Security Issues in a SOA-Based Provenance System

Victor Tan, Paul Groth, Simon Miles, Sheng Jiang, Steve Munroe, Sofia Tsasakou, and Luc Moreau

School of Electronics and Computer Science
University of Southampton
Southampton, SO17 1BJ, UK
vhkt@ecs.soton.ac.uk

Abstract. Recent work has begun exploring the characterization and utilization of provenance in systems based on the Service Oriented Architecture (such as Web Services and Grid based environments). One of the salient issues related to provenance use within any given system is its security. Provenance presents some unique security requirements of its own, which are additionally dependent on the architectural and environmental context that a provenance system operates in. We discuss the security considerations pertaining to a Service Oriented Architecture based provenance system. Concurrently, we outline possible approaches to address them.

1 Introduction

The concept and utilization of *provenance* has been recently explored in the areas of Grid and Web Services-based systems and environments. The myGrid project implemented a system for recording the documentation of process in the context of in-silico experiments represented as workflows aggregating Web Services [8]. The GriPhyn Virtual Data System project provides a set of tools for expressing and executing workflows in a Grid environment, where the definitions of the workflows are specified in a high-level workflow language and are stored in a catalog to provide for tracking of provenance of all files derived by the workflow [7]. A trial implementation of an architecture based around a workflow enactment engine was used to demonstrate several mechanisms for handling documentation about the invocation of various Web Services was presented in [17]. Studies are now being conducted towards assessing the use of provenance in large scale applications [3].

Most of the work described however does not explicitly consider *security requirements* revolving around the utilization of provenance. Such an omission will hinder eventual evolution of these systems to industrial strength level, where security is likely to be of primary consideration. This is particularly applicable where provenance is concerned with information of a commercially or legally sensitive nature. Our paper seeks to address this shortcoming by analyzing some

L. Moreau and I. Foster (Eds.): IPAW 2006, LNCS 4145, pp. 203–211, 2006.

of the security issues that arise within a generic provenance system based on a Service Oriented Architecture (SOA). The primary contributions are:

- Discussing basic security issues within such a system;
- Discussing security issues that arise from scalability concerns.

In the next section, we provide a brief description of the provenance representation we employ; the motivation and justification for this type of representation has been covered extensively elsewhere ([9], [11]). The basic security issues a provenance architecture employing this representation are expounded upon in length in Section 3. Section 4 examines new issues that arise as a result of attending to scalability concerns, and we conclude in Section 6.

2 Provenance Representation

We discuss provenance in the context of the Service Oriented Architecture view [5], which provides the underlying architectural basis for the Web Services / Grid environment. In this view, services are simply considered as components that take inputs and produce outputs, which can be brought together to solve a given problem typically via a workflow that specifies their composition. Interactions with services take place using messages that are constructed in accordance with service interface specifications. In a SOA, clients typically invoke services, which may themselves act as clients for other services; we use the term *actor* to denote either a client or a service in a SOA. We refer to the execution of a workflow that is composed of these interacting actors as a *process*.

We adopt the following definition for provenance within an SOA: the provenance of a piece of data is the process that led to that piece of data. Such a provenance will be represented by some suitable documentation of the process (i.e. workflow execution) that led to the data ([10,9]). It is possible to distinguish between a specific piece of information documenting some step of a process from the whole documentation of the process. We refer to the former as a *p-assertion*, which is essentially an assertion made by an actor pertaining to any aspect of a process. Equivalently, the documentation of a process would therefore consist of a set of p-assertions made by all the actors involved in that process.

The various logical components of an architecture for a SOA-based provenance system is detailed in [12]. For purposes of our discussion in the following chapter, we note that the key feature of this architecture is a dedicated repository for holding only process documentation, which we term a *provenance store*. Actors creating p-assertions about a particular process store them into a provenance store. Actors who wish to answer provenance queries (i.e. queries about the provenance of various data items produced during that process) would submit these queries in an appropriate format to the provenance store, which in turn would return the set of p-assertions holding the necessary data required to answer the query. The answering of queries in the provenance store could be augmented.

