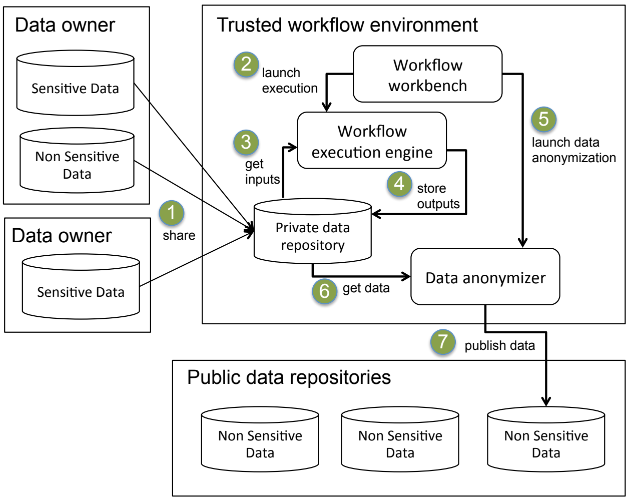

Figure 2: Chronology of operations in the WfMS

Different techniques can be used for data anonymization, e.g., generalization [27], perturbation [15], suppression [10], encryption, k-anonymization [23] and differential privacy [11]. Differential privacy is perhaps the most sophisticated method with better privacy guarantees. That said, it is not suitable for our purpose. Indeed, differential privacy is used to protect individual

Table 1: Nutrition information of patients

Patient	ID	Fruits & Veg	Dairy	Meat	Dessert
John	1	80g	33cl	150g	200g
Ahmed	2	100g	20cl	200g	150g
Ian	3	100g	50cl	300g	250g
Suzanne	4	50g	50cl	400g	300g
Yassmine	5	300g	0cl	0g	100g
Xin	6	250g	0cl	0g	100g

Table 2: Oncology information of patients

Patient	ID	Type of Cancer	Age
John	1	Melanoma	25
Ahmed	2	Lung cancer	28
Ian	3	lymphoma	35
Suzanne	4	Breast Cancer	40
Yassmine	5	Cervical cancer	65
Xin	6	Ovarian cancer	70

Table 3: Combined nutrition and oncology information of patients

Patient	ID	Age	Cancer	Fruits & Veg	Dairy	Meat	Dessert
John	1	25	Melanoma	80g	33cl	150g	200g
Ahmed	2	28	Lung Cancer	100g	20cl	200g	150g
Ian	3	35	Lymphoma	100g	50cl	300g	250g
Suzanne	4	40	Breast Cancer	50g	50cl	400g	300g
Yassmine	5	65	Cervical Cancer	300g	0cl	0g	100g
Xin	6	70	Ovarian Cancer	250g	0cl	0g	100g

privacy in the context of statistical queries. In our case, we are interested in providing users with the means to explore data produced the executions of a workflow, as opposed to generating some statistics, which is what differential privacy is mainly targeted for. Because of this, we use in the context of this paper k-anonymity. k-anonymity has been extensively studied in the database and data mining communities [12, 25]. However, its use in data analysis workflows is still limited. To illustrate k-anonymity, let us consider a dataset (d) of records referring each to an individual, e.g., age, address, and gender that could be used to reveal his identity. Such attributes are known as quasi-identifiers. (d) is k-anonymized, where (k) is an integer, if each quasi-identifier tuple occurs in at least (k) records in (d). For example, the dataset illustrated in Table 4 is 2−anonymized. Each tuple occurs at least twice in the dataset. Therefore, each patient contained in the anonymized version of (d) cannot be distinguished from at least 2 individuals. In the remainder of the paper, we use the term *anonymity degree* to refer to (k).

3.2 How to achieve a privacy-preserving WfMS?

Datasets that a workflow uses or generate are not independent of each other. In particular, the workflow operations will derive new datasets from an initial set of datasets that are eventually sensitive during the workflow execution. Dependencies between the datasets should, therefore, be considered, when setting the anonymity degree of the derived datasets based on the anonymity degree of the initial sensitive datasets. With this in mind, we present hereafter the requirements that should be met by a workflow environment to preserve the privacy of the datasets it uses and generates during the execution of workflows.

(1) The scientist should be able to specify the `DWf`'s inputs that are bound to sensitive datasets during the execution of `DWf`.
(2) Datasets' providers that submit sensitive inputs to a workflow should establish their privacy requirements in terms of degree of anonymization. This degree will then be used to anonymize such datasets prior to their publication by the `WfMS`.
(3) The dependencies between the parameters of the operations that compose the workflow should be extracted. Such dependencies allow identifying the sensitive datasets that were used to derive a given dataset, with the view to calculate the anonymity degree of the later based on the anonymity degrees of the former. Indeed, protecting a workflow's input datasets may not be sufficient to protect private information. Intermediate and final datasets that result from a workflow execution can contain sensitive data, too.
(4) A `WfMS` should assist scientists in identifying workflow parameters that are bound to sensitive datasets, and calculating the anonymity degree that needs to be enforced when publishing such datasets.

