Comments from the editors and reviewers:

-Reviewer 1

-

The article describes a formal method for hiding and obfuscating PROV-encoded provenance documents. In this method, users select provenance nodes of interest to hide, and the system expands the initial node selection to encompass a sub-graph, replacing it with similar type nodes. The objective is to show the user a valid provenance graph consisting of new and original nodes and edges, where in validity is checked based on temporal constraints and based on event orderings described in PROV standards.

Strengths:

S1: Unlike past methods which have not explicitly considered the type of hidden/obfuscated/summarized nodes, this paper assigns a specific type, and with the designated type aims to present a valid provenance graph to the user.

S2: The paper leverages bi-partite graph property of provenance graphs to argue for validity of the structure of the graph, and uses it to justify an provenance event ordering. The mathematical definitions are sound and cleanly described, though they can be further simplified (see below).

S3: The paper extends the operators to include hiding of agent nodes, not considered formally in previous methods.

Weaknesses:

W1. The motivation scenario is quite weak. The authors provide two rationales for abstracting provenance information: complexity and selective disclosure. The objective of this paper is to abstract primarily due to selective disclosure as presented in the motivating example. However, the motivating examples provides no basis for subgraph obfuscation. Para2 on page 3 provide some reasons for eliminating node information but they are all node property information not rationale for subgraph obfuscation. Figure 2 is nearly unreadable. It is not clear what real world phenomenon this graph represents. It is not obvious why a user in Figure2 would like to remove nodes consolidate AJC and consolidateBNC, and what impact will that removal have—in the sense what information would have divulged otherwise and how it will affect. The scenario is presented in loose and vague terms with significant hand waving: Pg4, para 3, “some of it may be…”

Need to design a scenario that motivates node removal (or adapt current one to), then justify subgraph removal by pointing to validity. We should ensure that in the example a user can decide they want to obfuscate information *without* needing to know anything about the structure of the graph. This is to counter W7 below.

Action: PM

W2: Definitions of two crucial elements are note clear. It is not obvious what Vgr represents: does it represent “a set” of nodes to obfuscate or does the user have any knowledge that there exists a path between the nodes in Vgr and so chooses only those nodes? In other words, what if Vgr nodes have no path amongst them?

We now say that Vgr represents a set of nodes that the user wishes to obfuscate. (Done)

On pg 5 made clear that Vgr is chosen by user. Also in sec 4 page 12.

Early on, I think we should propose an framework in which an algorithm operates over the selected nodes and calculates “optimal” groupings (between secs 3 and 4), then focus on building a single group in section 4 (so no changes required there) Point out the importance of domain knowledge in setting the parameters for this algorithm. No need (I hope!) to go into the details of the algorithm.

Identify connected components.

Action: JWB – Approach in revision: on p 13 (defn of path closure), make the assumption around connected components of the undirected graph. (Done)

When we relax this assumption, we will need to demonstrate that the order in which we carry out “Group” operations is irrelevant.

I’ve also corrected pclos operator to take (Vgr, G) as parameters, not (Vgr,V).

If they have no path then are they replaced by as many “blank” or “null” nodes as in the set Vgr, and if such nodes for validity only retain their original type?

This becomes (sort of) anonymization, but if we enforce the type (Ag/Ev) in advance, then pick a single node of the other type, things will get messed up. We’ll have to discuss this as well.

If so, then why can’t this property be adopted universally instead of opting for expensive operations of path closure, extend, replace.

Because we want to obfuscate graph structure as well as node typing information.

Why is path closure referring to “a” path between nodes and not all paths?

If the idea is indeed to obfuscate the lineage between the selected nodes, it must include all paths and not “a” path.

Pg 12: Agreed unclear. Refer to all paths.

Done

W3: The motivation for extension operator is not clear.

Need to include an explicit discussion of “convex hull” concept, and re-iterate the importance (to us) of maintaining graph validity. We don’t dwell on this in the text, but it is an important part of the explanation.

It is not evident what is the impact of hiding information which the user did not select, especially information that was obfuscated to maintain validity? What if the non selected obfuscated content is actually the information that must be communicated between the two parties.

Point back to prev paper where we deal with this via a proposed confidentiality and relevance ranking for each node. Utility.

The authors have not considered what domains enforce a validity constraint and what if it is relaxed to show an partially inconsistent graph?

point forward to next paper, where domain constraints are more to the fore. Include some examples of domains that might enforce validity: ones that involve significant interoperability? (e.g. e-science?)

Alternatively what if nodes are replaced by a subgraph? Thus in Figure 6 instead of replacing by a single node of type e’, what if a replacement subgraph of e’->a’ is provided. This subgraph signifies some entities connect to an activity, and the edge remains unlabeled (similar to the edge in Figure 6(d)). If replaced by a sub-graph instead of an entity or activity, it can be shown that no new nodes need to be included in the obfuscation cover to maintain validity. In general? It can also be shown that it represents the orginal subgraph correctly because the original consists of both activities and entities.

Good point. I suggest this is out-of-scope for this paper, and that we want to restrict ourselves here. Act that this is an alternative idea.

In conclusions we could point forward to next paper.

Action: JWB

W4: The replacement on page 13 has little evidential basis. It is an incorrect operator not a naive one, in which the structure of a graph is destroyed to lead a new node for the purpose of reducing complexity or performing selective disclosure. That operation may be a transient state in 3, 12, 18 but is never performed. I don’t know what the reviewer means here. Are these definitions 3, 12 and 18? I am not sure what is the basis for considering a wrong operator and terming it as a naive operation.

The purpose was to motivate the work that we do in 4.1. In particular extend.

It looks like we either labour this point too much or don’t make it clearly enough for this reviewer. Just make the point more succinctly that simple replacement violates the graph properties? Drop the diagram? Def drop the word “naïve” since the reviewer takes offence at it.

Action: jwb

W5: Despite a mathematical foundation, the paper lacks complexity analysis of the operators. It has no experimental results to validate how efficient are these operators on real provenance graphs. The authors have provided no basis for why these operators should be considered.

We should at least do a complexity analysis for Group and Group\_hom.

Could we generate synthetic prov and time Carls’ toolbox?

I’ve done some initial thinking around a complexity analysis. Attached is a paper to provide some inspiration as to what the reviewer wants here.

Action: PM

W6: Definition 6 on homogenous grouping is not re-constructed based on outcut and incut definitions considered earlier. Again incut and outcut are not highlighted in the Figure to understand their definitions, ,making the definitions very hard to read. There is a lingering comment in this definition.

Do this reconstruction. Highlight incut and outcut in figures 6(b) and 6(c).

Action: JWB

Comment removed

W7: Related Work:

1. It is not obvious why in Zoom the user has to have an understanding of workflow structure while this is not the case in their approach. In their approach also the user is selecting nodes and activities to eliminate (similar to selecting relevant workflow modules). Similar to Zoom, the paper considers validity of the provenance graph, except this paper is speciifc to PROV semantics.

If we do as I suggest in answer to W3, this point has less weight.

2. The distinction to compressing provenance graphs [18] is not evident. The proposed technique seems a strategic spin on the lineage by types. More clarification is needed as to who summaries are different than obfuscation or how a result produced by that technique is different than the result produced by path/extend/replace.

[18] This seems to be about grouping multiple graphs by type, similar to our current paper.

Other comments

Figure 7 (b) e5 should be shaded.

Agreed

Meeting 17-01-2019

Actions:

* PM: need algorithm for pclos to agrue complexity. Is there a dynamic programming argument for lower than V^(no of outgoing paths) -- create a lookup table of all paths, once.

Baseline for reachability is Floyd marshall.

Note: only has to happen once.