Applying Provenance to Protect Attribution in Distributed Computational Scientific Experiments

Luiz M. R. Gadelha Jr.[1] and Marta Mattoso[2]

[1] National Laboratory for Scientific Computing, Brazil
lgadelha@lncc.br
[2] Federal University of Rio de Janeiro, Brazil
marta@cos.ufrj.br

Abstract. The automation of large scale computational scientific experiments can be accomplished with the use of scientific workflow management systems, which allow for the definition of their activities and data dependencies. The manual analysis of the data resulting from their execution is burdensome, due to the usually large amounts of information. Provenance systems can be used to support this task since they gather details about the design and execution of these experiments. However, provenance information disclosure can also be seen as a threat to correct attribution, if the proper security mechanisms are not in place to protect it. In this article, we address the problem of providing adequate security controls for protecting provenance information taking into account requirements that are specific to e-Science. Kairos, a provenance security architecture, is proposed to protect both prospective and retrospective provenance, in order to reduce the risk of intellectual property disputes in computational scientific experiments.

1 Introduction

Provenance allows for the precise description of how a computational scientific experiment was set up, and what happened during its execution. It also makes it easier to reproduce an experiment for the purpose of verification. New scientific results may be derived from the analysis of an experiment, which may produce valuable intellectual property. Therefore, this ease of reproducibility can also be seen as a threat to intellectual property, if the proper security mechanisms are not in place to protect provenance information. This article follows the computer security terminology used by Anderson [3]. An *entity* can be defined as a person, a computer system or an organization. *Secrecy* can be defined as the property of access to some information being limited to a number of entities. Particular cases of secrecy are *confidentiality*, when a group of entities can limit access to some information they share, and *privacy*, when an entity is able to limit access to some information it knows. *Integrity* is the property of preventing unauthorized or accidental modifications to some information. *Authenticity* is the assurance of identity of an entity in a communication. A *threat* is a possible event that may compromise the protection of a system. A *vulnerability* is property of a system that, in conjunction with a threat, may cause a system to be compromised. An *adversary* can be defined as an entity that seeks to exploit some vulnerability. One needs to identify threats, vulnerabilities and the potential damage to provenance information, and to propose mechanisms to reduce or eliminate the risk of these vulnerabilities being explored. The lack of adequate security controls may also lead to vulnerabilities that can cause provenance information being accessed without permission, or being modified intentionally or accidentally. Many scientific communities, such as the life sciences, are sensitive to security issues, so the absence of appropriate security controls may prevent wider adoption of provenance systems in production environments in these areas. Kairos is not as relevant, but still can be applied, in *Open Science*, where all the steps in a scientific experiment are publicly accessible during its execution and often open to participation. In this case, intellectual property protection is usually not a concern due to the transparency of this methodology. The main objective of this work is to address the problem of providing adequate security controls for protecting the authorship of computational scientific experiments, taking into account the requirements that are specific to e-Science. These requirements include, as we describe later in this article, being able to share scientific workflow provenance without loosing control on intellectual property. The early steps in the life cycle of a computational experiment, such as the design phase, are critical in the production of intellectual property since it is typically where the hypothesis of the experiment is defined. In previous work, we have defined Kairos [12], a security architecture for protecting the authorship of computational scientific experiments by securing retrospective provenance information. However it lacked mechanisms for protecting prospective provenance information. In this work, we improve Kairos by including such mechanisms, allowing for security controls to be applied at an earlier stage of the computational experiment, the design phase, for protecting hypothesis formulation. Applying security controls in this phase is more effective since it is less vulnerable to attacks that are typical of distributed environments used in the execution phase of the experiment. A combination of digital signatures and cryptographic timestamps [16] are used to build verifiable assertions on authorship and temporal information about the computational experiment.

This work has the following contributions: a threat model for provenance in e-Science; a new version of Kairos comprising extended security support to different phases of the experiment; an evaluation with the proposed techniques using a real application with provenance records from the Swift system [26]; and an overhead analysis of the security controls implemented in this new version of Kairos.

The remainder of this article is organized as follows. In section 2, we review related work in this subject. In section 3, we present security requirements for provenance systems in the context of e-Science, and describe a threat model for them. In section 4, we extend Kairos [12] by implementing the proposed techniques as an extension of MTCProv [15], a provenance management system for many-task computing. In section 5, we evaluate the implementation both in terms of additional storage space required and execution time. Finally, in section 6, we close with some concluding remarks.

