A Security Framework for Scientific Workflow Provenance Access Control Policies

Fahima Amin Bhuyan, *Student-Member, IEEE,* Shiyong Lu, *Senior Member, IEEE,*
Robert Reynolds, *Senior Member, IEEE,* Jia Zhang, *Senior Member, IEEE,*
and Ishtiaq Ahmed, *Student-Member, IEEE*

Abstract—The notion of collaborative scientific workflow is coined to address the increasing need for collaborative data analytics. In collaborative environments, access control policies are necessary for controlling the sharing of workflows, data products, and provenance information among collaborating parties. In particular, the protection of workflow provenance is critical because it often encodes the detailed protocol of a scientific experiment and carries the intellectual property of the respective stakeholders. In addition, since scientific workflows often evolve quickly, the corresponding access control policies for workflow provenance have to evolve as well. It is important to ensure that the evolution of workflow provenance access control policies maintain certain properties, in order to guarantee the correctness and performance of the corresponding policy enforcement. In this paper, we 1) propose a role-based access control model for scientific workflow provenance; 2) define three quality requirements for scientific workflow provenance access control policies - consistency, completeness, and conciseness; 3) develop a mechanism mapping from specifications of workflows to their counterparts in a provenance that preserves such quality properties, and 4) conduct a case study on a scientific workflow for autism behavioral data analysis that demonstrates the feasibility of our proposed analysis algorithms.

Index Terms—Provenance; access control policy; policy quality; security view of provenance.

1 Introduction

Provenance refers to information about the history, origin, derivation, and context of data. Provenance is useful in interpreting an analytical result, repeating a scientific discovery, and tracing the source of errors in data. Provenance is also useful to help answer lineage queries and to decide the trustworthiness of a data product. Therefore, provenance management has become critical in various data systems such as database, workflow, and web information systems [1], [2], [3]. All major scientific workflow systems [4], [5], [6], [7], [8] support provenance. The past few years have also witnessed great efforts on provenance standardization, including Open Provenance Model (OPM) [9], [10] and PROV [11], and active community engagement in the provenance challenge series [12].

It has been well recognized that the provenance security problem is critical for modern scientific workflow systems [13], [14], [15], [16]. Unauthorized access to provenance information might disclose confidential information about related data products. The code for collecting, querying, and mining provenance can be compromised, forged, or replayed by intruders. The linkages among data products, provenance, and workflow specifications can be severed or forged in a malicious environment. Compromised provenance can lead to misinterpretation of analytic results, unintentional errors, and compromise the confidentiality of related data sets. As science becomes more and more interdisciplinary and collaborative, the notion of *collaborative scientific workflow* was coined to address the increasing need for collaborative data analytics leveraging the scientific workflow paradigm [17], [18], [19], [20], [21]. In such collaborative environments, adequate access control policies are necessary to control the sharing of workflows, data products, and provenance information among collaborating parties [13]. In this research, we focus on the secrecy of provenance, so that provenance is accessible only to authorized collaborative parties. This is important because provenance often encodes the detailed protocol of a scientific experiment and constitutes the intellectual property of the respective stakeholders. Our starting point is existing access control mechanisms serving for the protection of the confidentiality of scientific workflow provenance [13], [15].

While business workflows are relatively stable over time, scientific workflows tend to evolve rapidly since many scientists frequently generate, explore, and test various hypotheses about a scientific problem simultaneously [22]. For example, an existing workflow w_1 might be extended with additional sub-workflows or turned into workflow w_{11} that performs a more advanced scientific analysis. The sub-workflow w_{11} can be further decomposed into w_{111} and w_{112} that contain additional sub-workflows, tasks, and data channels. All such workflows can be used simultaneously in order to explore different hypotheses or to perform various but related scientific analysis. As a result, it is important to evolve the corresponding access control policies

- *Fahima Amin Bhuyan is a student in the Department of Computer Science, Wayne State University, Detroit, MI. E-mail: fahima.amin@wayne.edu.*
- *Shiyong Lu is with the Department of Computer Science, Wayne State University, Detroit, MI. E-mail: shiyong@wayne.edu.*
- *Robert Reynolds is with the Department of Computer Science, Wayne State University, Detroit, MI. E-mail: robert.reynolds@wayne.edu .*
- *Ishtiaq Ahmed is with the Department of Computer Science, Wayne State University, Detroit, MI. E-mail: ishtiaq@wayne.edu .*
- *Jia Zhang is with the Department of Computer Science, Carnegie Melon University Silicon Valley, Mountain View, CA. E-mail: jia.zhang@sv.cmu.edu.*

Manuscript received April 12, 2018; revised May 31, 2019.

simultaneously as well. In dealing with such large sets of evolving policies, manually checking the quality of each policy becomes impractical. Instead, automated analysis algorithms for access control policies of scientific workflow provenance are necessary in order to ensure the correctness and performance of corresponding policy enforcement.

The contributions of this paper are four-fold. 1) We propose a role-based access control model for scientific workflow provenance management. 2) We define three quality requirements for scientific workflow provenance access control polices - consistency, completeness, and conciseness. 3) We develop a mapping from specifications over workflows to their counterparts in the provenance, and prove that such a mapping preserves these quality properties. 4) We conduct a case study on a scientific workflow for autism behavioral data analysis and demonstrate the feasibility of our proposed analysis algorithms.

The rest of the paper is organized as follows. Section 2 defines the basic terminologies of the security framework. Section 3 sketches the life span of a provenance security policy. Section 4 presents our ProvSec prototype and a case study in the autism domain, which is continued in Section 9. Section 5 presents a formal security scientific workflow specification mechanism for task, port and data channel with proposed algorithms of access control policies. Section 6 formalizes a mapping between workflows to security views and presents security view for provenance. Section 7 presents an algorithm that analyzes the policies with respect to policy quality requirements in order to determine whether the evolving policies are consistent, complete and concise. This section also provides proof of holding policy quality requirements for provenance. Section 8 presents policy evolution based on quality requirements. Section 10 reviews related work. Finally, Section 11 concludes the paper and points out some possible directions for future work.

1.1 Security in Workflow vs. Security in Provenance

Since scientific workflow captures the intellectual property of scientific experiments and composition of various computational services into workflow, workflow security protects the access to those workflow tasks and data. There can be differences in perspective in terms of how to provide access control policies in a workflow. Based upon scientists' preferences, one can only publish source data and final scientific results, but not the scientific workflow in between. Whereas, for other scientists, they can publish source data, scientific results and all the workflow used, but keep the parameter settings as a secret for the workflow.

Security in provenance is a major aspect of scientific workflow. As provenance captures all of the derivation history including original data sources, intermediary data products and all the steps involved to produce those data products, imposing security means implementing access control policies on those data products (source, intermediary, final) and the dependencies among them. Provenance access control policies can be applied and used in order to release provenance information of source data, scientific results and parameter settings, but still can hide intellectual property of certain provenance information.

Access control policies can be applied on composite tasks or sub-workflows of provenance at different abstraction levels, where users are only allowed to access provenance information based on their requirements and preferences [23], [24], [25], [26], [27]. In provenance security, there are no foundational models yet to define and relate security goals such as availability, confidentiality and privacy. In order to make meaningful progress on these issues, a foundational model is on demand.

1.2 Examples for Importance of Provenance Security

The importance of provenance security can be illustrated with several examples:

- Without proper provenance or in circumstances of provenance failure, information could be misinterpreted. An old news article can bring misinterpretation when the date of information is not stored and can result in sudden economic loss [28].
- Lack of information makes it difficult for reviewers to evaluate the contributions of a submitted scientific manuscript. Keeping provenance of those scientific discoveries aims to help keep transparency and repeatability [28].
- Unintentional release of provenance information may violate privacy and confidentiality. This may happen when provenance information is employed in written documents describing a project.
- At the end of the process of peer-review, the content of the reviews are delivered to the authors, but the identities of the reviewers are not. Here, the reviews (data) are public, but who wrote the reviews (provenance) remains confidential.
- In a letter of recommendation, the subject of the letter is not allowed to know the content, but allowed to know the author. Here the content of letter (data) is confidential, but the author of the letter (provenance) is public.

2 Provenance Security Framework

In this section, we present a provenance security framework, where formal and precise security properties, like confidentiality, privacy, and availability, are managed to enforce suitable and desirable security policies.

In the era of big data, scientific workflows have become essential to automate scientific experiments and guarantee repeatability [29], [30], [31]. Increasingly in many scientific domains, such as health and medicine [32], personalization in information processing has become a key to success. Hence, access control protocols in scientific workflows have become a prerequisite. Workflow provenance systems, while making the management of data and process lineage possible, also need to adhere to the access control protocol inherent in the scientific workflows. In this paper, we propose a security scientific workflow specification mechanism using role-based access control policies. We demonstrate how policies are inherited by the workflow provenance system. Then, we characterize the desirable properties of role-based access control protocols in scientific workflows, and delineate how the properties are maintained in the workflow provenance systems as well.