The next section illustrates how the aforementioned requirements are taken into account in the design of a privacy-preserving data workflow.

4 PRIVACY-PRESERVING DATA ANALYSIS WORKFLOWS

We begin by presenting a formal model for a DWf and then specify the inputs of the workflow that are sensitive and their anonymity degree. Finally, we present a solution that automatically identifies the sensitivity and anonymity degree of the remaining parameters of the DWf.

4.1 Workflow model definition

Workflow model. We formally define a DWf as a tuple $\langle \mathtt{DWf}_{id},\ \mathtt{OP},\ \mathtt{DL}\rangle$ where $\mathtt{DWf}_{id}$ is a unique identifier of the workflow, OP is a set of data manipulation operations (op_i) that constitute the workflow, and DL is the set of data links between these operations.

An operation op_i is defined by $\langle \mathsf{name}, \mathtt{in}, \mathsf{out}\rangle$ where name is self-descriptive, and in and out represent input and output parameters, respectively. As some output parameters could be other operations' inputs, a parameter has a unique name ($\mathsf{p}_{\mathsf{name}}$).

Let $\mathtt{IN} = \cup_{\mathsf{op}\in\mathsf{OP}}(\mathsf{op}.\mathtt{in})$ and $\mathtt{OUT} = \cup_{\mathsf{op}\in\mathsf{OP}}(\mathsf{op}.\mathtt{out})$ be the sets of all operations' inputs and outputs in a DWf, respectively. The set of data links connecting the workflow operations must then satisfy the following: $\mathtt{DL} \subseteq (\mathtt{OP}\times\mathtt{OUT})\times(\mathtt{OP}\times\mathtt{IN})$. A data link relating op_1's output $\langle \mathsf{o}, \mathsf{op}_1\rangle$ to op_2's input $\langle \mathtt{i}, \mathsf{op}_2\rangle$ is therefore denoted by the pair $\langle\langle \mathsf{o}, \mathsf{op}_1\rangle, \langle \mathtt{i}, \mathsf{op}_2\rangle\rangle$. We use $\mathtt{IN}_{\mathtt{DWf}}$ and $\mathtt{OUT}_{\mathtt{DWf}}$ to denote DWf's inputs and outputs, respectively. In this work, we consider acyclic workflows that are free of loops. It is worth noting that most of existing scientific workflow languages do not support loops [17].

Sensitive parameters. To specify that a (DWf)'s given input or output parameter carries sensitive data, we use the following boolean function:

$$\mathtt{isSensitive}(\langle \mathsf{op}, \mathsf{p}\rangle)$$

that is true if the data bound to $\langle \mathsf{op}, \mathsf{p}\rangle$ during the DWf's execution are sensitive; otherwise, false. For example, in the running example (Section 2), the two initial parameters of the workflow are sensitive in that their instances are collections of records about patients along with their nutritions and cancer histories.

Parameter anonymity degree. The execution of a DWf corresponds to a DWf instance denoted by (insWf). The anonymity degree of a DWf's parameter ($\langle\mathsf{p}, \mathsf{op}\rangle$) is defined with respect to a given DWf instance (insWf). Indeed, different instances of DWf may have as input datasets different anonymity degree requirements. For example, the owner of an input dataset used for a given workflow instance ($\mathtt{insWf}_1$) may impose a more stringent anonymity degree than the owner of an input dataset used for a different workflow instance ($\mathtt{insWf}_2$). As a result the same workflow parameter may have different anonymity degrees depending on the workflow instance in question. Due to this difference in requirement, we use the following function to specify the anonymity degree of a given parameter $\langle \mathsf{p}, \mathsf{op}\rangle$ with respect to a workflow instance insWf:

$$\mathtt{anonymity}(\langle \mathsf{p}, \mathsf{op}\rangle, \mathtt{insWf})$$

For example, $\mathtt{anonymity}(\langle \mathsf{p}, \mathsf{op}_1\rangle, \mathsf{w}_1) = 3$ specifies that the parameter $\langle \mathsf{p}, \mathsf{op}_1\rangle$ has an anonymity degree of 3 within the workflow instance w_1. Consider that the dataset (d) is bound to the parameter $\langle \mathsf{p}, \mathsf{op}_1\rangle$ within the workflow instance (w_1). Given that $\mathtt{anonymity}(\langle \mathtt{i}, \mathsf{op}_1\rangle, \mathsf{w}_1) = 3$, (d) must be anonymized before its publication. Specifically, each record (individual) in the anonymized (d) must not be distinguished from at least (2) other individuals [23].