2 Related Work

Provenance security is a relatively recent research issue [25, 17, 27, 7, 10, 22], found in different areas such as scientific workflows, databases, and storage systems. There are cases in which the subject of provenance data may lead to privacy concerns [10]. Intellectual property issues are also frequently mentioned in the literature about provenance systems [18], a clear indication that provenance information is a valuable information asset that must be protected. The most common approach for protecting provenance is to use access control mechanisms to prevent unauthorized access to this information [5, 20, 21]. This can be seen as an approach that targets the protection of confidentiality and privacy. Tan et al. [25] observed that access control is a provenance security requirement and that digital signatures can be used attribution and integrity. Hasan, Sion and Winslett [17] also target confidentiality of provenance records, they use asymmetric cryptography to achieve it and integrity is also obtained with the use of digital signatures. Dai et al. [9] presented an approach that allows for evaluating data trustworthiness from provenance information before using it as input to scientific workflows that are often time consuming. Qian et al. [22] introduce a method for building *editable signatures*, where multiple parties sign data records in a chained process to assure their trustworthiness. These approaches focus on assurance mechanisms for provenance information one gets from third parties. Our work, on the other hand, focuses on protecting provenance information that one owns and wants to share. The main contribution of this work is the evaluation of security threats to attribution in provenance systems in the context of e-Science and the proposition of security controls for protection against these threats. As far as we know, no other work provides protection of computational experiment attribution with the same flexibility as in the new version of Kairos, allowing for provenance information sharing at the same time. To our knowledge, none of these approaches found in related work propose security controls for protecting temporal information, a critical aspect in asserting attribution as we argue in section 3. Instead, most of them target controlling access to provenance information. A fundamental limitation of these approaches is that they restrict scientific collaboration. Due to concerns about correct attribution, scientists usually start sharing their experiment descriptions and data more openly only when their results are published in some academic journal or conference. By protecting the integrity and authenticity of temporal information, along with authenticity of authorship through digital signatures in different phases of a computational experiment, the approach proposed in the new version of Kairos, provenance information can be shared and disseminated earlier with less concern with respect to maintenance of correct attribution. In securing log files and audit trails [23] one is concerned with preserving integrity with the purpose of, for instance, chronologically reproducing an attack. However, differently from Kairos, verifiable assertion of the time-stamp of each event is not taken into account, which is a requirement in protecting attribution and intellectual property.

3 Security Requirements for Provenance Systems

Scientific research pursues the generation of knowledge [4], which often involves going through the steps of formulating a question, generating a hypothesis, making a prediction, performing an experiment, and analyzing its outcome. If the analysis confirms the hypothesis, one can say that new scientific knowledge was generated as a product. A computational scientific experiment follows a similar knowledge derivation process, in which provenance information supports its analysis phase. Therefore, one can say that provenance information is one the most important information assets for a scientist. In figure 1, we describe a model for provenance management systems upon which we analyze security threats. It fits the definition of provenance management system commonly found in surveys about provenance [6]. Provenance may be classified as *prospective*, when it is captured during the workflow design phase and it describes its activities and data dependencies, or as *retrospective*, when it is captured during the workflow execution phase and it describes activity executions and data artifacts generated. This information is used by the scientific workflow management system (SWMS) to plan the execution of the scientific workflow and submit its application components for execution on computational resources. A provenance management system is given by a provenance collection service, a provenance database, and a provenance access service. The provenance collection service gathers prospective and retrospective provenance information and stores it in the provenance database. In our threat model, we are assuming that provenance information is gathered at the workflow level. The provenance access service provides a browsing or querying interface to the provenance database, where users can retrieve provenance information for computational experiment analysis.