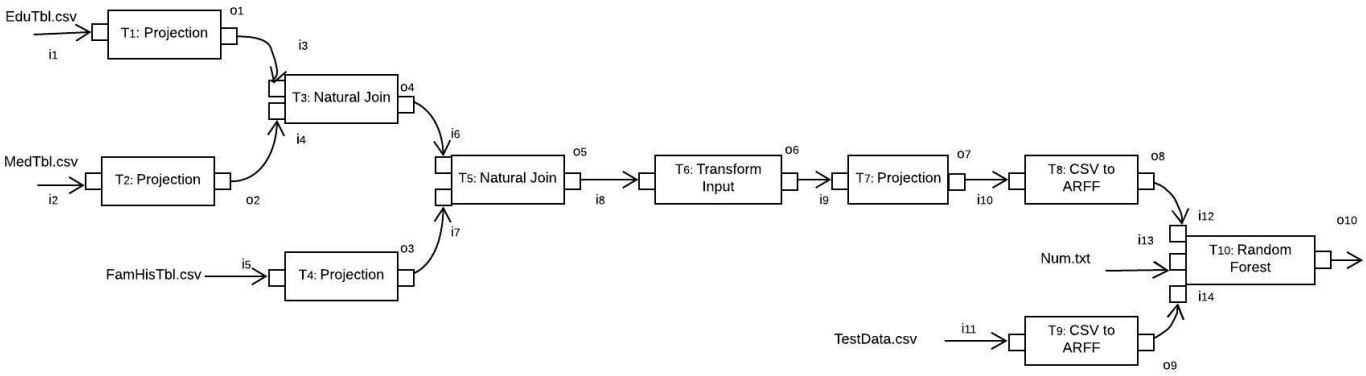

Fig. 1: Autism Workflow.

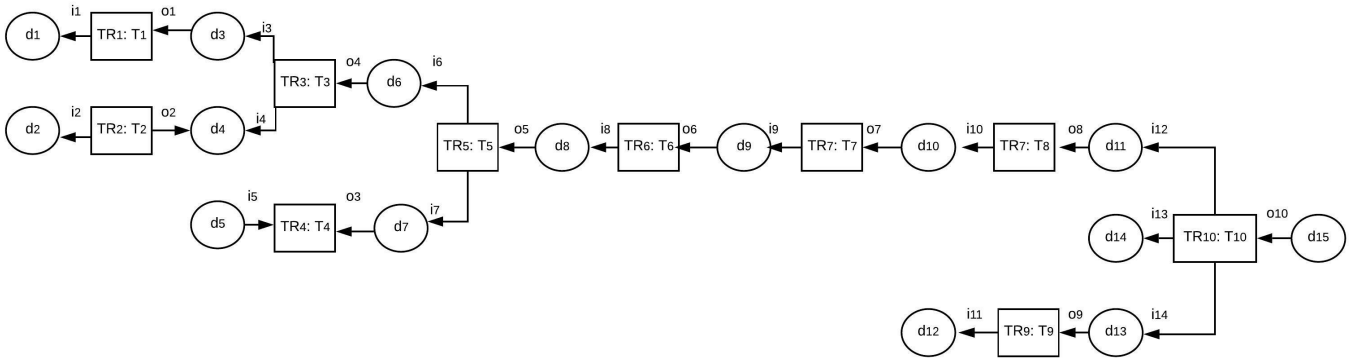

Fig. 2: Provenance of Autism Workflow.

Here we illustrate our concept using an example in health informatics, where secure communication in scientific workflow plays an important part for Autism Spectrum Disorder processing. In [33], an autism workflow system has been developed for the analysis, prediction, classification, and mining of a large corpus of autism data. From a security perspective, the access and analysis of such sensitive data should be handled based on a particular usage role. For this reason we develop a provenance security framework to allow permission for specific task and data products for specific roles. Ideally, in the Autism community, parents can have full access to all the diagnostic data, including medical, therapeutic, and school information. Meanwhile, for a school district, teachers by default may not have a privilege to see a child's medical details unless explicitly granted by their parents. Similarly, therapists can have access to certain sensitive parts of a workflow, but not the entire workflow. Therefore, secure communication of a workflow in the Autism community should be granted and a security framework is needed.

Fig. 1 is a sample workflow in the autism spectrum disorder domain. The example workflow depicts how unique attributes pertaining to a child's family, education and medical history can be harnessed to aid predictive analysis. In the figure, rectangles represent workflow tasks, little squares represent input/output ports, and directed edges represent data channels. A workflow task (task for short) is a functional building block of a workflow. Each task represents a computational or analytical step in the whole data analysis process. During execution, a workflow task takes a set of input data products from its input ports as input, and produces another set of data products to its output ports as output. Each input port is a placeholder for one of the input data products of a task before its execution, and each output port is a placeholder for one of the output data products of a task after its execution. A data channel links an output port o of an upstream task T_1 to an input port i of a downstream task T_2. During execution, the data product produced at output port o by task T_1 will be transferred to input port i for task T_2 to serve as one of its inputs. A data channel can also connect from a workflow input data product to an input port of a task, or from an output port of a task to an output data product placeholder, to model the inputs and outputs of the entire workflow.

In Fig. 2, we show the provenance graph corresponding to the workflow graph in Fig. 1, which captures the data lineage and morphology. Input data (e.g., $d5$) after being processed via tasks, i.e., $T4$, generates output data ($d7$ for

example in Fig. 2). After executing this workflow, in Fig. 2 we illustrate most detailed workflow run provenance information. In Fig. 2, circles, rectangles and edges represent data products, task runs, and data dependencies (i.e., *Used* and *GeneratedBy*), respectively. We further elaborate these figures in Section 4.

Below we define the basic PROV-DM provenance graph and access control policies.

Definition 2.1 (Provenance Graph)**.** *A provenance graph PG = (N, Ed) consists of:*

- *a set of Nodes* $N = Entity \bigcup Activity \bigcup Agent$*, where Entity is a set of entities, Activity is a set of activities and Agent is a set of agents, based on the PROV-DM model.*
- *a set of directed edges Ed =* $Ed_u \bigcup Ed_g \bigcup Ed_d \bigcup Ed_i \bigcup Ed_a \bigcup Ed_{ab} \bigcup Ed_{at}$
  where, i) $Ed_u \subseteq$ *Activity* $\times$ *Entity and (a,e)* $\in Ed_u$ *means that activity a* used *entity e.*
  ii) $Ed_g \subseteq$ *Entity* $\times$ *Activity and (e,a)* $\in Ed_g$ *means that entity e* was generated by *activity a.*
  iii) $Ed_d \subseteq$ *Entity* $\times$ *Entity and* $(e_1,e_2) \in Ed_d$ *means that entity* e_1 was derived from *entity* e_2.
  iv) $Ed_i \subseteq$ *Activity* $\times$ *Activity and* $(a_1,a_2) \in Ed_i$ *means that activity* a_1 was informed by *activity* a_2.
  v) $Ed_a \subseteq$ *Activity* $\times$ *Agent and (a,ag)* $\in Ed_a$ *means that activity a* was associated with *agent ag.*
  vi) $Ed_{ab} \subseteq$ *Agent* $\times$ *Agent and* $(ag_1,ag_2) \in Ed_{ab}$ *means that agent* ag_1 acted onbehalf of *agent* ag_2.
  vii) $Ed_{at} \subseteq$ *Entity* $\times$ *Agent and (e,ag)* $\in Ed_{at}$ *means that entity e* was attributed to *agent ag.*

Definition 2.2 (Role Based Access Control Policies)**.** *A Role-Based Access control policy* $\hat{R}$ *for provenance security is a tuple (* U*,* R*,* μ*,* A*,* W*,* E*,* ϕ *), where*

- U *is a set of users;*
- R *is a set of roles;*
- μ*:* $U \rightarrow R$ *is a function that maps users to their roles.*
- A *is a set of actions;*
- W *is a workflow;*
- E *is the set of elements including all the tasks, ports, and data channels in workflow* W *.*
- ϕ*:* $E \times R \times A \rightarrow \{0,1\}$ *is a function that maps permissions for elements, roles, and actions to* 0 *or* 1*. Here,* 0 *denotes restricted access and* 1 *denotes full access.*

The function ϕ is further defined as:

$$\phi(e,r,\alpha) = \begin{cases} \Gamma(e,r,\alpha), & \text{if } e \text{ is a task} \quad (1a) \\ \rho(e,r,\alpha), & \text{if } e \text{ is a port} \quad (1b) \\ \delta(p_1,p_2,r,\alpha), & \text{if Data Channel} \quad (1c) \end{cases}$$

For the function ϕ, the element could be either a task, a port, or a data channel. For tasks, we define function Γ; for ports, we define function ρ; and for data channels, we define function δ. Functions Γ, ρ and δ will be defined in detail in the following sections.

3 Provenance Security Policy Life Span

The provenance security policy life span comprises four iterative stages: i) Security policy specification, ii) Security policy enforcement, iii) Security policy analysis, and iv) Security policy evolution. A administrator of access control policies coordinates with the system users, and determines the policies to be enforced in either one or all levels at task, port and data channel level. In the security policy enforcement stage, based on system users access on protected elements, the policies are applied to either grant or restrict access. In correspondence to context or environment of the application, the policies evolve to adopt correlated changes. In the policy analysis stage, policy quality requirements are analyzed. This phase analyzes the policy qualities like consistency, completeness, conciseness to make sure the proposed policies adhere to all those qualities. Finally, in the policy evolution stage, we evaluate quality requirements and identify any quality discrepancies and modify those policies based on the identified discrepancies in policies. Fig. 3 shows a graphical representation of the provenance security policy life span.

4 The ProvSec Prototype and A Case Study

We have developed the ProvSec prototype to validate the effectiveness of our protocol, with workflow views and mapped provenance views, in DATAVIEW [4], [34]. We specify our security policies on a workflow graph and map the security policies to their counterparts on provenance graphs, based on the role of the user. The security view of provenance does not have to be a connected graph. The reason is that security is imposed based on corresponding roles. Therefore, the dependencies between the subgraphs are hidden. In the DATAVIEW system, the *Provenance Manager* is responsible for managing scientific workflow provenance.