4.2 Detecting sensitive parameters and inferring their anonymity degrees

Manual identification of a workflow's parameters that are sensitive and setting their anonymity degrees can be tedious. This becomes a serious concern when the workflow includes a large number of operations. To address this issue, we propose in this section, an approach that takes as input the sensitivity of the input parameters of the workflow (DWf) together with their anonymity degrees. It then detects the list of (intermediate and final) parameters in (DWf) that may be sensitive, and infer the anonymity degree that should be applied to the datasets bound to those parameters during the execution of the (DWf).

Parameter dependencies. Dependencies between a workflow (DWf)'s parameters is a key element to our approach. A parameter $\langle \mathsf{op}, \mathsf{p}\rangle$ depends on a parameter $\langle \mathsf{op}', \mathsf{p}'\rangle$ in a workflow (DWf), if during the execution of (DWf) the data bound to the parameter $\langle \mathsf{op}', \mathsf{p}'\rangle$ contribute to or influence the data bound to the parameter $\langle \mathsf{op}', \mathsf{p}'\rangle$[5].

Parameter dependencies can be specified by examining the workflow specification (DWf)[6]. Given a workflow (DWf), the dependencies between its parameters are inferred as follows:

- Given an operation (op) that belongs to (DWf), we can infer that the outputs of (op) depends on its inputs. Consider for example that $\langle \mathtt{i}, \mathsf{op}\rangle$ and $\langle \mathsf{o}, \mathsf{op}\rangle$ are an input and output of (op). We can infer that $\langle \mathsf{o}, \mathsf{op}\rangle$ depends on $\langle \mathtt{i}, \mathsf{op}\rangle$, which we write:

$$\mathsf{dependsOn}(\langle \mathsf{o}, \mathsf{op}\rangle, \langle \mathtt{i}, \mathsf{op}\rangle)$$

- If the workfow (DWf) contains a data link connecting an output $\langle \mathsf{op}, \mathsf{o}\rangle$ to an input $\langle \mathsf{op}, \mathtt{i}\rangle$, then we infer that $\langle \mathsf{op}, \mathtt{i}\rangle$ depends on $\langle \mathsf{op}, \mathsf{o}\rangle$, i.e., $\mathsf{dependsOn}(\langle \mathsf{o}, \mathsf{op}\rangle, \langle \mathtt{i}, \mathsf{op}'\rangle)$. This is because the data bound to $\langle \mathsf{o}, \mathsf{op}\rangle$ during the workflow execution is a copy of the data bound to $\langle \mathtt{i}, \mathsf{op}'\rangle$.

We also transitively derive dependencies between the operation parameters of a workflow based on the following rules:

$$\begin{aligned} R_1 :\ & \mathsf{dependsOn}^*(\langle \mathsf{p}, \mathsf{op}\rangle, \langle \mathsf{p}', \mathsf{op}'\rangle)\ :-\ \mathsf{dependsOn}(\langle \mathsf{p}, \mathsf{op}\rangle, \langle \mathsf{p}', \mathsf{op}'\rangle) \\ R_2\ & \mathsf{dependsOn}^*(\langle \mathsf{p}, \mathsf{op}\rangle, \langle \mathsf{p}', \mathsf{op}'\rangle)\ :-\ \mathsf{dependsOn}^*(\langle \mathsf{p}, \mathsf{op}\rangle, \langle \mathsf{p}'', \mathsf{op}''\rangle), \\ & \qquad \mathsf{dependsOn}^*(\langle \mathsf{p}'', \mathsf{op}''\rangle, \langle \mathsf{p}', \mathsf{op}'\rangle) \end{aligned}$$

Applying the above rules to our example workflow, we conclude for instance, that $\mathsf{dependsOn}^*(\langle \mathsf{o}, \mathsf{op}_3\rangle, \langle \mathtt{i}, \mathsf{op}_2\rangle)$, where i and o are parameter names.

Detecting sensitive parameters. We use parameter dependencies to assist the workflow designer identify the intermediate and final parameters that may be sensitive. Specifically, a parameter $\langle \mathsf{p}', \mathsf{op}'\rangle$ that is not an input to the workflow, i.e., $\langle \mathsf{p}', \mathsf{op}'\rangle \notin IN_{DWf}$, may be sensitive if it depends on a workflow input that is known to be sensitive, i.e.,

$$\exists \langle \mathtt{i}, \mathsf{op}\rangle \in \mathtt{IN}_{\mathtt{DWf}}\ \mathtt{s.t.}\ \mathtt{sensitive}(\mathtt{i}, \mathsf{op}) \wedge \mathsf{dependsOn}^*(\langle \mathsf{p}', \mathsf{op}'\rangle, \langle \mathtt{i}, \mathsf{op}\rangle)$$

Note that we say that $\langle \mathsf{p}', \mathsf{op}'\rangle$ *may be* sensitive. This is because an operation that consumes sensitive datasets may produce

[5] The notion of contribution and influence are in line with the derivation and influence relationship defined by the W3C PROV recommendation [19].