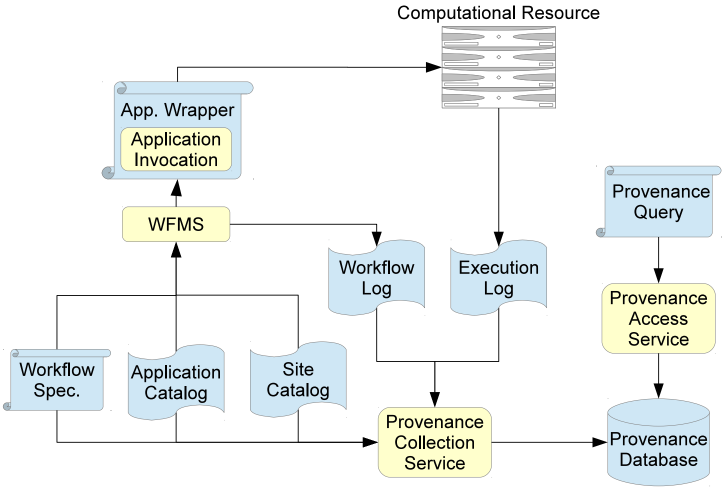

Fig. 1. Provenance management system model (modified from [15]).

Our main objective in this section is to identify threats to the confidentiality, integrity, authenticity, and availability of provenance information. As far as we know, this is the first work to identify these threats and their relationship to intellectual property protection adapted to e-Science. The methodology used follows commonly used steps for modeling threats [24]. First one needs to identify the main information assets of computational scientific experiments. Then, one needs to attribute a value to each of these assets. Next, one needs to identify existing threats to these assets and their likelyhood of materializing as attacks. For each of these threats, the potential loss in case of a successful attack needs to be evaluated as well as the cost of the respective security controls. Finally, depending on the relation between potential loss and cost of security control, one needs to decide whether to accept a risk or to establish protective mechanisms. This risk analysis procedure should be periodically repeated for refinement and for taking emerging threats into consideration. In e-Science, experiments are performed using computational models to simulate phenomena. Therefore all artifacts involved in applying the scientific method *in silico* can be considered as important information assets. This includes its input and output data sets, and all related provenance information. Next, we explore scenarios that illustrate threats to these assets.

{**S1**} **Illegitimate claim of attribution**. An adversary is able to intercept network communication between the site from which a scientist submits his or her scientific workflow for execution and the site hosting computational resources that will execute its component applications. If the adversary is able to retrieve retrospective provenance records he/she might be able to infer both the intent and results of the computational scientific experiment. The adversary might obtain the same information if he/she is able, for instance, to obtain privileged administrative rights either in the remote computational resources. Using the provenance records, the adversary might be able reproduce the computational experiment. With knowledge of the intent and the results of the computational experiment, and by having reproduced it, the adversary can eventually claim its attribution before the original author.

{**S2**} **Unauthorized access to private data**. If proper access control is not in place in the provenance database, or if network communication is not secured, an adversary might have access to private data manipulated by a scientific computational experiment. For instance, patient data in biomedical workflows.This might legal action because of adequate security controls not being used while manipulating private data.

{**S3**} **Intentional modification of provenance records**. An adversary could modify provenance records to mislead the scientist during the analysis of a computational scientific experiment. As a consequence, experiments that had a relevant outcome might be disregarded. The opposite situation is also possible, where one might spend time in experiments that did not produce valuable results.

{**S4**} **Dissemination of illegitimate provenance data**. An adversary disseminates forged provenance records, for instance, by feeding the provenance collection service with illegitimate data. If scientists are not able to infer the trustworthiness of provenance information, they might reuse this forged provenance data, for instance, in scientific workflow re-executions. This can induce scientists to spend their time in computational scientific experiments that will likely lead to irrelevant or incorrect results.

{**S5**} **Obstruction of provenance information collection and access**. An adversary might generate a large number of requests to either the provenance collection service or the provenance access service beyond their processing capacity, turning the provenance management system unavailable to legitimate users. This would delay the upload of provenance data from computational scientific experiments that could be under execution. Consequently, the analysis phase of the experiment would be hindered during this type of attack due to the unavailability of supporting provenance information.