3 Security Issues in a SOA Provenance System

We classify our discussion into several main areas of security concern:

3.1 Enforcing Access Control over Process Documentation

An obvious security requirement is the need to control access to process documentation. Access control to data in general is a well studied subject for which many practical techniques already exist. A typical approach in many of these techniques is to identify the sensitivity of information within a specific data item (a database record, for example) and then restrict access to a user base in accordance to their predefined roles or identities. Extending this idea directly to our provenance system by restricting access on the basis of individual p-assertions may not be useful, as p-assertions do not generally provide much useful information if accessed as individual data items. Rather, information about a specific aspect of a process (such as all the services that participated in the production of the final result of a workflow) would be obtained by processing the data contained within an appropriate aggregation of p-assertions from the entire set of p-assertions that constitute the process documentation for the workflow in question.

A useful way then to delineate access control boundaries in this example might be to identify different types of provenance related information with differing levels of sensitivity that can be obtained from processing different groups of p-assertions, and then structure access control on the basis of these groups. Since it is likely that a single p-assertion can belong within many groups, there is now the problem that a user without access to a designated group of p-assertions for a specific purpose may still be able to gather together all the constituent p-assertions of this designated group via his or her access to other grouping of p-assertions that inadvertently contains smaller parts of this designated group.

This is a more general problem related to the issue of inference control, i.e. preventing the inferring of information from existing information [18]. Various approaches to address this concern have been proposed within the context of statistical databases, such as perturbing the stored information and query restriction [16]. Provenance complicates the situation because relationship information between the p-assertions is explicitly stored as well, which significantly eases the ability to infer new information from information in existing p-assertions. Hence, existing approaches are likely require modification to tailor them to this environment. A possible solution may involve supporting the specification of access control authorizations at the granularity of these groups and their associated provenance queries. In addition, suitable cryptographic protocols can be used to ensure that users cannot access data within a set of p-assertions returnd as a result of a provenance query, unless they have access rights to all the groups that those p-assertions belong to.

We believe that this problem presents a unique angle on access control for data from the perspective of the data being process documentation in a provenance system. More in-depth investigation into this aspect is required if coherent

access control on process documentation and the subsequent provenance related information derivable from it is to be achieved in industrial strength provenance systems.

3.2 Trust Framework for Actors and Provenance Stores

In a large scale distributed environment, actors that create and store p-assertions regarding specific events of interest may not be directly under the control or even known to actors that will eventually use these p-assertions in some manner to answer a provenance query. Signatures provide a way to link actors with p-assertions they create; a methodology is now needed to provide a trustworthiness measure or rating to specific actors and their p-assertions. Ratings could be based on independent third party ratification of the accuracy of the p-assertions or subjective opinions of all potential consumers of p-assertions produced by specific actors. The methodology could also include methods to provide an aggregated measure of reliability of information obtained from processing a group of p-assertions with different levels of associated trustworthiness.

Similar comments are equally applicable to provenance stores; querying actors could elect to establish trust in provenance stores instead and assume that the stores will in turn assume the responsibility of filtering p-assertions from the various actors that send p-assertions to it for storage. There is clearly some work to be done in articulating the various trust models and relationships possible between actors producing and utilizing p-assertions as well as the provenance store holding these p-assertions. Work of this nature could ideally draw on existing extensive work in the area of trust and reputation in agent mediated interactions [19].

3.3 Accountability and Liability for p-Assertions

An important consideration in any provenance system is the accuracy or objectivity of the documentation recorded. In our representation, a p-assertion is a statement about some aspect of a process by an actor. From a more abstract viewpoint, this statement is however only a subjective view of that aspect by an actor. It can be difficult sometimes, if not impossible, to determine how closely this view tallies with actual reality. This is particularly true in our system, where all information about past processes is only obtainable via actor-created p-assertions. With respect to this, it becomes paramount to forge a clear link between an actor and an assertion that it is responsible for. Such a link, which can be provided through digital signatures, ensures that responsibility and corresponding liability is attributable to the correct actor.