[6] Parameter dependencies correspond to what is referred to in the scientific workflow community by retrospective provenance. This is because such dependencies can be inferred from the workflow specification as opposed to other kinds of information, e.g., execution log, which can only be obtained retrospectively once the workflow execution terminates.

non-sensitive datasets. For example, op_5 in Fig. 1 generates non-sensitive information although its outputs are sensitive inputs of the workflow. The output of such an operation is a report that is free from information about individual patients.

Inferring anonymity degree. In addition to assisting the designer identify sensitive intermediate and final output parameters, we also infer details about the anonymity degree that should be applied to dataset instances of those sensitive parameters. To illustrate this, consider that $\langle \mathsf{p}', \mathsf{op}' \rangle$ is a sensitive intermediate or final output parameter. The anonymity degree of such a parameter given a workflow execution `insWf` can be defined as the maximum degree of the sensitive datasets that are used as input to the workflow and that contribute to the datasets instances of $\langle \mathsf{p}', \mathsf{op}' \rangle$. Taking the maximum anonymity degree of the contributing inputs ensures that the anonymity degrees imposed on such inputs is honored by the dependent parameter in question. That is:

$$\begin{aligned} &\mathtt{anonymity}(\langle \mathsf{p}', \mathsf{op}' \rangle, \mathtt{insWf}) = \\ &\max(\{\mathtt{anonymity}(\langle \mathsf{i}, \mathsf{op} \rangle, \mathtt{insWf})\ \mathtt{s.t.}\ \mathtt{sensitive}(\langle \mathsf{i}, \mathsf{op} \rangle) \\ &\qquad\qquad \wedge\ \mathtt{dependsOn}^*(\langle \mathsf{p}', \mathsf{op}' \rangle, \langle \mathsf{i}, \mathsf{op} \rangle)\}) \end{aligned}$$

Once anonymity degree is computed, the `WfMS` uses an anonymization algorithm proposed in the literature like Mondarian [16] before publishing the datasets used and generated as a result of the workflow execution.

5 IMPLEMENTATION

Fig. 3 depicts the system architecture implementing our privacy-aware workflow approach. Not all the components reported in Fig. 2 have been implemented. Indeed, instead of reinventing the wheel, we make use of some existing popular scientific workflow systems [6, 14, 28]. We have, therefore, focused on implementing the `Anonymizer` component which consists of the following modules.

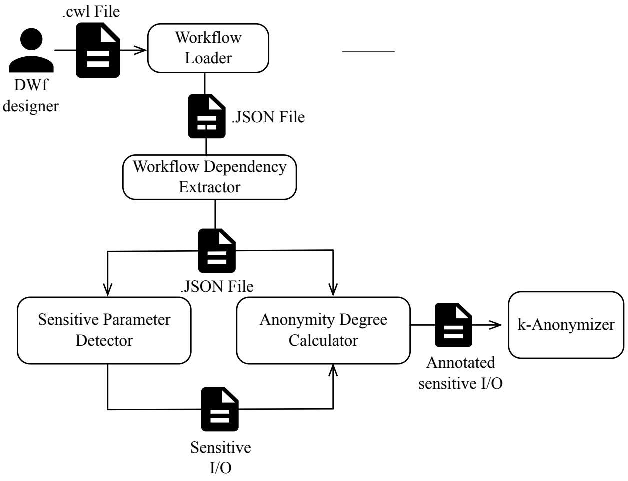

Figure 3: System's technical architecture

Workflow Loader. To ensure our system interoperability with existing workflow systems, we decided on handling the workflows specified in the Common Workflow Language (CWL[7]). CWL has recently gained momentum and is currently supported by major scientific workflow systems. The Workflow Loader module converts a CWL workflow into an equivalent JSON format, which is used internally by our system.

Workflow Dependency Extractor. This module is used to identify the dependencies between workflow parameters. It takes as input a workflow specification and produces as output a list of pairs of parameters $\langle \mathsf{p}_1, \mathsf{p}_2 \rangle$ where p_2 depends on p_1. Let us consider our running example of Section 2. Applying the Workflow Dependency Extractor to this workflow reveals, for instance, that the input of op_3 depends on the inputs of op_1 and op_2, among other dependencies.

Sensitive Parameter Detector. This module identifies workflow parameters that *may be* sensitive. It takes as input the workflow input that is indicated (by the user or workflow's author) as sensitive, and the parameter dependencies produced by Workflow Dependency Extractor. It produces as output a list of parameters that may be sensitive. Let us consider our running example along with the inputs of operations op_1 and op_2 that the scientist sets as sensitive because of handling personal information. The Sensitive Parameter Detector concludes that the remaining parameters of the workflow *may be* sensitive. Indeed, the workflow's all intermediate and final parameters depend on op_1 and op_2 inputs. It is worth underlining that the $\mathtt{sensitive} - \mathtt{detector} - \mathtt{parameter}$ identifies the parameters that *may* be sensitive. In other words, not all the parameters that are returned by this module will be flagged as sensitive. This is the case for the outputs of $\mathsf{op}_4 : \mathtt{establish\ correlations}$ and $\mathsf{op}_5 : \mathtt{generate\ report}$, which, respectively, deliver a machine learning model and report that are free of any personal detail, and as such do not need to be anonymized. Note, however, that if a parameter is not returned by the $\mathtt{sensitive} - \mathtt{detector} - \mathtt{parameter}$, then that means that such parameter is definitely not sensitive.