Provenance records are analogous to laboratory notebooks from traditional scientific experiments. They record the plan of an experiment, its initial parameters, and its outcome. Many scientific institutions maintain guidelines [1] for protecting any resulting intellectual property, observing that the laboratory notebook is one of the most important elements in the process of applying for a patent, where one should prove that the work that lead to some result was performed before the work of anyone else that could claim the same result. One of the guidelines is that notes should be signed and dated and not modified afterwards. In the same manner, provenance records about computational scientific experiments should be protected with appropriate security controls that enable one to assert by whom and when an experiment was performed, and that its provenance records were not modified afterwards. Therefore, to prevent scenario {S1} from happening, security controls that prevent illegitimate claims of attribution are an important security requirement for provenance systems. A combination of digital signatures and cryptographic timestamps [16] were used in the Kairos [12] security architecture for provenance systems to protect retrospective provenance, which we extend in this work to also cover prospective provenance. Since digital signatures also protects the integrity of provenance records, Kairos also prevents scenario {S3} from happening. Preventing scenario {S2} is a concern when personal data is manipulated [10], which is not the predominant case in e-Science. Also, personal data manipulated by scientific workflows is not as important to the scientist as an asset as information that leads to knowledge generation, which is the primary goal in scientific research. Therefore, scenario {S1} has a higher potential damage than scenario {S2}. Both scenarios {S4} and {S5} may lead a scientist to loose significant time by either being unable to access provenance information required for experiment analysis or by consuming data that might not be valid, leading to incorrect results [9, 22]. In both of these situations, scientists are often able to detect and correct the problems by either blocking the source of attack and re-establishing availability or by identifying and discarding untrustworthy data sources. Hence, we see both scenarios {S4} and {S5} as less threatening than scenario {S1}, placing the protection of attribution of computational scientific experiments as a security requirement that should be given high priority. In the next section, we present security controls for preventing this particular scenario.

4 Protecting Attribution in Distributed Scientific Workflows

In order to protect intellectual property, Kairos provides tools given by the combined use of digital signatures and the TSP [16] to securely determine the author of provenance assertions and the date in which they were created. The secure time-stamping process involves computing a hash value of the provenance record, which is sent to the Time-Stamping Authority [16] (TSA). The TSA appends to the hash the current date, obtained from a trustworthy source of time. This pair is digitally signed, which requires access to the private key of the TSA, resulting in a *time-stamp receipt*. The time-stamp receipt is sent to the user and can be used to prove the date of creation of the provenance record. This can be done by verifying the date contained in the digital receipt and the digital signature of the TSA, which requires access to the public key of the TSA. We use the notation $\mathsf{Sign}(\langle\text{object}\rangle, \langle\text{credential}\rangle)$ to indicate the resulting object of a digital signature operation over object ⟨object⟩ using credential ⟨credential⟩, which consists of computing the hash value of ⟨object⟩ and encrypting it with the private key associated to $\langle\text{credential}\rangle$; and $\mathsf{TSP}(\langle\text{object}\rangle)$ to indicate the digital receipt that results from applying the TSP to object $\langle\text{object}\rangle$ which results in a time-stamp receipt, as described in section 1. To also prove authorship of a provenance record, we add a digital signature performed by the scientist. This allows for the verification of both authorship and date of creation of the provenance record. This was proposed in our previous work for protecting retrospective provenance records [12]. However, this process was still susceptible to attacks since retrospective provenance records are usually generated on remote computational resources during the execution of component activities of a scientific workflow. These records can still be vulnerable to network or privileged user attacks from the time they are generated on remote computational resources to the time one applies the security techniques described. Our approach for mitigating this threat consists of extending Kairos to also protect prospective provenance, which is usually generated at the beginning of the computational scientific experiment life cycle, before anything is sent to remote computational resources. The procedure $\mathsf{Sign\text{-}and\text{-}Time\text{-}stamp}(\mathcal{P},\mathcal{C})$, for digitally signing and time-stamping a provenance trace $\mathcal{P}$ using a credential $\mathcal{C}$ and a TSA, consists of computing $\mathcal{S} = \mathsf{Sign}(\mathcal{P},\mathcal{C})$; and then computing $\mathcal{T} = \mathsf{TSP}(\mathcal{S})$. Finally $\mathcal{S}$ and $\mathcal{T}$ are stored in the provenance database.

In table 1, we present the Kairos protocol, for applying the $\mathsf{Sign\text{-}and\text{-}Time\text{-}stamp}$ to both the prospective and retrospective provenance traces of a scientific workflow execution run_i, denoted by $\mathcal{P}_{\text{prospective}}(\text{run}_i)$ and $\mathcal{P}_{\text{retrospective}}(\text{run}_i)$ respectively. The same protocol is illustrated in figure 2 using corresponding steps. The pair of objects produced by the protocol in the $\mathsf{Sign\text{-}and\text{-}Time\text{-}stamp}$ steps will be called a *secure provenance receipt*, or $\mathcal{SP}$-receipt.