Since p-assertions are created within the context of a process that they describe, actors may elect to include metadata within a p-assertion that links it to another p-assertion created by another actor within that context. Incorporation of incorrect metadata in a p-assertion could potentially create a chain of p-assertions that are incorrectly associated, making it difficult or impossible for a querying actor to correctly answer a provenance query. Again, signatures on this metadata ensures responsibility is attributable to the correct actor. We note

that signatures on p-assertions also serve an additional purpose of guaranteeing their integrity and ensuring that no other parties (for example, the provenance store or other intermediary actors that access the p-assertions) change them intentionally or accidentally.

3.4 Sensitivity of Information in p-Assertions

In a basic example, the p-assertion pertaining to a message exchange between two actors would simply contain the contents of that message verbatim. Depending on application domain requirements however, parts of the message may need to be obscured or transformed in some way when they appear in a p-assertion. A good example of this is found in the electronic health care records domain, where privacy requirements mandate that patient identity on health care records be anonymized ([14], [2]) if the information on the record is being utilized for non-diagnostic reasons (for example, to answer provenance questions about a medical process). If p-assertions are utilized in such a context, then certain data items (such as patient identifiers) that are transmitted in cleartext in the original message exchange between actors must be obfuscated in some manner when stored as part of any created p-assertion.

Along similar lines, there may be situations where an actor may want to ensure that certain parts of the p-assertion it creates is only accessible to certain parties. In the simplest case, this can be achieved by ensuring the appropriate access controls are instituted on the provenance store. However, once a p-assertion is retrieved from the provenance store, it is very difficult to control which parties it is subsequently propagated to. If the asserting actor shares a secret key with certain parties, it can elect to encrypt parts of the p-assertion with this key so that only those parties are able to view it.

3.5 Long Term Storage of p-Assertions

Another issue surrounding provenance storage is long term archival of p-assertions. As p-assertions are signed (and possibly encrypted) prior to storage, there will subsequently be a need to verify the signatures or decrypt them when they are extracted for processing. The certificates for the corresponding encryption / signing keys may expire if the storage duration is substantial, and in extreme cases, the underpinning cryptographic algorithms may themselves become outdated. Such issues must be catered for in some way, for example, by having a key archival facility and re-signing / re-encrypting provenance information periodically over the intended storage duration.

3.6 Creating Authorisations for New p-Assertions

It is likely that p-assertions contain or are derived in some fashion from an existing piece of data in the system. For example, an actor with access to a database may send a message containing an item from that database to another actor. This item is likely to have certain access control restrictions enforced upon it within the security domain of the database in question. When a p-assertion

is created for the transmitted message and recorded to the provenance store, appropriate authorisations must now be established for this new entry to ensure that any future access to it is in accordance with the security policies of the provenance store. Such authorisations may be articulated in the form of access control at the level of groups of p-assertions, as discussed in Section 3.1.

In many cases, it is useful to relate the authorisation for the newly recorded p-assertion in some way to the access control restrictions on the original database item that the p-assertion is based upon. This effectively allows for a more flexible specification of authorisations on p-assertions by taking into account information other than that found in statically predefined security policies on the provenance store. A possible approach towards this end is for an actor to submit additional information along with the p-assertion to be stored. This additional information would be provided by the actor and can then be utilised in an automated manner by the provenance store to generate appropriate authorisations for the new p-assertion.

On the issue of relating authorisations of p-assertions to the authorisations of the data that the p-assertions are based on, we note that an interesting situation may sometimes arise where a more stringent level of access control is mandated on the p-assertions themselves rather than the original data. As an example, consider a bioinformatics domain, where a new drug might ostensibly be designed through some dynamic, unplanned novel application of a standard workflow involving publicly accessible data. In such an instance, the exact sequence and logic of the workflow itself (which can be reconstituted from its provenance) becomes more valuable than the actual data used in the workflow, hence necessitating tighter access controls on it.