Anonymity Degree Calculator. This module computes the anonymity degree of a workflow's sensitive parameters. To this end, it establishes the anonymity degree that must be met by a sensitive parameter that is not a workflow's initial input. Indeed, the anonymity degree of the initial parameters of the workflow as a whole is specified by the user. It takes as input the anonymity degree of each input of the workflow that is known to be sensitive, the list of parameter dependencies that are produced by the Workflow Dependency Extractor, and the list of workflow parameters that are identified as sensitive by the Sensitive Parameter Detector. It then produces the anonymity degree of each sensitive parameter of the workflow (other than the initial workflow inputs). Let us consider the nutrition and oncology departments that state that their data should be 2-anonymized before publication. By using the $\mathtt{anonymity} - \mathtt{degree} - \mathtt{calculator}$, we establish that the anonymity degree $\mathsf{op}_{1,2,3}$'s outputs should be equal to 2.

k-Anonymizer. Once the anonymity degrees of the parameters are produced, the k-Anonymizer is enabled to anonymize the dataset instances of these parameters during a workflow execution. The anonymization operation is out of the scope of this paper. Instead, existing k-anonymization algorithms (e.g., ARX [20], an open source data anonymization tool) can be used. For instance, Tables 4, 5, and 6 show the data obtained by anonymizing the data of Tables 1, 2, and 3, respectively, with the anonymity degree $\mathsf{k} = 2$.

[7] https://github.com/common-workflow-language/common-workflow-language

Table 4: Anonymized nutrition information of patients with k = 2.

Patient	ID	Fruits & Veg	Dairy	Meat	Dessert
*	*	80g ≤ Fruits ≤ 100g	20cl ≤ Dairy < 40cl	100g ≤ Meat ≤ 200g	100g < Dessert ≤ 200g
*	*	80g ≤ Fruits ≤ 100g	20cl ≤ Dairy < 40cl	100g ≤ Meat ≤ 200g	100g < Dessert ≤ 200g
*	*	0g ≤ Fruits ≤ 50g	40cl < Dairy ≤ 50cl	200g < Meat ≤ 400g	200g < Dessert ≤ 300g
*	*	0g ≤ Fruits ≤ 50g	40cl < Dairy ≤ 50cl	200g < Meat ≤ 400g	200g < Dessert ≤ 300g
*	*	200g ≤ Fruits ≤ 300g	0cl ≤ Dairy < 20cl	0g < Meat ≤ 50g	0g < Dessert ≤ 100g
*	*	200g ≤ Fruits ≤ 300g	0cl ≤ Dairy < 20cl	0g < Meat ≤ 50g	0g < Dessert ≤ 100g

Table 5: Anonymized oncology data of patients with k = 2.

Patient	ID	Type of Cancer	Age
*	*	Melanoma	20 ≤ Age ≤ 30
*	*	Lung cancer	20 ≤ Age ≤ 30
*	*	lymphoma	30 < Age ≤ 40
*	*	Breast Cancer	30 < Age ≤ 40
*	*	Cervical cancer	60 ≤ Age ≤ 70
*	*	Ovarian cancer	60 ≤ Age ≤ 70

Table 6: Combined nutrition and oncology information of patients anonymized with k = 2

Patient	ID	Age	Type of Cancer	Fruits & Veg	Dairy	Meat	Dessert
*	*	20 ≤ Age ≤ 30	Melanoma	80g ≤ Fruits ≤ 100g	20cl ≤ Dairy < 40cl	100g ≤ Meat ≤ 200g	100g < Dessert ≤ 200g
*	*	20 ≤ Age ≤ 30	Lung Cancer	80g ≤ Fruits ≤ 100g	20cl ≤ Dairy < 40cl	100g ≤ Meat ≤ 200g	100g < Dessert ≤ 200g
*	*	30 < Age ≤ 40	Lymphoma	0g ≤ Fruits ≤ 50g	40cl < Dairy ≤ 50cl	200g < Meat ≤ 400g	200g < Dessert ≤ 300g
*	*	30 < Age ≤ 40	Breast Cancer	0g ≤ Fruits ≤ 50g	40cl < Dairy ≤ 50cl	200g < Meat ≤ 400g	200g < Dessert ≤ 300g
*	*	60 ≤ Age ≤ 70	Cervical Cancer	200g ≤ Fruits ≤ 300g	0cl ≤ Dairy < 20cl	0g < Meat ≤ 50g	0g < Dessert ≤ 100g
*	*	60 ≤ Age ≤ 70	Ovarian Cancer	200g ≤ Fruits ≤ 300g	0cl ≤ Dairy < 20cl	0g < Meat ≤ 50g	0g < Dessert ≤ 100g

6 VALIDATION

For validation purposes, different experiments were carried out upon the system described in Section 5. 20 different CWL workflows[8] (500 executions per workflow) have been used so that parameters like loading times, identifying parameter dependencies and sensitive parameters, and computing anonymity degree have been assessed. Number of operations, sensitive inputs, and anonymity degrees highlight the differences between these workflows.