We illustrate our workflow provenance security mechanism with a real-life example, by collecting data from the SFARI project [35] about the Autism Spectrum Disorder (ASD). The autism workflow [36], [33] created in the DATAVIEW system is used here. This running workflow contains ten tasks. The workflow in Fig. 1 explores all of the unique attributes of children's family history, education history, and medical history and identifies predictive features pertaining to each individual child. This workflow implements data mining techniques for predicting the outcome based on these features. Both tasks T_1 and T_2 perform the *Projection* operation, which projects the predominant attributes out of a pool of attributes. Based on the SFARI id, task T_3 then performs another *Natural Join* operation. Task T_4 performs *Projection* on the SFARI's follow-up family history dataset. On the retrieved result of tasks T_3 and T_4, T_5, the *Natural Join* operation is performed. Task T_6 checks whether there are any missing or null values in a retrieved data set. Then Task T_7 performs another *Projection* operation. The output of this task works as an input of task T_8, which converts CSV files to the ARFF file format. The final result is retrieved by executing data mining task T_{10}. For data mining and predictive analytics, a test dataset is required, and that test dataset is provided to task T_9 for converting it to the ARFF format. The training set and test set are used to tune the best hyper-parameters for the random forest algorithm. The best set of parameter values are finally used to obtain the final prediction result. After executing this workflow in Fig. 2, we illustrate most detailed workflow

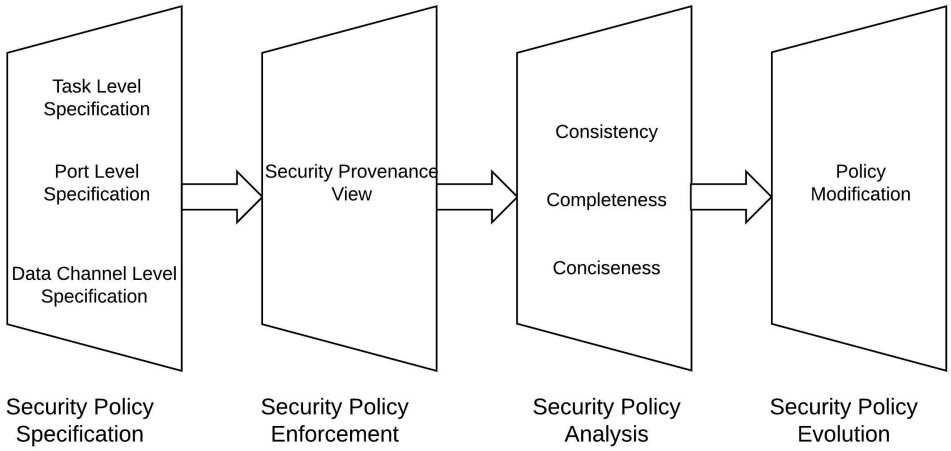

Fig. 3: Provenance Security Policy Life Span.

run provenance information. In Fig. 2, circles represent data products, and rectangles represent workflow task runs. The edges between data products and tasks represent relations. For example, an edge from a data product to a task is called a *wasGeneratedBy* relation, and an edge from a task to a data product is called a *used* relation.

5 Security Policy Specification

In this section, we describe three levels of security policy specifications.

5.1 Task Level Specification

Definition 5.1 (Task Annotation)**.** *A task level specification is a function denoted by* Γ*:* $T \times R \times A \rightarrow \{0, 1\}$*, which maps specific users and tasks to the permission level and is defined by:*

$$\Gamma(t, r, \alpha) = \begin{cases} Invalid, & \text{if } \Pi(t, r, \alpha) = 1 \text{ and } \quad (2a) \\ & \Gamma(t_p, r, \alpha) = 0 \quad (2b) \\ \Pi(t, r, \alpha), & \text{if } \Pi(t, r, \alpha) \neq -1 \quad (2c) \\ \Gamma(t_p, r, \alpha), & t_p \text{ is not null and } \quad (2d) \\ & \Pi(t, r, \alpha) = -1 \quad (2e) \\ Invalid, & t_p \text{ is null and } \quad (2f) \\ & \Pi(t, r, \alpha) \neq -1 \quad (2g) \end{cases}$$

In task specification, the access permission can be annotated by 0 or 1. Here we define a function Π: $E \times R \times A \rightarrow \{0, 1, -1\}$, which returns permission of role, element, and action triplet. If it returns -1, it means there is no explicit specification for (t, r, α); otherwise, it returns the explicit annotation for triple (t, r, α).

If the permission is not explicitly specified in RBAC, then child task t can inherit the permission from its parent task t_p, $\alpha \in A$, $r \in R$. In other words, the task level security specification, if explicitly stated, is validated against the consistency requirement of the protocol. In this case, if its parent task does not have security access, the child task inherits the restriction, and this restriction cannot be overridden by explicit specification. One important feature of the task is that when it is annotated as 1 then all other tasks, ports or data channels contained in task T should be accessible; otherwise a 0 annotation is explicitly specified or derived from them.

Our definition captures the inconsistency specification between a task and any of its ancestors, while [13] only captures the inconsistency specification between a task and its containing task, the task that immediately contains task t.

Here, we identify four cases that are exclusive in the given order:

- Case a: If the parent task differs from the child task in question in terms of access control permission such that the parent task does not have access yet, the child task has the explicit specification to have secure access, and will result in inconsistency in access control protocol.
- Case b: If the task in question has an access control protocol explicitly specified, then this will override the ancestral access control protocols.
- Case c: If the current task does not have an explicit specification but has a valid parent, then it will inherit its parent's access control privileges.
- Case d: Lastly, if the current task does not have a valid parent and valid specification, an exception will be thrown.

The permission specification can be calculated using the *FindTaskSpec* function in Algorithm 1.

5.2 Port Level Specification

Definition 5.2 (Port Annotation)**.** *A port level specification is a function denoted by* ρ*:* $P \times R \times A \rightarrow \{0, 1\}$*, that maps a specific role and port to the permission level and is defined by:*

$$\rho(p, r, \alpha) = \begin{cases} Invalid, & \text{if } \Pi(p, r, \alpha) = 1 \text{ and } \quad (3a) \\ & \Gamma(t_p, r, \alpha) = 0 \quad (3b) \\ \Pi(p, r, \alpha), & \text{if } \Pi(p, r, \alpha) \neq -1 \quad (3c) \\ \Gamma(t_p, r, \alpha), & \text{otherwise} \quad (3d) \end{cases}$$

Algorithm 1: Algorithm for calculating security specification in Task

Input: Task t, Role r, Action α.
Output: Task security annotation $< t, a \in \{0,1\} >$.

```
If ∃ a = Π(t, r, α)
        a_p = FindTaskSpec(t_p, r, α), where t_p ∈ Parent (t)
        if (a & ! a_p)
                return Invalid
        return < t, a >
else
        return < t, FindTaskSpec(t_p, r, α) >
```

Fig. 4: Task Level Security Specification.

Algorithm 2: Algorithm for calculating security specification on Port

Input: Port p, Role r, Action α.
Output: Port security annotation $< p, a \in \{0,1\} >$.

```
If ∃ a = Π(p, r, α)
        a_p = FindTaskSpec(t_p, r, α), where t_p is the task that contains p
        if (a & ! a_p)
                return Invalid
        return < p, a >
else
        return < p, FindPortSpec(t_p, r, α) >
```

Fig. 5: Port Level Security Specification.

Ports can be specified with 0 or 1. In a port level specification, when a port has no specified security specification, it will inherit its permission from its containing task. The administrator can explicitly specify all or some port access permissions. For all workflow runs, the port annotation 1 or 0 specified for any given task port restricts the accessibility of the corresponding data product.

Here we identify three cases that are exclusive in the given order:

- Case a: If the parent task does not have access permission, but the port contained in that task has explicit specification to have secure access, then this will result in invalid access control protocol.
- Case b: If the port in question has an access control protocol explicitly specified, then this will override ancestral access control protocols.
- Case c: If the port does not have an explicit specification but its containing task has an access control specified, it will inherit the task's access control privileges.

Here, t_p denote the containing task of port p.

In appearance, our port-level security specification looks like the same as [13]. However, it improves the inconsistency specification check due to the improvement of the task-level security specification, which affects the result of the port-level specification inconsistency check.

Note that our proposed port-level specification mechanism is significantly simplified from our previous definition, as we do not allow the accessibility of a data channel when its respective ports are not accessible.

The annotation of a port is calculated by the *FindPortSpec* function in Algorithm 2.

5.3 Data Channel Level Specification

Definition 5.3 (Data Channel Annotation)**.** *A data channel level specification is a function denoted by* δ*:* $P \times R \times A \rightarrow \{0,1\}$*, that maps specific role and port to a permission and is defined by:*

$$\delta(p_1, p_2, r, \alpha) = \begin{cases} \rho(p_1, r, \alpha), & \text{if } \rho(p_1, r, \alpha) = \rho(p_2, r, \alpha) \quad (4a) \\ Invalid & \text{Otherwise} \quad (4b) \end{cases}$$

Data Channel specification is quite straight-forward. When both ports have access permission, then data channel must have access permission. When both ports' permissions are denied, the data channel's permission is denied as well.

Our definition significantly simplifies the specification effort, at a small cost of not allowing the specification of data dependency without the accessibility of respective ports, which has very rare use cases in practice.

The permission specification can be calculated in Algorithm 3.

Algorithm 3: Algorithm for calculating security specification on Data Channel

Input: p_1, p_2, Role r, Action α
Output: Port security annotation $< (p_1, p_2), a \in \{0,1\} >$.

```
If (FindPortSpec(p_1, r, α) = FindPortSpec(p_2, r, α))
        return < (p_1, p_2), FindPortSpec(p_1, r, α) >
else
        return Invalid
```

Fig. 6: Data Channel Level Security Specification.

