2018 IEEE International Conference on Services Computing

Quality Analysis for Scientific Workflow Provenance Access Control Policies

Fahima Amin Bhuyan, Shiyong Lu, Robert Reynolds, Ishtiaq Ahmed
Department of Computer Science
Wayne State University
Detroit, MI
fahima.amin, shiyong, robert.reynolds, ishtiaq@wayne.edu

Jia Zhang
Electrical and Computer Engineering
Carnegie Mellon University
Mountain View, CA
jia.zhang@sv.cmu.edu

***Abstract*—The notion of collaborative scientific workflow was coined to address the increasing need for collaborative data analytics using the scientific workflow technique. In such collaborative environments, adequate access control policies are necessary for controlling the sharing of workflows, data products, and provenance information among collaborating parties. Meanwhile, it is important to ensure that the evolution of workflow provenance access control policies meets certain qualities to guarantee the correctness and performance of the policy enforcement engine. To address this concern, this paper proposes a role-based access control model for scientific workflow provenance. Three quality requirements are defined for scientific workflow provenance access control policies - consistency, completeness, and conciseness. A mapping mechanism from the specifications of workflows to their counterparts in the provenance is developed to preserve quality properties.**

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

I. Introduction

Provenance is information about the history, origin, derivation, and context of data. Provenance is useful to interpret an analytical result, to repeat a scientific discovery, and to trace errors in the data. Provenance is also a useful vehicle to answer lineage queries and to decide the trustworthiness of a data product. Therefore, provenance management has become critical in various data systems [1], [2], [3].

The provenance security problem is critical for modern scientific workflow systems [4], [5]. Unauthorized access to provenance information might disclose confidential details about the related data products. The code for the collection, the querying and the mining of provenance can be compromised, forged, or replayed by intruders. Compromised provenance can lead to misinterpretation of the analytical results, unintentional errors, and can compromise the confidentiality of related datasets. 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 using the scientific workflow technique [6], [7], [8]. In such collaborative environments, adequate access control policies are necessary for controlling the sharing of workflows, data products, and provenance information among collaborating parties [4], [9], [10], [11]. In this research, we focus on the secrecy of provenance information 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 will be a discussion of existing access control mechanisms proposed for the protection of the confidentiality of scientific workflow provenance [4], [5].

The remainder of the paper is organized as follows. Section II to VI will discuss our provenance security framework, policy life span, policy specification, policy enforcement, and policy requirements and analysis. Section VII will draw conclusions.

II. Provenance security framework

In this section, we will introduce a provenance security framework, where formal and precise security properties such as confidentiality, privacy, and availability are needed for enforcing suitable and desired security policy.

We illustrate our workflow provenance security mechanism in the context of a real-life example of collecting data from the SFARI project about Autism Spectrum Disorder(ASD). The autism workflow created in the DATAVIEW [12], [13] system is used here containing 10 tasks. As shown in Fig. 1, the workflow explores all of the unique attributes of a child's family history, education history and medical history, and identifies predictive features pertaining to each individual child. Both tasks T_1 and T_2 perform a Projection task, which projects the predominant attributes out of a pool of attributes. Based on the SFARI id, the task T_3 then performs another Natural Join operation. Task T_4 performs a Projection on SFARI's follow-up family history dataset. On the retrieved result of both tasks T_3 and T_4, a natural join operation T5 is performed. Task T_6 checks that if there is 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 predicts dataset retrieved by executing a data mining task T_{10}. For data mining and predicting, a test dataset is required, and that test dataset is provided to task T_9 to convert it to the ARFF format. After accumulating a training set, a test test, and sample numbers of tree parameters, we get the final prediction result. After executing this workflow, we illustrate the provenance information from the detailed run

2474-2473/18/$31.00 ©2018 IEEE
DOI 10.1109/SCC.2018.00044

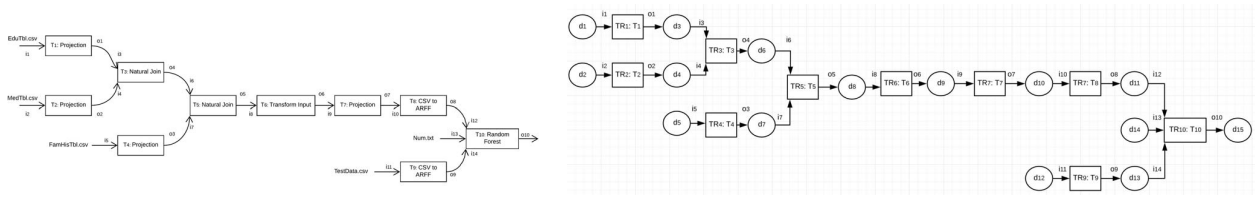

Fig. 1: Workflow View and Corresponding Provenance View of Autism Spectrum Disorder Mining.

in Fig. 1. Here, circles represent data products, and rectangles represent workflow task runs. The edges between data products and tasks are relations. For example an edge from a data product to a task is called wasGeneratedBy relation, and an edge from a task to a data product is call used relation.