For each workflow, we compute the minimum, maximum, and average overhead due to workflow loading, parameter dependency extraction, sensitive parameter identification, and anonymity degree computation, across the 10K executions. On the one hand, Fig. 4 is for workflow loading. The minimum time is nearly $0ms$ in most cases, which can hardly be seen on the chart. The average time is almost the same for all workflows; i.e., approximately equal to $0.1ms$. Regarding the maximum time, it varies between $1ms$ and $3ms$, which are small numbers. On the other hand, Fig. 5 is for parameter dependency extraction. Required minimum and average time can be hardly seen on the chart; in fact, the extraction of dependencies is instantaneous in most cases. For the required maximum time, it is less than $0.2ms$ for most workflows. However, 3 outliers have been identified, Workflows 2, 13, and 20, that take almost $15ms$ in the worst case. This can be explained by the fact that dependency extraction is influenced by the number of input and output parameters the workflow has. The examination of Workflows 2, 13, and 20 revealed that they have a larger number of outputs compared with the rest of workflows.

Regarding the overhead due to sensitive parameter detection and anonymity degree calculation, it is almost instantaneous for all workflows, and therefore there was no need to show the charts for them (also due to limited space). In summary, the result of the experiment we ran are encouraging and show that the overhead due to the solution can bearly be noticed.

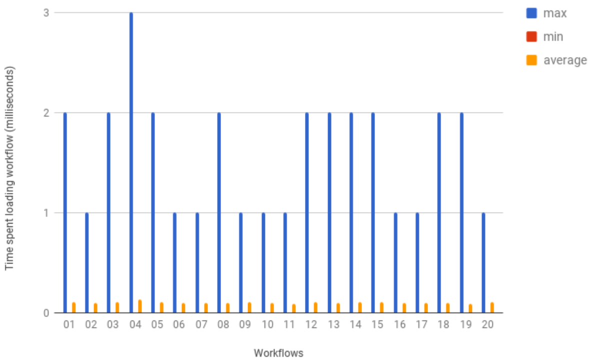

Figure 4: Overhead due to workflow loading

[8] view.commonwl.org/workflows

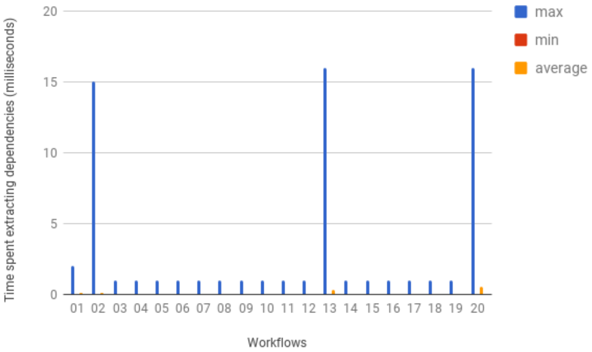

Figure 5: Overhead due to parameter dependency extraction

7 RELATED WORK

Privacy concerns in the context of workflows have been examined by a number of proposals. We present in this section these proposals and conclude the section by discussing how our work advances the state of the art.

In [13], Gil et al. address the issue of data privacy in the context of DWfs. To this end, they propose an ontology that preserves this privacy along with enforcing access control over data with respect to a given set of access permissions. The ontology specifies eligible privacy-preserving policies (e.g., generalization and anonymization) per DWf's input/output parameter. To support privacy policy enforcement in DWfs, a framework was developed to represent policies as a set of elements that include applicable context, data usage requirement, privacy protection requirement, and corrective actions if the policy is violated.

In [7], Chebbi and Tata propose a workflow reduction-based abstraction approach for workflow advertisement purposes. The approach reduces a workflow inter-visibility using 13 rules that depend on dependencies between operations in the workflows along with the operation types (i.e., internal *versus* external.

In [24], Teepe et al. analyze a business workflow specification to determine the properties that would achieve privacy protection of a company's partners and customers. To this end, they represent workflows as Color-X diagrams and then translate them into Prolog so that privacy relevant properties over data are analyzed, e.g., need-to-know principle. This analysis inspects the messages sent by all employees involved in the business workflow to detect "gossipy" employees, i.e., those who exchange more information than they are asked for.