6 Security Policy Enforcement

In security policy enforcement, provenance systems maintain a different view of information for different roles and enforce associated privileges.

We define a security provenance view as a restricted view of provenance only consisting of the information that users are authorized to access. Security Provenance view is inherited from the security protocol imposed on the underlying workflow. In order to guarantee that there are no data vulnerabilities, we formalize the inheritance in the following way as shown in definitions 5.1 and 5.2. Task level access control policies for the provenance are inherited from the workflow tasks, and port level policies are inherited from the corresponding ports in the workflow.

To illustrate this view in the PROV-DM model [37], we graphically represent the provenance model relation "Used" in Fig. 7 and "wasGeneratedBy" in Fig. 8 and corresponding mapping from workflow to provenance.

However, in order to impose relation security, we analyze Table 1 as an example. Table. 1 shows the specification mapping from workflow to provenance.

We observe that the edge security policy is derived from the associated task and does not depend on the port policy. This is reflected in definitions 5.1 and 5.2.

Let E be the elements in a workflow consisting of tasks, ports and data channels and let Ψ be a mapping function $\Psi : E \rightarrow N$ that maps elements in the workflow to their corresponding nodes in the provenance graph. The inverse function $\Psi^{-1} : N \rightarrow E$ returns the reverse mapping.

We also introduce the following two notations, Let $\Im : E \rightarrow E$ be a function defined as follows:

$$\Im(e) = \begin{cases} e, & \text{if } e \text{ is task} \quad (5a) \\ t_p, & \text{if } e \text{ is port, } t_p \text{ is container task.} \quad (5b) \end{cases}$$

Let $\wp : E \rightarrow E$ be a function defined as follows:

$$\wp(e) = \begin{cases} e, & \text{if } e \text{ is port} \quad (6a) \\ \{p_e\}, & \text{if } e \text{ is task, } \{p_e\} \text{ are ports of } e. \quad (6b) \end{cases}$$

Definition 6.1 (Security Provenance View of the "Used" Relation)**.**

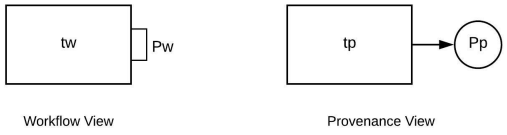

Fig. 7: Provenance Security from Workflow Security in the "Used" Relation.

- $\Gamma(\Psi(t_w)$,r,view) = $\Gamma(t_w$,r,view)
- $\Delta(\Psi(P_w)$,r,view) = $\rho(P_w$,r,view)
- ζ(edge ($\Psi(t_w)$, $\Psi(P_w)$), r, view) = $\Gamma(t_w$,r,view)

Definition 6.2 (Security Provenance View of the "wasGeneratedBy" Relation)**.**

Fig. 8: Provenance Security from Workflow Security in the "wasGeneratedBy" Relation.

- $\Gamma(\Psi(t_w)$,r,view) = $\Gamma(t_w$,r,view)
- $\Delta(\Psi(P_w)$,r,view) = $\rho(P_w$,r,view)
- ζ(edge ($\Psi(P_w)$, $\Psi(t_w)$), r, view) = $\Gamma(t_w$,r,view)

We illustrate security policy requirements based on the Autism provenance system in Section 2 and define those access control policies in Table 2.

7 Security Policy Quality Requirements and Analysis

We define and illustrate our security policy quality requirements in this section.

7.1 Consistency

acp_i and acp_j are consistent if and only if:

acp_i.u = acp_j.u,
$\wedge$ $\mu(acp_i$.u) = $\mu(acp_j$.u)
$\wedge$ acp_i.e = acp_j.e
$\wedge$ acp_i.a = acp_j.a
$\Longrightarrow$ $\phi(\mu(acp_i$.u),e,a) = $\phi(\mu(acp_j$.u),e,a),
$\forall$ u $\in$ U, $\forall$ e $\in$ E, $\forall$ a $\in$ A

Here we define the consistency between two policies acp_i and acp_j, by constraining that if these two policies have the same role, the same element, and the same activity, then both policies should have the same access permission. If there is any inconsistency in policies, that requires conflict resolution which can be minimized with consistent policies.

Example 1**:** As shown in Table. 2, in teacher role, acp_{14} and acp_{15} are not consistent. Both policies need to have the same access permission when they have the same role, user, element, and activity. Here acp_{14} and acp_{15} do not meet that criterion. They are inconsistent because one port is specified with negative access, while the other one is specified with positive access. In Table. 2, for a single data channel, the output port O_6 is specified negative and the input port i_9 is specified positive. From our Port level specification algorithm, both ports should have same permission. In this case, the output and the input port of a single data channel have different permissions. Therefore, this is an inconsistency in the policy. We can correct this inconsistency in the policy evolution phase.

7.2 Completeness

Any access control policy acp_i is complete if and only if:

$\forall$ i, $\mu(acp_i$.u) is defined $\wedge$ $\phi(\mu(acp_i$.u),e,$\alpha)$ is defined;
where $\exists$ u $\in$ U, $\exists$ e $\in$ E, $\exists$ $\alpha \in$ A

Completeness of an access control policy is where for any role, an access control policy is defined. A complete access control policy has both role defined and access policy defined. An incomplete policy has either role undefined or access policy for task/port undefined.

Example 2**:** In Table. 2, there is no access control policy for the teacher role for allowing or denying access to the Family History table dataset of Task T_4. Without setting up the access control policy for input i_5 or task T_4 the policy defining accessing or denying the information of the Family history is incomplete.

7.3 Conciseness

An access control policy $acp_i \in \hat{R}$ is concise if and only if:

$\exists$ $acp_j \in \hat{R}$
$\wedge$ $\mu(acp_i$.u) = $\mu(acp_j$.u)
$\wedge$ acp_i.e = acp_j.e
$\wedge$ acp_i.a = acp_j.a,
$\wedge$ $\phi(\mu(acp_i$.u),e,a) = $\phi(\mu(acp_j$.u),e,a)

TABLE 1: RBAC Security Specification for the "Used" and "wasGeneratedBy" Relations.

Workflow RBAC			*Provenance RBAC*	
Task	***Port***	***Task***	***Port***	***Relation***
+	-	+	-	+
+	+	+	+	+
-	-	-	-	-
-	+		INVALID	

TABLE 2: Role Based Access Control Policy for the Provenance System.

Access Control Policy	*Role*	*Permission*		
		Element	***Action***	***Sign***
acp_1	Parents	T_1	Read	+
acp_2		i_1	Read	+
acp_3		T_2	Read	+
acp_4		i_2	Read	+
acp_5		T_4	Read	+
acp_6		i_5	Read	+
acp_7		T_9	Read	+
acp_8		O_{10}	Read	+
acp_9	Teachers	i_1	Read	+
acp_{10}		T_2	Read	+
acp_{11}		i_2	Read	−
acp_{12}		T_4	Read	+
acp_{13}		T_5	Read	+
acp_{14}		O_6	Read	−
acp_{15}		i_9	Read	+
acp_{16}		O_{10}	Read	+
acp_{17}	Therapist	T_1	Read	+
acp_{18}		i_1	Read	+
acp_{19}		T_2	Read	+
acp_{20}		i_2	Read	+
acp_{21}		T_4	Read	+
acp_{22}		T_5	Read	+
acp_{23}		T_9	Read	+
acp_{24}		T_{10}	Read	+
acp_{25}		O_{10}	Read	+

$\implies$ i = j ;
$\forall$ u $\in$ U, $\forall$ e $\in$ E, $\forall$ a $\in$ A.

The conciseness of an access control policy means that there exists no other policy that shares the same role, the same element, the same action, and the same permission. If there are two access control policies acp_i and acp_j, where both policies have the same role, same user, same element and same activity, but defined as two different policies, then we consider these two policies are not concise.

Example 3**:** Based on access control policies in Table. 2, acp_{23} and acp_{24} are not concise. From task specification, we know that when the parent task's accessibility is positive, then a child task's accessibility is positive too unless otherwise stated. We do not have to specify both cases here.

Theorem 1. *If RBAC in WF_{RBAC} is consistent, then RBAC in Provenance $Prov_{RBAC}$ is consistent as well.*

Proof. Let us assume that WF_{RBAC} is consistent and $Prov_{RBAC}$ is not consistent.

From the definition we know WF_{RBAC} is consistent if and only if:
$i \neq j \wedge acp_i$.r = acp_j.r
$\wedge$ acp_i.e = acp_j.e $\wedge$ acp_i.a = acp_j.a
This implies
ϕ(acp_i.r, acp_i.e, acp_i.a) = ϕ(acp_j.r, acp_j.e, acp_j.a).

If $Prov_{RBAC}$ is inconsistent, then either or all of the following is true:
$\Gamma(\Psi(\Im(acp_i.e)), acp_i.r, acp_i.a) \neq \Gamma(\Psi(\Im(acp_j.e)), acp_j.r, acp_j.a)$ or
$\rho(\Psi(\wp(acp_i.e)), acp_i.r, acp_i.a) \neq \rho(\Psi(\wp(acp_j.e)), acp_j.r, acp_j.a)$ or
$\zeta(edge(\Psi(\Im(acp_i.e)), \Psi(\wp(acp_i.e))), acp_i.r, acp_i.a) \neq \zeta(edge(\Psi(\Im(acp_j.e)), \Psi(\wp(acp_j.e)), acp_j.r, acp_j.a)$

However,
$\Gamma(\Psi(\Im(acp_i.e)), acp_i.r, acp_i.a) = \Gamma(\Im(acp_i.e), acp_i.r, acp_i.a)$ and
$\Gamma(\Psi(\Im(acp_j.e)), acp_j.r, acp_j.a) = \Gamma(\Im(acp_j.e), acp_j.r, acp_j.a)$.