To secure the provenance in such a workflow, we propose a workflow provenance security framework comprising a directed provenance graph based on the PROV-DM standard, equipped with access control policy.

Definition II.1 (Role Based Access Control)**.** *Let Role-Based Access control $\hat{R}$ for provenance security be defined as a 7-tuple (U, R, A, W, E, ϕ, μ), where:*

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

The function ϕ is further defined as follows:

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

The element could be either a task, a port, or a data channel. For task we define the function Γ, for the port we define the function ρ, and for the data channel we define the function δ. The functions Γ, ρ and δ will be defined in detail in the following sections.

III. Provenance Security Policy Life Span

A Provenance security policy life cycle is comprised of four iterative stages: i) Security policy specification, ii) Security policy enforcement, iii) Security policy analysis, and iv) Security policy evaluation. The administrator of the access control policies coordinates with the system users and determines the policies to be enforced in one or more tasks at the port and data channel levels. In a security policy enforcement stage, the policies are applied to either grant or restrict access based upon how system users access the protected elements. The security policies evolve to adopt to changes in the current execution environment. In a policy analysis phase, policy quality requirements are analyzed. This phase analyzes the policy qualities like consistency, completeness, and non-redundancy in order to make sure the proposed policies adhere to all predefined qualities. Finally, in a policy evaluation phase, quality requirements are evaluated and any quality discrepancy is identified and modified. Fig. 2 shows a graphical representation of provenance security policy lifespan.

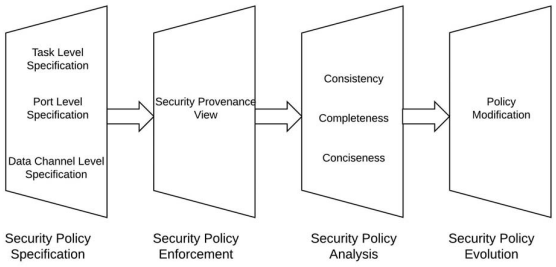

Fig. 2: Provenance Security Policy Life Span.

IV. Security Policy Specification

A. Task Level Specification

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

$$\Gamma(t, r, \alpha) = \begin{cases} \Pi(t, r, \alpha), & \textit{if specified and} \quad (2a) \\ & !\Gamma(t_p, r, \alpha) \wedge \rho(t, r, \alpha) \quad (2b) \\ \Gamma(t_p, r, \alpha), & \textit{if not explicit} \quad (2c) \\ Invalid & \textit{Otherwise} \quad (2d) \end{cases}$$

In task specification, the access permission can be annotated by + or -. Here we define a function Π: $R \times E \times A \rightarrow \{0, 1\}$, which returns permission of role, element, and action triplet if they are explicitly specified in the RBAC. If the permission is not explicitly specified in RBAC, then the child task t can inherit permission from task t_p, where t_p denotes a parent of t, $\alpha \in A$, $r \in R$. In other words, the task-level security specification, if explicitly stated, is validated against consistency requirement of the protocol. In this case, if the parent task does not have security access then its child task inherits the restriction, and an explicit specification cannot

override this restriction. One important feature of a task is that when it is annotated as + then all other task, ports or data channels contained in task T should be accessible, otherwise a − annotation is explicitly specified or derived for them.

B. Port Level Specification

Definition IV.2 (Port Annotation)**.** *A port-level specification is denoted by ρ:P × R × A → {0, 1} that maps specific roles and ports to the permission level and is defined by:*

$$\rho(p,r,\alpha) = \begin{cases} \Pi(p,r,\alpha), & \textit{if specified and} \quad (3a) \\ & !\Gamma(t_p,r,\alpha) \wedge \rho(t,r,\alpha) \quad (3b) \\ \Gamma(t_p,r,\alpha), & \textit{if not explicit} \quad (3c) \\ Invalid & \textit{Otherwise} \quad (3d) \\ & (3e) \end{cases}$$

Ports can be specified with + or -. In port-level specification, when a port has no specified security specification then it inherits either access or denied permission from its owning task. The administrator can explicitly specify all or some ports access permissions. For all workflow runs, the port annotation + or - specified for any given task will demonstrate the accessibility of the data products.

C. Data Channel Level Specification

Definition IV.3 (Data Channel Annotation)**.** *A data channel-level specification is denoted by δ:P × R × A → {0, 1} that maps specific roles and ports to the permission level and is defined by:*

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

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

V. Security Policy Enforcement

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

We define security provenance view as a restricted view of provenance consisting only of the information that users are authorized to access.