In [21], Sharif et al. introduce MPHC standing for Multiterminal Cut for Privacy in Hybrid Clouds framework to minimize the cost of executing workflows while satisfying both task/data privacy and deadline/budget constraints. In [22], Sharif et al. extend MPHC with Bell-LaPadula rules so that all data and tasks are deployed over hybrid cloud instances with greater or equal privacy levels.

In [2], Alhaqbani et al. propose a privacy-enforcement approach for business workflows based on 4 requirements: (i) capture the *subject* (i.e., data owner)'s privacy policy during the workflow specification on top of the privacy policies defined by the workflow administrator, (ii) define data properties (i.e., hide and generalize) linked to private data so that these properties influence the workflow engine to protect data as per the *subject*'s privacy policy, (iii) allocate work while preserving privacy, i.e., assign the task referring to some manipulation of data, to the employee who has the lowest restriction level according to the *subject*'s privacy policy, and (iv) keep the subject informed about any attempt for accessing his/her data.

In [5], Barth et al. present a privacy-policy violation detection approach based on execution logs of business processes. The aim is to identify a set of employees potentially responsible for privacy breach. The authors introduce two types of compliance: strong and weak. An action is strongly compliant with a privacy policy given a trace if there exists an extension of the trace that contains the action and satisfies the policy. An action is weakly compliant with a policy given a trace if the trace augmented with the action satisfies the present requirements of the privacy policy.

In [8], Davidson et al. discuss privacy-preserving management of provenance-aware workflow systems. The authors first formalize the privacy concerns: (i) *data privacy* that requires outputs of the workflow's modules (*aka* operations) should not reveal to users without an access privilege, (ii) *module privacy* that requires the functionality of this module is not revealed, and (iii) *structural privacy* that refers to hiding the data flow's structure in the given execution.

The aforementioned proposals can be classified into two categories. Those that preserve the privacy of tasks (operations) of workflows. This is exemplified in the works by Barth et al. [5] and Davidson et al. [8]. And those that preserve the privacy of data that workflows manipulate at run-time. This is exemplified with the works of Gil et al. [13], Teepe et al. [24], and Alhaqbani et al. [2]. Contrarily, the work of Sharif et al. [21] addresses the privacy of both task and data. In the context of our work, we are concerned with the privacy of workflow data and hence, is in line with the second category of proposals. However, achieving this privacy requires that the workflow designer manually identifies sensitive workflow parameters and sets the degree to which the datasets bound to those parameters need to be anonymized. We have taken care of both aspects in our work.

8 CONCLUSION

We presented an approach for preserving privacy in the context of scientific workflows that heavily rely on large datasets. We have shown how data plays a role in (*i*) identifying sensitive operation parameters in the workflow and (*ii*) deriving the anonymity degree that needs to be enforced when publishing the datasets instances of these parameters. To the best of our knowledge, this is the first work that looks into these aforementioned items (i) and (ii). We have also implemented a system that showcases our solution and conducted some experiments for efficiency needs. This work opens up opportunities for more research in the field of anonymization of workflow data. In this respect, our ongoing work includes investigating the applicability of our solution to anonymization techniques, other than k-anonymity, e.g., l-diversity and t-closeness [1].

REFERENCES

[1] [n. d.]. A critique of k-anonymity and some of its enhancements.

[2] B. Alhaqbani, M. Adams, C. J. Fidge, and A. H. M. ter Hofstede. 2013. *Privacy-Aware Workflow Management.* Springer, Dortmund, Germany, 111–128.

[3] E. Alpaydin. 2014. *Introduction to Machine Learning* (2nd ed.). The MIT Press, Cambridge, Massachusset, USA.

[4] G. Antoniou, M. Baldoni, P. A. Bonatti, W. Nejdl, and D. Olmedilla. 2007. In *Secure Data Management in Decentralized Systems.* Springer, 169–216.

[5] A. Barth, J. C. Mitchell, A. Datta, and S. Sundaram. 2007. Privacy and Utility in Business Processes. In *Computer Security Foundations Symposium – CSF, 6-8 July.* IEEE, Venice, Italy, 279–294.

[6] S. P. Callahan, J. Freire, E. Santos, et al. 2006. Vistrails: Visualization meets data management. In *SIGMOD*. ACM Press, Chicago, IL, USA, 745–747.

[7] I. Chebbi and S. Tata. 2007. Workflow Abstraction for Privacy Preservation. In *International Conference on Web Information Systems Engineering – WISE, December 3*. Springer Link, Nancy, France, 166–177.

[8] S. B. Davidson, S. Khanna, V. Tannen, S. Roy, Y. Chen, T. Milo, and J. Stoyanovich. 2011. Enabling Privacy in Provenance-Aware Workflow Systems. In *Biennial Conference on Innovative Data Systems Research, January 9-12*. CIDR Conference, Asilomar, CA, USA, 215–218.