Again since,
ϕ(acp_i.r, $\Im(acp_i$.e), acp_i.a) = ϕ(acp_j.r, $\Im(acp_j$.e), acp_j.a),

We can conclude that,
$\Gamma(\Im(acp_i.e), acp_i.r, acp_i.a) = \Gamma(\Im(acp_j.e), acp_j.r, acp_j.a)$.

Hence,
$\Gamma(\Psi(\Im(acp_i.e)), acp_i.r, acp_i.a) = \Gamma(\Psi(\Im(acp_j.e)), acp_j.r, acp_j.a)$.

Similarly, we can show that,

$\rho(\Psi(\wp(acp_i.e)), acp_i.r, acp_i.a) =$
$\rho(\Psi(\wp(acp_j.e)), acp_j.r, acp_j.a).$

Lastly, since,
$\zeta(edge(\Psi(\Im(acp_i.e)), \Psi(\wp(acp_i.e))), acp_i.r, acp_i.a) =$
$\Gamma(\Im(acp_i.e), acp_i.r, acp_i.a)$

and

$\zeta(edge(\Psi(\Im(acp_j.e), \Psi(\wp(acp_j.e)), acp_j.r, acp_j.a) =$
$\Gamma(\Im(acp_j.e), acp_j.r, acp_j.a)$

and
$\Gamma(\Im(acp_i.e), acp_i.r, acp_i.a) = \Gamma(\Im(acp_j.e), acp_j.r, acp_j.a),$

We can then conclude that
$\zeta(edge(\Psi(\Im(acp_i.e)), \Psi(\wp(acp_i.e)), acp_i.r, acp_i.a) =$
$\zeta(edge(\Psi(\Im(acp_j.e)), \Psi(\wp(acp_j.e))), acp_j.r, acp_j.a).$

So, $Prov_{RBAC}$ cannot be inconsistent.

□

Theorem 2. *If RBAC in WF_{RBAC} is complete, then RBAC in Provenance $Prov_{RBAC}$ is complete as well.*

Proof. An access control policy acp_i is complete if and only if:

μ(acp_i.u) is defined $\wedge$ ϕ(μ(acp_i.u), $acp_i.e$,α) is defined
$\forall$ u $\in$ U, $\forall$ e $\in$ E, $\forall$ $\alpha \in$ A.

Again, since we are assuming that RBAC in $Prov_{RBAC}$ is incomplete:
$\Gamma(\Psi(\Im(acp_i.e))$,r,view) is undefined)
$\vee$ $\Delta(\Psi(\wp(acp_i.e))$,r,view) is undefined
$\vee$ ζ(edge ($\Psi(\Im(acp_i.e))$, $\Psi(\wp(acp_i.e))$), r, view) is undefined.

However, since
$\Gamma(\Psi(\Im(acp_i.e))$,r,view) = $\Gamma(\Im(acp_i.e)$,r,view),
$\Delta(\Psi(\wp(acp_i.e))$,r,view) = $\rho(\wp(acp_i.e)$,r,view),
ζ(edge ($\Psi(\Im(acp_i.e))$, $\Psi(\wp(acp_i.e))$), r, view) = $\Gamma(acp_i.e$,r,view),
and $\Gamma(\Im(acp_i.e)$,r,view), $\rho(\wp(acp_i.e)$,r,view) and $\Gamma(acp_i.e$,r,view) are defined.

Hence $Prov_{RBAC}$ cannot be incomplete. □

Theorem 3. *If RBAC in WF_{RBAC} is concise, then RBAC in Provenance $Prov_{RBAC}$ is concise as well.*

Proof. Since, RBAC in WF_{RBAC} is concise, we get:
if $\exists$ acp_i, $acp_j \in \hat{R}$ such that:
μ(acp_i.u) = μ(acp_j.u),
$\wedge$ acp_i.e = acp_j.e,
$\wedge$ acp_i.a = acp_j.a,
$\wedge$ ϕ(acp_i.r, acp_i.e, acp_i.a) = ϕ(acp_j.r, acp_j.e, acp_j.a)
$\wedge$ i = j; where $\forall$ u $\in$ U, $\forall$ e $\in$ E, $\forall$ a $\in$ A.

Since we are assuming that RBAC in $Prov_{RBAC}$ is redundant, it implies:
$\Gamma(\Psi(\Im(acp_i.e))$,r,view) = $\Gamma(\Psi(\Im(acp_j.e))$,r,view) and
$\Delta(\Psi(\wp(acp_i.e))$,r,view) = $\Delta(\Psi(\wp(acp_j.e))$,r,view) and
ζ(edge ($\Psi(\Im(acp_i.e))$, $\Psi(\wp(acp_i.e))$), r, view) = ζ(edge ($\Psi(\Im(acp_j.e))$, $\Psi(\wp(acp_j.e))$), r, view) and $i \neq j$

However, from the definition we know that:
$\Gamma(\Psi(\Im(acp_i.e))$,r,view) = $\Gamma(\Im(acp_i.e)$,r,view)
$\Delta(\Psi(\wp(acp_i.e)$,r,view) = $\rho(\wp(acp_i.e)$,r,view)

In addition
$\Gamma(\Psi(\Im(acp_j.e))$,r,view) = $\Gamma(\Im(acp_j.e)$,r,view)
$\Delta(\Psi(\wp(acp_j.e)$,r,view) = $\rho(\wp(acp_j.e)$,r,view)

Furthermore, since
$\Gamma(\Im(acp_i.e)$,r,view) = $\Gamma(\Im(acp_j.e)$,r,view) and
$\rho(\wp(acp_i.e)$,r,view) = $\rho(\wp(acp_j.e)$,r,view),
it implies that i = j.

Hence, RBAC in $Prov_{RBAC}$ should be concise as well.

□

8 Security Policy Evolution

The security policy evolution phase is for modification of policies based on the quality analysis phase after finding all the inconsistent, incomplete and redundant policies. The administrator holds the right to do the modification after finding those incorrect policies. For instance, inconsistent policies in Table. 2, for the role of teachers, policies acp_{14} and acp_{15} are inconsistent because the ports of a data channel are specified with two different permissions. For a single data channel, the output port O_6 is specified negative and the input port i_9 is specified positive. From our Port level and data channel level specification algorithms, both ports should have the same permission. As in this case, the output and the input port of a single data channel have different permissions, in the evolution phase, the administrator will do the modification and specify explicitly both ports O_6 and i_9 are negative. For incomplete policies like the one in the example, where no access control policy for the teacher role is specified for allowing or denying access to the family History table dataset of Task T_4, a policy evolution is needed. Without setting up the access control policy for input i_5 or task T_4, the policy defined for accessing or denying the information of the family history is incomplete. For that, the administrator modifies the policies by adding access right for Task T_4 or input i_5. For redundant policies like acp_{23} and acp_{24}, the administrator can remove the policy acp_{24} because when the parent task's accessibility is positive, then the child task's accessibility is positive as well unless otherwise stated.

9 The ProvSec Prototype and the Case Study (con't)

We use the ProvSec prototype for an autism workflow with both the defined and then evolved policies. Based on each role we can see a security view of provenance by imposing the defined policies.

As evidenced in our policy specification, our approach improves the state of the art [13], by introducing the notion of recurrent upstream inconsistency specification as opposed to an inconsistency specification as a function of the

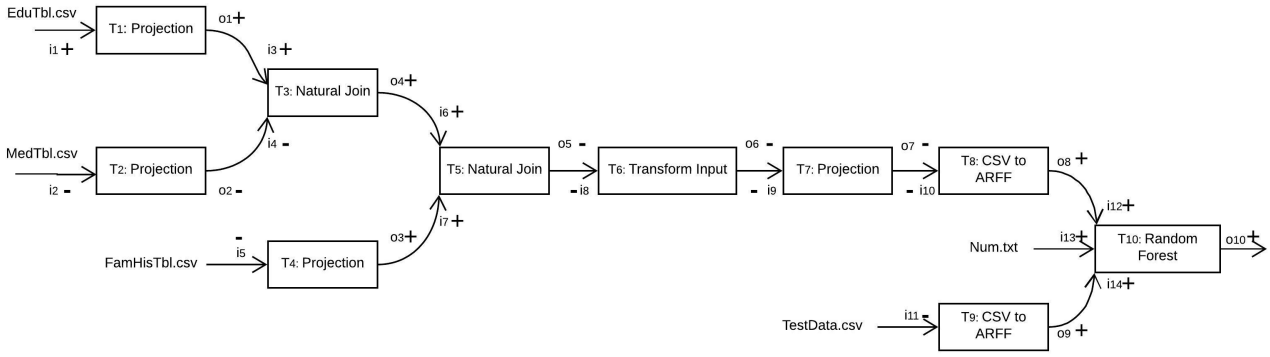

Fig. 9: Workflow Permission for Teachers' Access in the Autism Provenance System.