3.7 Summary

The first security consideration (Section 3.1) we believe is unique to process documentation intended for provenance purposes; data intended for generic processing is unlikely to have such a requirement. The remaining considerations however are likely to be applicable as well when considering the securing of access to data in the general case. The last two considerations (Section 3.5 - 3.6) are additional enhancements to a provenance security architecture that already adequately addresses the core concerns of access control and non-repudiation. They are not intended to further secure the system, but rather to extend flexibility in the enforcement of security: always an important consideration towards increasing the acceptance and adoptability of security measures.

4 Scalability Related Security Issues

So far our discussion has revolved around the notion of a centralized provenance store, but in practice this will inevitably be distributed for the usual reasons of scalability: the elimination of a central point of failure, the spreading of demand across multiple stores and the ability for stores to exist in different network areas.

In such a situation, related p-assertions (such as p-assertions from two actors pertaining to a message exchange between them) could be recorded in different provenance stores. Actors may then record pointers or links to other provenance stores additionally with or as part of the p-assertions in order to provide a trail for interested parties to retrieve related p-assertions. Such links must again be made attributable to actors through signatures, with a similar motivation as well. Distributed provenance stores may exist in different security domains; hence parties that are recognized and authorised for specific actions on a provenance store in one domain may be unrecognized or be granted different access levels in a provenance store of a different domain. In this instance, a federated identity management infrastructure must be operated and installed in order to permit the authorised parties to follow the trail of links and retrieve all relevant distributed p-assertions.

On a similar theme, if p-assertions themselves are copied or moved between stores that are located in different security domains (for example, in staging of data or for load distribution purposes), the access control restrictions on them in their new destinations needs to be defined. In the simplest case, the newly moved or copied p-assertions retain the same access control restrictions that were associated with them in their original domain and the federated identity infrastructure will function to ensure that any newly introduced identities are recognized appropriately in the correct domain.

5 Related Work

The issues of access control, authorization, integrity and privacy within the context of generic databases is well known and numerous approaches have been proposed and implemented [15]. Within the area of data mining, the issue of maintaining user privacy has become paramount and a large amount of work is ongoing in this area to develop more efficient techniques towards this end [1,6]. Related work explicitly investigating provenance-related security issues is however still relatively scarce. In [4], an abstract security model was developed by identifying generic security relevant attributes based on user requirements across a large range of application domains. The myGrid project [20] also investigated security issues and solutions, but in a manner that was highly application dependent. Specific implementation details of a secure annotation service utilizing Grid and Web Services-centric security technologies is discussed at some length in [13].

Our work is pitched at an intermediate level between an abstract model and a concerete implementation. In particular, the issues that we raise are couched specifically within the context of a SOA-based provenance system. In discussing these issues, we also note which ones are potentially more 'provenance-centric' and which ones resemble existing database security issues.

6 Conclusion

In this paper, we provide a representation for provenance in a SOA. We then proceed to describe some of the basic security issues pertaining to provenance in such

an environment and possible ways of addressing them. The issue of enforcing accessing control over process documentation represents a unique challenge that potentially distinguishes security considerations for accessing provenance information from that of a more generic data store. Other issues such as developing a trust framework for actors and provenance stores, establishing liability for creation of p-assertions, sensitivity of information in p-assertions as well as their long term storage considerations are more representative of conventional data security concerns.

The notion of scalability is introduced and the additional approaches required to address the new security considerations that arise as a consequence are discussed. We note that all of these security issues have not been explored in depth here; they represent possible pointers to future research on security in provenance systems which is necessary to create industrial strength systems.

Acknowledgements

This research is funded in part by the EU Provenance project as part of the European Community's Sixth Framework Program (IST511085).