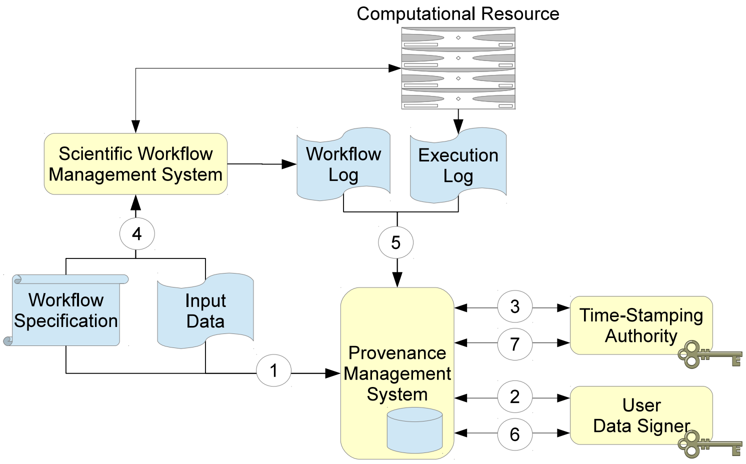

Fig. 2. Kairos: procedure overview.

An auditor can verify the $\mathcal{SP}$-receipt produced by the protocol using the public keys of both the user and the TSA. To verify the time-stamp receipt one needs to apply

Table 1. Description of the Kairos protocol.

Step 1. Store $\mathcal{P}_{\text{prospective}}(\text{run}_i)$ in the provenance database;

Steps 2 and 3. Sign-and-Time-stamp($\mathcal{P}_{\text{prospective}}(\text{run}_i), \mathcal{C}$);

Step 4. Execute scientific workflow;

Step 5. Store $\mathcal{P}_{\text{retrospective}}(\text{run}_i)$ in the provenance database;

Steps 6 and 7. Sign-and-Time-stamp($\mathcal{P}_{\text{retrospective}}(\text{run}_i), \mathcal{C}$);

the encryption function to the digital signature performed by the TSA using its public key and compare the result with the hash value of the concatenation of the time-stamp and the object produced by the digital signature performed by the user. If they match, the time-stamp receipt is valid. To verify the digital signature performed by the user, the process is analogous and uses the public key of the user instead. A complete validation would also verify the digital signatures in the digital certificates used in the process. These certificates usually form a chain and the validation is completed when one reaches a trusted certificate authority.

Next, we discuss how the protocol can support the preservation of the correct attribution of a computational experiment. Suppose a user is the first one to run a scientific workflow on a remote computational resource, with some specification and input data sets, and follows the Kairos security protocol. Therefore, he or she gets an $\mathcal{SP}$-receipt as result. Now suppose an adversary was able to compromise a computational resource and obtain both the prospective and retrospective provenance traces. If the adversary tries to claim the authorship of the computational scientific experiment, the user can challenge him or her to present an $\mathcal{SP}$-receipt with an earlier time-stamp for prospective provenance. Since the prospective provenance trace was generated and time-stamped before submitting the scientific workflow for execution to the computational resource, the adversary would only be able to access this trace if the submitting host was also compromised. This is less likely to happen since the submitting host is often not shared with other users. Therefore, it is unlikely that the adversary would be able to forge an $\mathcal{SP}$-receipt containing an earlier time-stamp that the one contained in $\mathcal{SP}$-receipt of the user. The portion of the $\mathcal{SP}$-receipt related to the retrospective provenance trace can be useful, for instance, when claiming a patent based on the outcome of the computational experiment, since detailed description of experiment execution and its respective temporal information are critical steps in this process.