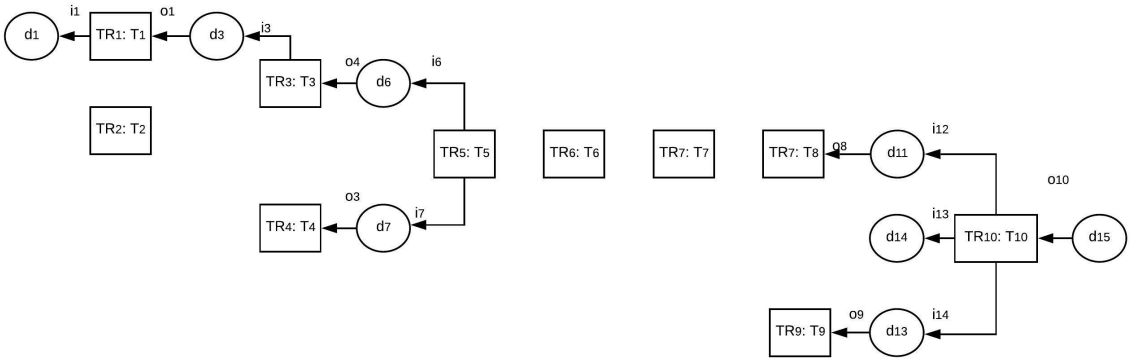

Fig. 10: Security View of Teachers's Access in the Autism Provenance System.

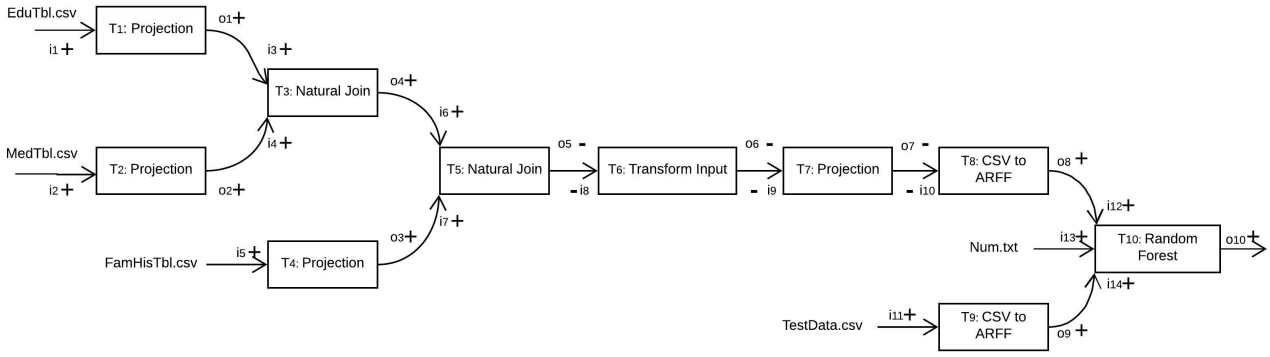

Fig. 11: Workflow Permission for Therapists's Access in the Autism Provenance System.

immediate parent node for a task. Furthermore, our port-level security specification improves the implementation of inconsistency specification detection via the enhancement of task-level security specification, resulting in a more consistent policy holistically.

Because of the sensitive nature of the Autism workflow, we propose the restriction on data product and their provenance information for different roles. In ProvSec, we define three roles for the Autism workflow:

- Parent's access permission specification and the corresponding provenance security view.
- Teacher's access permission specification and the corresponding provenance security view.
- Therapist's access permission specification and the corresponding provenance security view.

The parents have access permission to all the tasks, ports

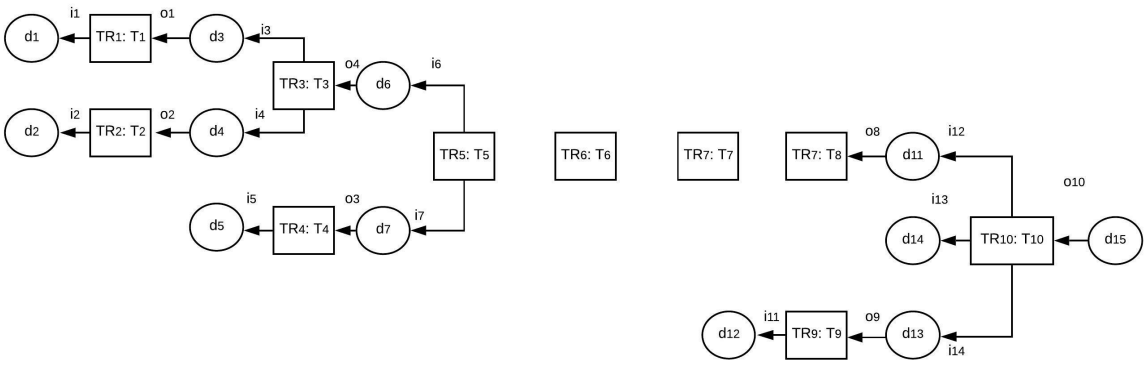

Fig. 12: Security View of Therapists's Access in the Autism Provenance System.

and data channels. For the parent role, in the provenance security view, parents can see all the sensitive data products and their corresponding relations. In addition to the input and output data products, they can have access to all the intermediary data products and can provide the test set of data for projecting output.

For the teacher role, teachers or educators can have access to everything except the medical input data product i_2, the projected output O_2 of the data product, and the family history input data i_5. When any data channel in the workflow is specified as negative, the data product generated in provenance are not allowed to be seen by users. Any negative annotation on ports implies merely that the generated data product should not be visible to users of that particular role. Fig. 9 shows the workflow permission for teachers, and Fig. 10 shows the corresponding security provenance view for teachers.

For the therapist role, all therapists or clinicians can have access to the initial raw data to know about ASD children and prototyping appropriate program. This role does not require to access intermediate data products or relations. However, they have permission to view predicted output for the provided input parameters.

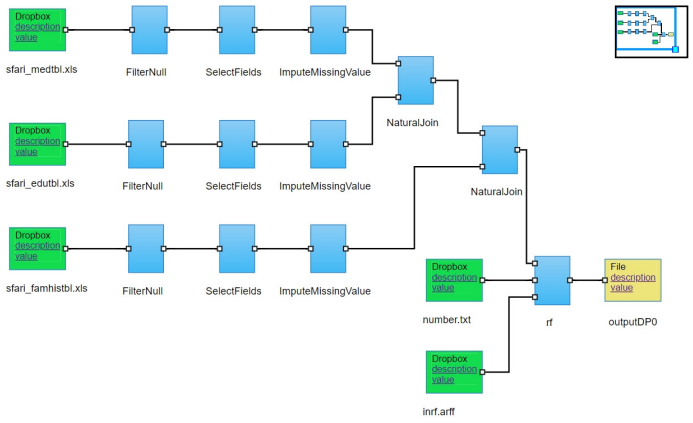

Fig. 13: The Autism Workflow in DATAVIEW.

Fig. 11 shows the workflow security specification for therapists, and Fig. 12 shows the corresponding security provenance view for them, after implementing all the security policies.

We have conducted a collection of experiments on a machine with Intel core $i7-3612QM$ CPU $@2.10GHz$ x 8 processor and 7.7 GB memory. We have used the DATAVIEW workflow management system in Fig. 13 for implementing data mining techniques for predicting the outcome based on the available features. The main reason of using DATAVIEW is to give flexibility to the researchers of the Autism Community and also parents and caregivers, so that they do not have to deal with any underlying complexity of the computation infrastructure.

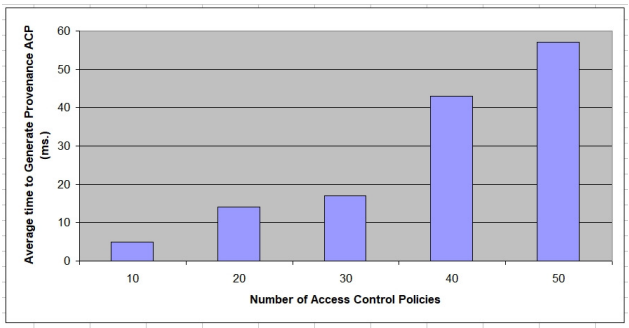

Fig. 14: The Average Time to Generate Provenance Access Control Policies.

In Figure 14, we plot time to inherit workflow specific access control protocol to the provenance system. We can observe that the inheritance process is not time intensive and can be computed very fast. We can also observe a linear relationship between the number of access control protocols in the scientific workflow system, and the time it takes to execute the translation process. For example, for a scientific workflow with 10, 20, 30, 40, 50, etc. access control protocols specified, it takes 5, 19, 17, 43, 57 milliseconds, respectively.

10 Related Work

For business workflows, the importance and requirements of security are well understood [38], [39], [40], [41], [42], [43].

From the perspective of a workflow system, the requirements for security can be managed by either the workflow system itself [44] or by a system outside of the workflow engine [45]. Most of the security work has been done in authentication [46], authorization [47], [48], [49], [50], [51], data privacy and secure workflow models [52], [53]. The security issues of provenance have recently been identified by some researchers [54].

The authors of [28] formalized a model for provenance with security properties like disclosure and obfuscation on workflow provenance graphs, database queries, and automata. They explained the most general form of provenance for the system through traces. Their framework defines primarily static provenance situation, but not dynamic provenance situation.

In [55], [56], the authors address a number of research questions on provenance security, and develop mechanisms for securing provenance, by using appropriate encryption and digital signature. They allow auditors to check the integrity of provenance without necessary access to underlying data and vice versa [28]. [56] maintains the integrity of provenance records in a stateful system and prevents forgery.

Based on the work of Cheney et. al. [57], Chong [58] formulated a syntactic model of traces and proposed semantic definitions of provenance security policies. Chong [58] formalized two properties, "provenance security" and "data security." In provenance security, provenance of a workflow run cannot be inferred from data; likewise, highly sensitive input data of a workflow cannot be inferred from its provenance.

In [59], Davidson et al. propose a formal definition of privacy and confidentiality policies for workflow provenance, and formalize the notion of privacy and focus on a mathematical model for solving privacy-preserving view as a result of query by an auditor. However, their approach remains theoretical and does not provide a framework for provenance models that address security.

In [60], the authors investigate the problem of securing data provenance in the cloud and propose a schema that supports encrypted search while protecting confidentiality of data provenance stored in the cloud. Their main contribution is that neither an adversary nor a cloud service provider can learn about the data provenance or the query [60].