References

1. R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proceedings of of the ACM SIGMOD International Conference on Management of Data*, Dallas, Texas, 2000.
2. Sergio Alvarez, Javier Vazquez-Salceda, Tamas Kifor, Laszlo Z. Varga, and Steven Willmott. Applying provenance in distributed organ transplant management. In *this volume*, 2006.
3. Miguel Branco and Luc Moreau. Enabling provenance on large scale e-science applications. In *this volume*, 2006.
4. Uri Braun and Avi Shinnar. A security model for provenance. Technical report, Harvard University, 2002.
5. Steve Burbeck. The tao of e-business services. Technical report, IBM Software Group, October 2000.
6. C. Clifton and D. Marks. Security and privacy implications of data mining. In *Proceedings of the ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 1996.
7. I. Foster, J. Voeckler, M. Wilde, and Y.Zhao. Chimera: A virtual data system for representing, querying and automating data derivation. In *Proc. of the 14th Conf. on Scientific and Statistical Database Management*, July 2002.
8. M. Greenwood, C. Goble, R. Stevens, J. Zhao, M. Addis, D. Marvin, L. Moreau, and T. Oinn. Provenance of e-science experiments - experience from bioinformatics. In Simon J Cox, editor, *Proc. UK e-Science All Hands Meeting 2003*, pages 223–226, September 2003.
9. P. Groth, M. Luck, and L. Moreau. Formalising a protocol for recording provenance in grids. In *Proc. of the UK OST e-Science second All Hands Meeting 2004 (AHM'04)*, Nottingham, UK, September 2004.

10. Paul Groth, Michael Luck, and Luc Moreau. A protocol for recording provenance in service-oriented grids. In Teruo Higashino, editor, *Proceedings of the 8th International Conference on Principles of Distributed Systems (OPODIS'04)*, volume Lecture Notes in Computer Science, pages 124–139, Grenoble, France, December 2004. Springer-Verlag.
11. Paul Groth, Simon Miles, and Steve Munroe. Principles of high quality documentation for provenance: A philosophical discussion. In *this volume*, 2006.
12. Paul Groth, Simon Miles, Victor Tan, Sheng Jiang, Steve Munroe, Sofia Tsasakou, and Luc Moreau. Architecture for provenance systems. Technical report, University of Southampton, February 2006.
13. Imran Khan, Ronald Schroeter, and Jane Hunter. Implementation of a secure annotation service. In *this volume*, 2006.
14. Tamas Kifor, Varga Laszlo, Sergio Alvarez, Javier Vazquez-Salceda, and Steven Willmott. Privacy issues of provenance in electronic healthcare record systems. In *Proc. 1st Workshop on Privacy and Security in Agent-based Collaborative Environments (PSACE 2006), 5th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2006)*, Japan, May 2006.
15. T.F. Lunt and E.D. Fernandez. Database security. *SIGMOD RECORD*, 19(4):90–97, 1990.
16. N.R.Adam and J.C.Wortmann. Security-control methods for statistical databases: A comparative study. *ACM Computing Surveys*, 21(4):515–556, December 1989.
17. M. Szomszor and L. Moreau. Recording and reasoning over data provenance in web and grid services. In *Int. Conf. on Ontologies, Databases and Applications of Semantics*, volume 2888 of *LNCS*, 2003.
18. S. R. Wiseman. On the problem of security in database. In D.L.Spooner and Landwehr, editors, *Database Security III*, pages 301–311, North Holland, 1990. Elsevier Science Publishers.
19. H.C. Wong and K.Sycara. Adding security and trust to multi-agent systems. In R.Falcone, C.Castelfranchi, Y.H. Tan, and B.Firozabadi, editors, *Workshop on Deception, Fraud and Trust in Agent Societies: Proceedings of the 3rd International Conference on Autonomous Agents*, Seattle, Washington, 1999. ACM Press.
20. J. Zhao, C. Goble, M. Greenwood, C. Wroe, and R. Stevens. Annotating, linking and browsing provenance logs for e-science. In *Proc. of the Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data*, October 2003.