The cryptographic data stored in MTCProv by Kairos enables queries involving security aspects of provenance to be answered. Given a dataset produced by a scientific workflow, one can securely determine all the individuals that were involved in the production of a particular scientific dataset. Such query can be answered, for instance, by traversing the provenance graph recursively to determine ancestral processes and datasets, and gathering respective name-value annotations containing digital signatures. One can also, given several provenance traces describing the generation of the same scientific dataset, securely verify which one was the earliest. This can be done, for instance, by retrieving name-value annotations associated to the respective executions containing time-stamping receipts and selecting the earliest one. One important aspect of the answers to these queries is that they are verifiable with cryptographic techniques if one has access to the respective public keys of either the TSA or the author of a digital signature. With these tools, one can more easily assert the *what*, *who*, and *when* of a computational scientific experiment, essential in any patenting process.

5 Implementation and Evaluation

The experiments with the proposed protocol are based on Swift [26], a parallel scripting system that allows for managing many-task scientific workflows. Swift generates provenance traces in its log files, and this information can be exported to a relational database using a data model [15] similar to PROV [19]. Therefore, the techniques presented in this work are also applicable to provenance information represented using these standards. MTCProv [15] is the provenance management component of Swift. It has a query interface with built-in procedures that supports commonly used provenance queries [13, 14]. We implemented a prototype of Kairos in the Python programming language as a wrapper that interacts with cryptographic functions of the OpenSSL library [2], Swift and MTCProv. The implementation uses cryptographic functions of the OpenSSL cryptographic toolkit: the `smime` function can be used for the digital signatures and the `ts` function can be used to both execute the TSP and to implement a TSA. The digital signatures and time-stamp receipts generated by the prototype described above are stored as name-value pair annotations associated to the respective scientific workflow execution in the provenance database.

To evaluate the impact of Kairos, we used a a ray-tracing workflow, `c-ray.swift`, that generates a number of scene definitions, invokes a ray-tracing application to render them, and converts the resulting image frames into a video. For each number of iterations, five executions were performed for gathering the storage space and execution time statistics. The evaluation was performed in an environment consisting of a submission host with a six-core Intel Xeon E7540 processor, where Swift was executed, and a remote multi-processed host with two 12-core AMD Opteron 6238 processors, where the computationally demanding application components of the workflow were executed. To scale the execution of the workflow, Swift is able to execute multiple ray-tracing tasks in parallel in this remote multi-processed host using the SSH execution provider. The TSA was installed in the submission host and we included a pause with a random duration between 100 and 400 milliseconds before submitting each time-staming request, in order to simulate the cost of communication with a remote TSA. In Figure 3, we plot the extra amount of storage space and execution time required by Kairos.

Since the time-stamp receipt is computed from a hash value, which has a fixed size, it will also have approximately fixed size, apart from minor variations due to padding. The size of digital signature performed with the `smime` tool, grows very slowly when compared to the size of the provenance trace. Therefore, as one can observe in figure 3, the size of digital signatures and time-stamp receipts becomes proportionally smaller as the size of the workflow grows. As mentioned in the previous section, the current prototype also stores the original objects that were signed, in addition to the digital

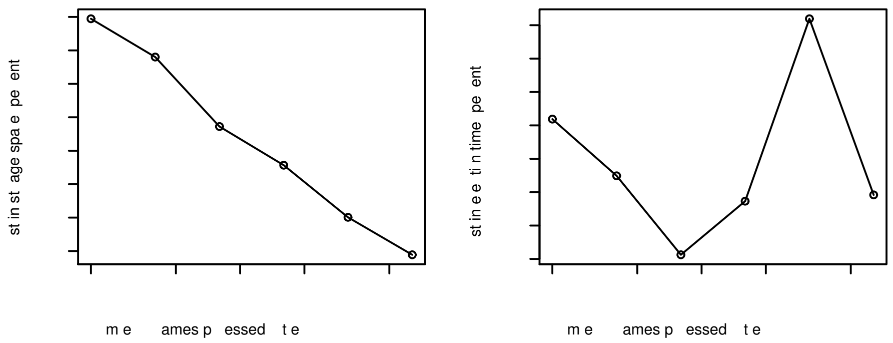

Fig. 3. Impact of Kairos in terms of storage space and execution time.