The Secure Provenance (SPROV) scheme in [56], [61] provides security assurances of confidentiality and integrity of the data provenance and automatically collects data provenance at the application layer. They ensure confidentiality by employing state-of-the-art encryption techniques where integrity is preserved by using the digital signature of the user who takes actions. However, the SPROV scheme has some limitations. It does not provide confidentiality to the source data whose data provenance is being recorded and it does not provide any mechanism to querying data provenance [60].

The PSecOn scheme in [62] proposes a cyber laboratory to collaborate and share scientific resources for provenance Security from Origin. Integrity of the scientific results and corresponding data provenance can be ensured through the PSecOn scheme in an e-science grid. This scheme encrypts the source data. The limitation of PSecOn is its strong assumption of relying on a trusted infrastructure, restricting the possibility of managing data provenance in the cloud [60].

Lu et al. [63] introduce a scheme to manage data provenance in the cloud, and provide user access to the online data where data is shared among multiple users. Confidentiality and integrity are guaranteed through user encryption and signs over the data, where a cloud service provider receives and verifies the signature before storing that data. The main drawback of this approach is that it only traces the user while it does not provide any details about how the data provenance is managed by the cloud service provider [60].

Aldeco et al. [64] provide concrete cryptographic constructs to ensure the integrity of data provenance. They describe four stages: recording provenance, storing provenance, querying provenance and analyzing provenance graph for answering questions regarding the execution of the entities of the system. When data provenance is recorded and stored, integrity is ensured. Their limitation is a lack of detail about how to provide confidentiality to data provenance.

In [55], data provenance is considered as a causality graph with annotations. They focus on the security models of data provenance at an abstract level. They mentioned security of data provenance is different from the source data it describes, thus it requires different access controls. But they do not address how to define and enforce these access controls.

Security issues related to a Service Oriented Architecture (SOA) based provenance system are discussed in [54]. They suggest to restrain auditors by limiting the access to the results of a query using cryptographic techniques, however they did not provide a concrete solution.

11 Conclusions and Future Work

In this work, we studied access control policies for data products and derivation history for protecting sensitive data and processes. First, we formalized secure scientific workflow specifications for tasks, ports and data channels with proposed algorithms of access control policies. Second, we analyzed those policies from the perspective of policy quality requirements. Third, we formalized the security view for provenance based on mapping between workflow and provenance. Forth, we provided proofs of holding policy quality requirements for provenance. Lastly, we evaluated with an example in the autism community to show the validity of our quality assurance of access control policies for provenance.

In the future, we plan to consider conducting security case studies with more complex data patterns and integrate our access control policies to deal with different granularities of data. We also plan to improve the usability of the system with large data sets.

Acknowledgment

This work is supported by National Science Foundation, under grant NSF CNS-1747095 and OAC-1738929. In addition, this material is based upon work supported in part by the National Science Foundation under Grant OAC-1443069.

REFERENCES

[1] P. Buneman and W. C. Tan, "Provenance in Databases," in *Proc. of the ACM SIGMOD International Conference on Management of Data*, 2007, pp. 1171–1173.

[2] S. B. Davidson and J. Freire, "Provenance and Scientific Workflows: Challenges and Opportunities," in *Proc. of the ACM SIGMOD international conference on Management of data*, 2008, pp. 1345–1350.

[3] L. Moreau, "The Foundations for Provenance on the Web," *Foundations and Trends in Web Science*, vol. 2, no. 2-3, pp. 99–241, 2010.

[4] A. Kashlev and S. Lu, "A System Architecture for Running Big Data Workflows in the Cloud, SCC," in *Proc. of IEEE International Conference on Services Computing*, 2014, pp. 51–58.

[5] J. Zhang, P. Votava, T. J. Lee, O. Chu, C. Li, D. Liu, K. Liu, N. Xin, and R. R. Nemani, "Bridging VisTrails Scientific Workflow Management System to High Performance Computing," in *Proc. of the IEEE Ninth World Congress on Services, SERVICES*, 2013, pp. 29–36.

[6] J. Sroka, J. Hidders, P. Missier, and C. A. Goble, "A Formal Semantics for the Taverna 2 Workflow Model," *Journal of Computer and System Sciences*, vol. 76, no. 6, pp. 490–508, 2010.

[7] E. Deelman, K. Vahi, M. Rynge, G. Juve, R. Mayani, and R. F. da Silva, "Pegasus in the Cloud: Science Automation through Workflow Technologies," *IEEE Internet Computing*, vol. 20, no. 1, pp. 70–76, 2016.

[8] D. Crawl, A. Singh, and I. Altintas, "Kepler WebView: A Lightweight, Portable Framework for Constructing Real-time Web Interfaces of Scientific Workflows," in *Proc. of the International Conference on Computational Science, ICCS*, 2016, pp. 673–679.

[9] L. Moreau, B. Clifford, J. Freire, J. Futrelle, Y. Gil, P. Groth, N. Kwasnikowska, S. Miles, P. Missier, J. Myers *et al.*, "The Open Provenance Model Core Specification (v1. 1)," *Future generation computer systems*, vol. 27, no. 6, pp. 743–756, 2011.

[10] C. Lim, S. Lu, A. Chebotko, and F. Fotouhi, "Storing, reasoning, and querying OPM-compliant scientific workflow provenance using relational databases," *Future Generation Computer Systems*, vol. 27, no. 6, pp. 781–789, 2011.

[11] P. Missier, K. Belhajjame, and J. Cheney, "The W3C PROV Family of Specifications for Modelling Provenance Metadata," in *Proc. of the Joint EDBT/ICDT Conferences*, 2013, pp. 773–776.

[12] "Provenance Challenge Series," http://twiki.ipaw.info/bin/view/Challenge/FourthProvenanceChallenge.

[13] A. Chebotko, S. Lu, S. Chang, F. Fotouhi, and P. Yang, "Secure Abstraction Views for Scientific Workflow Provenance Querying," *IEEE Transactions on Services Computing, TSC*, vol. 3, no. 4, pp. 322–337, 2010.

[14] F. A. Bhuyan, S. Lu, R. G. Reynolds, I. Ahmed, and J. Zhang, "Quality analysis for scientific workflow provenance access control policies," in *Proc. of the IEEE Conference on Services Computing, SCC*, 2018, pp. 261–264.

[15] R. Luo, P. Yang, S. Lu, , and M. I. Gofman, "Analysis of Scientific Workflow Provenance Access Control Policies," in *Proc. of the IEEE Ninth International Conference on Services Computing, SCC*, 2012, pp. 266–273.

[16] D. Nguyen, *Provenance-based access control models*. The University of Texas at San Antonio, 2014.

[17] S. Lu and J. Zhang, "Collaborative Scientific Workflows," in *Proc. of the IEEE International Conference on Web Services, ICWS*, 2009, pp. 527–534.

[18] S. Lu and J. Zhang, "Collaborative Scientific Workflows Supporting Collaborative Science," *International Journal of Business Process Integration and Management IJBPIM*, vol. 5, no. 2, pp. 185–199, 2011.

[19] J. Zhang, D. Kuc, and S. Lu, "Confucius: A Tool Supporting Collaborative Scientific Workflow Composition," *IEEE Transactions on Services Computing, TSC*, vol. 7, no. 1, pp. 2–17, 2014.

[20] J. Zhang, Q. Bao, X. Duan, S. Lu, L. Xue, R. Shi, and P. Tang, "Collaborative Workflow Composition as a Service - An Infrastructure Supporting Collaborative Data Analytics Workflow Design and Management," in *Proc. of the International Conference on Collaboration and Internet Computing, CIC*, 2016.

[21] X. Fei, S. Lu, and J. Zhang, "A Granular Concurrency Control for Collaborative Scientific Workflow Composition," in *Proc. of the IEEE International Conference on Services Computing, SCC*, 2011, pp. 410–417.

[22] J. Freire, C. T. Silva, E. S. S. P. Callahan, C. E. Scheidegger, and H. T. Vo, "Managing Rapidly-Evolving Scientific Workflows," in *Proc. of the International Provenance and Annotation Workshop, IPAW*, 2006, pp. 10–18.

[23] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *IEEE Computer*, vol. 29, no. 2, pp. 38–47, 1996.

[24] J. Jin and G. Ahn, "Role-based access management for ad-hoc collaborative sharing," in *Proc. of the 11th ACM Symposium on Access Control Models and Technologies, SACMAT*, 2006, pp. 200–209.

[25] E. Bertino, A. A. Jabal, S. B. Calo, C. Makaya, M. Touma, D. C. Verma, and C. Williams, "Provenance-Based Analytics Services for Access Control Policies," in *Proc. of the IEEE World Congress on Services, SERVICES*, 2017, pp. 94–101.

[26] A. M. Bates, K. R. B. Butler, and T. Moyer, "Take only what you need: Leveraging mandatory access control policy to reduce provenance storage costs," in *Proc. of the Theory and Practice of Provenance, TaPP*, 2015.

[27] W. J. Tolone, G. Ahn, T. Pai, and S. Hong, "Access control in collaborative systems," *ACM Computing Surveys*, vol. 37, no. 1, pp. 29–41, 2005.

[28] J. Chency, "A Formal Framework for Provenance Security," in *Proc. of the 24th Computer Security Foundations Symposium*, 2011, pp. 281–293.

[29] M. Ebrahimi, A. Mohan, A. Kashlev, S. Lu, and R. G. Reynolds, "Task And Data Allocation Strategies for Big Data Workflows," *International Journal of Big Data, IJBD*, vol. 2, no. 2, pp. 28–42, 2015.