[9] E. Deelman, D. Gannon, M. Shields, and I. Taylor. 2009. Workflows and e-Science: An Overview of Workflow System Features and Capabilities. *Future Generation Computer Systems* 25, 5 (2009), 528–540.

[10] R. B. Dolby, G. Harvey, N. P. Jenkins, and R. Raviraj. 2000. Data suppression and regeneration. (2000). US Patent 6,038,231.

[11] Cynthia Dwork. 2006. Differential Privacy. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*. Springer, 1–12. https://doi.org/10.1007/11787006_1

[12] A. Friedman, R. Wolff, and A. Schuster. 2008. Providing k-anonymity in data mining. *The VLDB Journal* 17, 4 (2008), 789–804.

[13] Y. Gil, W.K. Cheung, V. Ratnakar, and K-K. Chan. 2007. Privacy Enforcement in Data Analysis Workflows. In *AAAI Workshop on Privacy Enforcement and Accountability with Semantics (PEAS)*. AAAI, Busan, Korea, 41–48.

[14] Y. Gil, V. Ratnakar, J. Kim, et al. 2011. Wings: Intelligent Workflow-Based Design of Computational Experiments. *Intelligent Systems* 26, 1 (2011), 62–72.

[15] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. 2003. On the privacy preserving properties of random data perturbation techniques. In *International Conference on Data Mining – ICDM'03*. IEEE, Melbourne, Florida, USA, 99–106.

[16] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. 2006. Mondrian Multidimensional K-Anonymity. In *International Conference on Data Engineering, ICDE 2006, 3-8 April*. IEEE, Atlanta, GA, USA, 25.

[17] J. Liu, E. Pacitti, P. Valduriez, and M. Mattoso. 2015. A Survey of Data-Intensive Scientific Workflow Management. *J. Grid Comput.* 13, 4 (2015), 457–493.

[18] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. 2007. l-diversity: Privacy beyond k-anonymity. *Transactions on Knowledge Discovery from Data (TKDD)* 1, 1 (2007), 3.

[19] P. Missier, K. Belhajjame, and J. Cheney. 2013. The W3C PROV family of specifications for modelling provenance metadata. In *Joint 2013 EDBT/ICDT Conferences, EDBT '13 Proceedings, Genoa, Italy, March 18-22, 2013*. ACM press, 773–776.

[20] F. Prasser, F. Kohlmayer, R. Lautenschläger, and K. Kuhn. 2014. ARX - A Comprehensive Tool for Anonymizing Biomedical Data. In *American Medical Informatics Association Annual Symposium*. AMIA.

[21] S. Sharif, J. Taheri, A. Y. Zomaya, and S. Nepal. 2013. MPHC: Preserving Privacy for Workflow Execution in Hybrid Clouds. In *International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT, December 16-18*. Sponsored by IEEE, Taipei, Taiwan, 272–280.

[22] S. Sharif, P. Watson, J. Taheri, S. Nepal, and A. Y. Zomaya. 2017. Privacy-Aware Scheduling SaaS in High Performance Computing Environments. *IEEE Trans. Parallel Distrib. Syst.* 28, 4 (2017), 1176–1188.

[23] L. Sweeney. 2002. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, 05 (2002), 557–570.

[24] W. Teepe, R.P. van de Riet, and M.S. Olivier. 2003. WorkFlow Analyzed for Security and Privacy in using Databases. *Journal of Computer Security* 11, 3 (2003), 271–282.

[25] M. Terrovitis, N. Mamoulis, and P. Kalnis. 2008. Privacy-preserving anonymization of set-valued data. *VLDB Endowment* 1, 1 (2008), 115–125.

[26] B. Wang, B. Li, and H. Li. 2012. Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud. In *Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing*. IEEE Computer Society, 295–302.

[27] K. Wang, P. S. Yu, and S. Chakraborty. 2004. Bottom-up generalization: A data mining solution to privacy protection. In *International Conference on Data Mining – ICDM'04*. IEEE, Brighton, UK, 249–256.

[28] K. Wolstencroft, R. Haines, D. Fellows, et al. 2013. The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. *Nucleic acids research* (2013), W557âĂŞW561.

[29] S. Guadie Worku, C. Xu, J. Zhao, and X. He. 2014. Secure and Efficient Privacy-preserving Public Auditing Scheme for Cloud Storage. *Comput. Electr. Eng.* 40, 5 (2014), 1703–1713.

[30] X. Xiao and Y. Tao. 2006. Anatomy: Simple and effective privacy preservation. In *Proceedings of the 32nd international conference on Very large data bases*. VLDB Endowment, 139–150.

[31] M. Yiu, G. Ghinita, C. Jensen, and P. Kalnis. 2009. Outsourcing Search Services on Private Spatial Data. In *Proceedings of the 25th International Conference on Data Engineering, ICDE 2009, March 29 2009 - April 2 2009, Shanghai, China*. IEEE, 1140–1143.