signatures, with the purpose of enabling signature verification. To preserve the validity of an $\mathcal{SP}$-receipt, the respective original provenance trace should not be modified. The management of cryptographic data could be improved in Kairos by using cryptographic standards that have better support for managing both digital signatures and time-stamp receipts, such as XAdES [8]. The time to execute both the timestamp protocol and the digital signature procedure depend on the size of the provenance trace, since a hash value needs to be computed from its content. For the 64-step execution of the workflow, the retrospective trace has about 803KB in size and it takes about 35 milliseconds to digitally sign and time-stamp it on the submission host. Therefore, one can observe in figure 3 that the impact in terms of execution time is smaller than other factors, such as the scheduling heuristics used by the execution provider and the staging-in and staging-out of files between the submission and the multi-processed host. In the current implementation of Kairos, the granularity used for applying the security controls is at the provenance trace level. One could alternatively use a finer-grained granularity at the provenance assertion level, however the impact in terms of space would be considerably higher. Consider, for instance, the 64-step execution of the workflow. It is given by 6403 provenance assertions, with an average size of 124 bytes. The cryptographic data associated to the digital signatures and time-stamping receipts has an average size of 3.3KB per assertion, since it must contain also information about the credentials used. This results in 20.6MB of cryptographic data in comparison to the total provenance trace size of 803KB, a 26-fold increase. However, the need of fine-grained protection is diminished by the application of the security controls to the prospective provenance information prior to the execution of the scientific workflow.

6 Conclusion

In this work, we survey and analyze security requirements for provenance management systems. We propose that the main information asset of these systems is given by provenance traces describing the intellectual process of a computational scientific experiment, which require appropriate security controls for protection. This information is particularly vulnerable in current e-Science infrastructures since they often are transferred to third-party computational resources which scientists have little control of. Therefore, we have extended Kairos [12], which secures the authorship and temporal information of computational scientific experiments, to also protect prospective provenance and implemented it as part of MTCProv [15], a provenance management system for many-task computing. We describe useful queries that can be answered by MTCProv using the information generated by these security controls and stored in its relational database. The security controls implemented are essential to any claim of intellectual property, where individuals need to present evidence that they were the first ones to obtain some scientific result. The improvements implemented in Kairos, relative to the version presented in [12], allow for better protection of correct authorship attribution since it applies the proposed security controls also to prospective provenance information at the design phase of the computational scientific experiment life cycle. The hypothesis of the experiment is typically defined at this stage, which makes it critical in applying security controls for protecting intellectual property. At this stage the information is much less exposed to attacks commonly found in remote and distributed computational resources, where retrospective provenance is gathered. We also presented an evaluation of the impact of the proposed techniques in terms of storage space required and execution time, concluding that it is relatively small when they are applied at the provenance trace level of granularity. As in GSI [11], the security controls used in Kairos are based on common public key infrastructure techniques, where certificate authorities are trusted to digitally sign and publish, in the form of digital certificates, public keys associated to users. Therefore, Kairos should be relatively straightforward to integrate to existing grid computing infrastructures, where many large scale computational scientific experiments are performed.

Acknowledgment

This work is partially funded by CNPq and FAPERJ.

The original publication is available at www.springerlink.com:
Gadelha, L., Mattoso, M. (2015). Applying Provenance to Protect Attribution in Distributed Computational Scientific Experiments. In B. Ludäscher & B. Plale (Eds.), Provenance and Annotation of Data and Processes - 5th International Provenance and Annotation Workshop, IPAW 2014. Lecture Notes in Computer Science 8628 (pp. 139-151). Cologne, Germany: Springer.
`http://link.springer.com/chapter/10.1007/978-3-319-16462-5_11`