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 inverse 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 V.1 (Security Provenance View of 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 V.2 (Security Provenance View of 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)

Now we illustrate security policy requirements based on the Autism provenance system in Section IV and defines those access control policies in Table I.

TABLE I: An Example: Role Based Access Control Policies of Provenance System for Autism Spectrum Disorder.

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	+

VI. Security Policy Quality Requirements and Analysis

We define and illustrate our security policy quality requirements below:

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

Here we infer consistency between two policies acp_i and acp_j if for the same user, the same role, the same element, and the same activity, both policies should have the same access rights. If one policy allows access, it implies that the other policy allows access as well. If there is any inconsistency in policy, it requires conflict resolution to produce a consistent policy.

Example 1**:** As shown in Table. I, in the teacher role, acp_{14} and acp_{15} are not consistent. Based on our specification, both policies need to have the same access rights when they act in the same role, user, element and activity. Here acp_{14} and acp_{15} do not meet the criteria. They are inconsistent because one port is specified with negative access whereas at the other end of the data channel another port is specified with positive access. Also, 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 the same permission. In this case, the output and the input port of a single data channel have different permissions. Therefore, it is an inconsistent policy.

B. 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 when for any role, the access control policy is defined. A complete access control policy has both a role and an access policy defined. An incomplete policy has either a role or access policy for task/port undefined, or both.

Example 2**:** In Table. I, there is no access control policy for the teachers role in allowing or denying access to 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 defined for accessing or denying the information of family history is incomplete.

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

The Conciseness of an access control policy means that for any two policies, if they have the same role, the same element, the same action, and support the same permission then they are said to be concise. 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 access policies then we infer that these two policies are not concise.

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

VII. Conclusions

In this work, we have examined access control policies for data products and derivation history for protecting sensitive data and processes. We have formalized secure scientific workflow specification for task, port and data channel and analyzed the policies in perspective of policy quality requirements. We have also formalized the security view for provenance based on a mapping between workflow and provenance.

In the future, we will consider conducting security case studies with more complex data patterns and integrate our access control policies to deal with a different granularity of data. We will also study cases of usability of our system.

Acknowledgment

This work is supported by National Science Foundation, under grant NSF ACI-1738929, ACI-1443069, CNS-1747095, and 1747095. In addition, this material is based upon work supported in part by the National Science Foundation under Grant No. 1443069 and 1744367.

References

[1] P. Buneman and W. C. Tan, "Provenance in databases," in *ACM 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 *International Conference on Management of Data, SIGMOD*, 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. Chebotko, S. Lu, S. Chang, F. Fotouhi, and P. Yang, "Secure abstraction views for scientific workflow provenance querying," *IEEE Transaction on Services Computing*, vol. 3, no. 4, pp. 322–337, 2010.
[5] R. Luo, P. Yang, S. Lu, and M. I. Gofman, "Analysis of scientific workflow provenance access control policies," in *IEEE Ninth International Conference on Services Computing*, 2012, pp. 266–273.
[6] S. Lu and J. Zhang, "Collaborative scientific workflows," in *IEEE International Conference on Web Services, ICWS*, 2009, pp. 527–534.
[7] S. Lu and J. Zhang, "Collaborative scientific workflows supporting collaborative science," vol. 5, no. 2, 20011, pp. 185–199.
[8] J. Zhang, D. Kuc, and S. Lu, "Confucius: A tool supporting collaborative scientific workflow composition," *IEEE Trans. Services Computing*, vol. 7, no. 1, pp. 2–17, 2014.
[9] G. Ahn, R. S. Sandhu, M. H. Kang, and J. S. Park, "Injecting RBAC to secure a web-based workflow system," in *Fifth ACM Workshop on Role-Based Access Control, RBAC, Berlin, Germany, July 26-27*, 2000, pp. 1–10.
[10] V. Atluri and J. Warner, "Security for workflow systems," in *Handbook of Database Security - Applications and Trends*, 2008, pp. 213–230.
[11] P. C. K. Hung and K. Karlapalem, "A secure workflow model," in *The Australasian Information Security Workshop (AISW) and the Workshop on Wearable, Invisible, Context-Aware, Ambient, Pervasive and Ubiquitous Computing (WICAPUC)*, 2003, pp. 33–41.
[12] F. A. Bhuyan, S. Lu, D. Ruan, and J. Zhang, "Scalable provenance storage and querying using pig latin for big data workflows," in *IEEE International Conference on Services Computing, SCC*, 2017, pp. 459–466.
[13] F. A. Bhuyan, S. Lu, I. Ahmed, and J. Zhang, "Predicting efficacy of therapeutic services for autism spectrum disorder using scientific workflows," in *IEEE International Conference on Big Data*, 2017, pp. 3847–3856.