[30] M. Ebrahimi, A. Mohan, S. Lu, and R. Reynolds, "TPS: A Task Placement Strategy for Big Data Workflows," in *Proc. of the IEEE International Conference on Big Data*, 2015.

[31] D. Ruan, S. Lu, A. Mohan, X. Fei, and J. Zhang, "A User-Defined Exception Handling Framework in the VIEW Scientific Workflow Management System," in *Proc. of the IEEE International Conference on Services Computing, SCC*, 2012, pp. 274–281.

[32] I. Ahmed, S. Lu, C. Bai, and F. A. Bhuyan, "Diagnosis recommendation using machine learning scientific workflows," in *Proc. of the IEEE International Congress on Big Data*, 2018, pp. 82–90.

[33] F. Bhuyan, S. Lu, I. Ahmed, and J. Zhang, "Predicting Efficacy of Therapeutic Services for Autism Spectrum Disorder using Scientific Workflows," in *Proc. of the IEEE International Conference on Big Data*, 2017.

[34] A. Kashlev, S. Lu, and A. Mohan, "Big Data Workflows: A Reference Architecture and The Dataview System," *Services Transactions on Big Data (STBD)*, vol. 4, no. 1, pp. 1–19, 2017.

[35] "Simons Foundation Autism Research Initiative (SFARI)," https://www.sfari.org/.

[36] F. Bhuyan, S. Lu, D. Ruan, and J. Zhang, "Scalable Provenance Storage and Querying Using Pig Latin for Big Data Workflows," in *Proc. of the IEEE Conference on Services Computing, SCC*, 2017, pp. 459–466.

[37] "Provenance Model PROV-DM," https://www.w3.org/TR/prov-dm/.

[38] V. Atluri and W. Huang, "Security for Workflow Systems," in *Handbook of Database Security Applications and Trends*, 2007, pp. 213–230.

[39] D. Domingos, A. Silva, and P. Veiga, "Workflow Access Control from a Business Perspective," in *Proc. of the International Conferance on Enterprise Information Systems*, 2004, pp. 18–25.

[40] R. A. Botha and J. H. P. Eloff, "Separation of Duties for Access Control Enforcement in Workflow Environments," *End-to-end Security*, vol. 40, no. 3, 2001.

[41] G. Ahn, R. Sandhu, M. Kang, and J. Park, "Injecting RBAC to Secure a Web-based Workflow System," in *Proc. of the fifth ACM Workshop on RBAC*, 2000, pp. 1–10.

[42] G. Herrmann and G. Pernul, "Toward Security Semantics in Workflow Management," in *Proc. of the Thirty-First Annual Hawaii International Conference on System Sciences*, 1998.

[43] V. Atluri and W. Huang, "An Authorization Model for Workflows," in *Proc. of the fourth European Symposium on Research in Computer Security, ESORICS*, 1996, pp. 44–64.

[44] W. Huang and V. Atluri, "SecureFlow: A Secure Web Enabled Workflow Management System," in *Proc. of the fourth ACM Workshop on Role-based Access Control*, 1999, pp. 83–94.

[45] H. Chivers and J. McDermid, "Refactoring Service-based Systems: How to Avoid Trusting a Workflow Service," *Concurrency and Computation : Practice and Experience*, vol. 18, no. 10, pp. 1255–1275, 2006.

[46] R. Martinho, D. Domingos, and A. Rito-Silvas, "Supporting Authentication Requirements in Workflows," in *Proc. of the Eighth*

International Conference on Enterprise Information System: Databases and Information Systems Integration, ICEIS, 2006, pp. 181–188.
[47] J. Warner and V. Atluri, "Inter-instance Authorization Constraints for Secure Workflow Management," in *Proc. of the eleventh ACM Symposium on Access Control Models and Technologies*, 2006, pp. 190–199.
[48] E. Bertino, E. Ferrari, and V. Atluri, "The Specification and Enforcement of Authorization Constraints in Workflow Management Systems," *ACM Transactions on Information and System Security, TISSEC - Special issue on role-based access control*, vol. 2, no. 1, pp. 65–104, 1999.
[49] W. Huang and V. Atluri, "Analysing the Safety of Workflow Authorization Models," in *Proc. of the Twelfth International Working Conference on Database Security*, 1999, pp. 43–57.
[50] A. S. S. Wu, J. Miller, and Z. Luo, "Authorization and Access Control of Application Data in Workflow Systems," *Journal of Intelligent Information Systems*, vol. 18, no. 1, pp. 71–94, 2002.
[51] M. Kang, J. Park, and J. Froscher, "Access Control Mechanisms for Inter-organizational Workflow," in *Proc. of the sixth ACM Symposium on Access Control Models and Technologies*, 2001, pp. 66–74.
[52] P. Hung and K. Karlapalem, "A Secure Workflow Model," in *Proc. of the Australasian Information Security Workshop Conference on ACSW Frontiers*, 2003.
[53] S. Kandala and R. Sandhu, "Secure Role-based Workflow Models," in *Proc. of the Fifteenth Annual Working Conference on Database and Application Security*, 2001, pp. 45–58.
[54] V. Tan, P. Groth, S. Miles, S. Jiang, S. Munroe, S. Tsasakou, and L. Moreau, "Security Issues in a SOA-based Provenance System," in *Proc. of the third International Provenance and Annotation Workshop, IPAW*, 2006.
[55] U. Braun, A. Shinnar, and M. Seltzer, "Securing Provenance," in *Proc. of the 3rd USENIX Workshop on Hot Topics in Security, HotSec*, 2008.
[56] R. Hasan, R. Sion, and M. Winslett, "Preventing History Forgery with Secure Provenance," *Journal of Intelligent Information Systems*, vol. 5, no. 4, pp. 12:1–12:43, 2009.
[57] J. Chency, U. A. Acar, and A. Ahmed, "Provenance Traces," in *Proc. of the Computing Research Repository, CoRR Extended report*, 2008.
[58] S. Chong, "Towards Semantics for Provenance Security," in *Proc. of the First Workshop on the Theory and Practice of Provenance, TaPP*, 2009.
[59] S. B. Davidson, S. Khanna, S. Roy, J. Stoyanovich, V. Tannen, and Y. Chen, "On Provenance and Privacy," in *Proc. of the 14th International Conference Database Theory, ICDT*, 2011, pp. 3–10.
[60] M. R. Asghar, M. Ion, G. Russello, and B. Crispo, "Securing Data Provenance in the Cloud," in *Proc. of the International Federation for Information Processing, IFIP*, 2012, pp. 145–160.
[61] R. Hasan and R. Khan, "Unified Authentication Factors and Fuzzy Service Access using Interaction Provenance," *Computers & Security*, vol. 67, pp. 211–231, 2017.
[62] I. Jung and H. Yeom, "Provenance Security Guarantee from Origin up to Now in the e-Science Environment," *Journal of Systems Architecture*, 2010.
[63] R. Lu, X. Lin, X. Liang, and X. Shen, "Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing," in *Proc. of the 5th ACM Symposium on Information, Computer and Communications Security, ASIACCS*, 2010, pp. 282–292.
[64] R. Aldeco-Pérez and L. Moreau, "Securing provenance-based audits," in *Provenance and Annotation of Data and Processes - Third International Provenance and Annotation Workshop, IPAW*, 2010, pp. 148–164.

Fahima Amin Bhuyan is currently working toward the PhD degree in the Department of Computer Science, Wayne State University. She is currently a member of the Big Data Research Laboratory. Her current research interests include scientific workflows, provenance, big data and their applications, provenance security. She is a student member of the IEEE. She can be reached at fahima.amin@wayne.edu.

Shiyong Lu Ph.D., is a Professor in the Department of Computer Science at Wayne State University, and the director of the Big Data Research Laboratory. Dr. Lu received his Ph.D. in computer science from Stony Brook University in 2002. Dr. Lu's current research interests focus on scientific workflows, services computing, big data security, and provenance management. Dr. Lu is an author of two books and over 120 articles published in various international journals and conferences, including IEEE Transactions on Services Computing (TSC), Data and Knowledge Engineering (DKE), IEEE Transactions on Knowledge and Data Engineering (TKDE). He is the founding chair of the IEEE International Workshop on Scientific Workflows (SWF) and a Co-Editor-in-Chief of the International Journal of Cloud Computing and Services Science. He is a founding editorial board member of International Journal of Big Data and a senior member of the IEEE. Dr. Lu can be reached at shiyong@wayne.edu.

Robert Reynolds Ph.D., is an Professor of Department of Computer Science at Wayne State University. His current research interests focus on Artificial Intelligence, Game Programming, Artificial Intelligence in Games, Machine Learning, Evolutionary Computation, Cultural Algorithms, Multi-agent systems, intelligent agents, and multi-objective problem solving. He is a Senior Member of the IEEE. He can be reached at robert.reynolds@wayne.edu.

Jia Zhang Ph.D., is an Associate Professor of Department of Computer Science at Carnegie Melon University Silicon Valley. Her current research interests center on Service Oriented Computing, with a focus on Internet-centric collaboration, intelligent services, and semantic service discovery. She has co-authored 1 book and published over 160 journal articles, book chapters, and conference papers. Zhang is an associate editor of IEEE Transactions on Services Computing (TSC). She is a Senior Member of the IEEE and can be reached at jia.zhang@sv.cmu.edu.

Ishtiaq Ahmed is currently working toward the PhD degree in the Department of Computer Science, Wayne State University. He is currently a member of the Big Data Research Laboratory. His current research interests include scientific workflows, workflow scheduling, provenance and big data. He is a student member of the IEEE. He can be reached at ishtiaq@wayne.edu.