References

1. *Guidelines for Maintaining a Lab Notebook*. Los Alamos National Laboratory, 2014.
2. OpenSSL, http://www.openssl.org, 2014.
3. R. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley, 2nd edition, 2008.
4. W. C. Booth, G. G. Colomb, and J. M. Williams. *The Craft of Research, Third Edition*. University Of Chicago Press, third edition edition, Apr. 2008.
5. U. Braun, A. Shinnar, and M. Seltzer. Securing provenance. In *Proceedings of the 3rd conference on Hot topics in security*, page 4:1–4:5, Berkeley, CA, USA, 2008. USENIX.
6. L. Carata, S. Akoush, N. Balakrishnan, T. Bytheway, R. Sohan, M. Selter, and A. Hopper. A primer on provenance. *Communications of the ACM*, 57(5):52–60, May 2014.
7. A. Chebotko, S. Lu, S. Chang, F. Fotouhi, and P. Yang. Secure abstraction views for scientific workflow provenance querying. *IEEE Trans. on Services Computing*, 3(4):322–337, 2010.
8. J. Cruellas, G. Karlinger, D. Pinkas, and J. Ross. XML advanced electronic signatures (XAdES), http://www.w3.org/tr/xades, 2003.
9. C. Dai, D. Lin, E. Bertino, and M. Kantarcioglu. An approach to evaluate data trustworthiness based on data provenance. In *Proc. 5th VLDB Workshop on Secure Data Management (SDM 2008)*, volume 5159, pages 98, Springer, 2008.
10. S. B. Davidson, S. Khanna, T. Milo, D. Panigrahi, and S. Roy. Provenance views for module privacy. In *Proc. ACM PODS '11*, page 175–186. ACM, 2011.
11. I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke. A security architecture for computational grids. In *Proc. ACM CCS '98*, CCS '98, page 83–92, New York, NY, USA, 1998. ACM.
12. L. Gadelha and M. Mattoso. Kairos: An architecture for securing authorship and temporal information of provenance data in grid-enabled workflow management systems. In *IEEE Fourth International Conference on eScience (e-Science 2008)*, pages 597–602. IEEE, 2008.
13. L. Gadelha, M. Mattoso, M. Wilde, and I. Foster. Provenance query patterns for many-task scientific computing. In *Proceedings of the 3rd USENIX Workshop on Theory and Applications of Provenance (TaPP'11)*, 2011.
14. L. Gadelha, M. Wilde, M. Mattoso, and I. Foster. Exploring provenance in high performance scientific computing. In *Proceedings of the First Annual Workshop on High Performance Computing Meets Databases*, HPCDB '11, page 17–20, New York, NY, USA, 2011. ACM.
15. L. Gadelha, M. Wilde, M. Mattoso, and I. Foster. MTCProv: a practical provenance query framework for many-task scientific computing. *Distributed and Parallel Databases*, 30(5-6):351–370, 2012.
16. S. Haber and W. Stornetta. How to time-stamp a digital document. *Journal of Cryptology*, 3(2):99–111, 1991.
17. R. Hasan, R. Sion, and M. Winslett. Preventing history forgery with secure provenance. *ACM Transactions on Storage*, 5(4):12:1–12:43, Dec. 2009.
18. S. Miles, P. Groth, M. Branco, and L. Moreau. The requirements of recording and using provenance in e-science. *Journal of Grid Computing*, 5(1):1–25, 2007.
19. L. Moreau and P. Groth. Provenance: An introduction to PROV. *Synthesis Lectures on the Semantic Web: Theory and Technology*, 3(4):1–129, Sept. 2013.
20. M. Nagappan and M. Vouk. A model for sharing of confidential provenance information in a query based system. In *Proc. 2nd International Provenance and Annotation Workshop (IPAW 2008)*, volume 5272, pages 69, Springer, 2008.
21. Q. Ni, S. Xu, E. Bertino, R. Sandhu, and W. Han. An access control language for a general provenance model. In *Secure Data Management*, volume 5776, pages 68–88. Springer, 2009.
22. H. Qian and S. Xu. Non-interactive editable signatures for assured data provenance. In *Proc. ACM CODASPY '11*, page 145–156, New York, NY, USA, 2011. ACM.

23. B. Schneier and J. Kelsey. Secure audit logs to support computer forensics. *ACM Transactions on Information and System Security*, 2(2):159–176, May 1999.
24. F. Swiderski and W. Snyder. *Threat Modeling*. Microsoft Press, 2004.
25. V. Tan, P. Groth, S. Miles, S. Jiang, S. Munroe, S. Tsasakou, and L. Moreau. Security issues in a SOA-based provenance system. In *Provenance and Annotation of Data*, pages 203–211. Springer, 2006.
26. M. Wilde, M. Hategan, J. M. Wozniak, B. Clifford, D. S. Katz, and I. Foster. Swift: A language for distributed parallel scripting. *Parallel Computing*, 37(9):633–652, Sept. 2011.
27. S. Xu, Q. Ni, E. Bertino, and R. Sandhu. A characterization of the problem of secure provenance management. In *Proc. IEEE International Conference on Intelligence and Security Informatics (ISI 2009)*, page 314, 2009.