Securing Grid Workflows with Trusted Computing

Po-Wah Yau, Allan Tomlinson, Shane Balfe and Eimear Gallery.

Information Security Group
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
{p.yau, allan.tomlinson, s.balfe, e.m.gallery}@rhul.ac.uk

Abstract. We propose a novel scheme that uses Trusted Computing technology to secure Grid workflows. This scheme allows the selection of trustworthy resource providers based on their platform states. The integrity and confidentiality of workflow jobs are provided using cryptographic keys that can only be accessed when resource provider platforms are in trustworthy states. In addition, platform attestation is used to detect potential workflow execution problems, and the information collected can be used for process provenance.

1 Introduction

Grid computing [7] is a distributed computing paradigm which seeks to exploit the synergies of technology and social collaboration to solve data or computation-intensive problems. Solving such problems requires the management of multiple tasks, their relationships and execution to produce valid and reliable results — this is the focus of Grid workflow research [23]. Although the authorisation and authentication of Grid jobs has been extensively studied [2, 8], the management of Grid workflows introduces additional security issues. The risk to a user's data and results is dramatically increased when using workflows, because the entire dataset is exposed to the Grid.

The use of historical information has been proposed to schedule jobs so that 'untrusted' nodes can be avoided [22], but such information may be incorrect or open to manipulation. Moreover, this approach cannot prevent, detect, or react to single job manipulation, the effects of which would propagate throughout the associated workflow. The provision of a data provenance service, which records how data has been collected and processed, [19, 21], helps to address this problem, but only forms part of a reactive solution to detect problems after running a workflow.

The application of Trusted Computing (TC) technology is emerging as a potential solution to a number of Grid security problems [14, 15] and this paper investigates how this technology may be applied to secure Grid workflows. Section 2 provides a brief summary of Grid workflows, and outlines a set of security requirements. Section 3 gives an overview of TC and its application to Grid security. Section 4 describes our proposal for securing Grid workflows, and section 5 contains an analysis. Final remarks are given in section 6.

2 Grid Workflows

A workflow defines a logical ordering of tasks to be completed, and can be represented as a directed acyclic graph or flowchart with parallel, sequential and choice branches and loops [25]. Each workflow task can operate on either a new set of data or the results from a parent task, i.e. intermediate data. Thus, data is input from storage resources that may be either internal or external to where the computation is taking place.

Typically, abstract workflow specifications are passed to a Workflow Resource Broker (WRB), i.e. a workflow execution engine/system, which maps workflow tasks onto physical jobs that will be submitted to a Grid. The creation of a physical workflow of Grid jobs requires the WRB to select Grid resource providers and schedule jobs to be submitted to them. The system will select resource providers that meet static and dynamic workflow requirements, for example the availability of software applications and libraries or, indeed, specific security policies.

During workflow execution, data can be moved using one of three approaches — centralised, mediated or peer-to-peer [25]. Centrally managed data movement is the easiest to implement, as all data is transferred via a central point. Mediated data movement involves a distributed management system with synchronised replication catalogue services. Finally, using a peer-to-peer method involves transferring data directly between resource providers.

Workflow Security: Trust in the WRB is critical, as it is relied upon by a user to ensure that their workflow will be executed as expected, and thus produce valid results. A user delegates control to a WRB to map workflow tasks to jobs, which must then be submitted to the appropriate resource providers. The WRB is also trusted not to divulge workflow information that would allow an attacker to coordinate attacks on the workflow. The compromise of a single job might not reveal any sensitive information, whereas an attack on several jobs might. Therefore, it is essential to maintain confidentiality of the locations to which workflow jobs are submitted.

Resource providers might be selected based on direct experience and/or other indirect metrics, such as reputation or trust measurements based on provenance services [19, 22]. However, there is a risk that this information is unreliable, incorrect or out-of-date. Thus, a WRB needs to be able to reliably determine if it can trust a resource provider to behave as expected before sending it a workflow job.

Many observers have commented on the vulnerabilities surrounding Grid middleware and the subsequent risk of job execution compromise [4, 15]. Therefore, it is also necessary for the assurances determined during the selection process to hold true until job execution has finished. This includes the protected transport of output data to the required destination, be it the WRB or another resource provider. If the integrity of the job execution platform is not maintained, then the WRB must be alerted to the potential compromise so that it can react accordingly. An undetected compromise could mean that resources are wasted on executing the rest of the workflow using incorrect data.

Finally, audit trail information must be reliably collected. As stated above, provenance information, or the procedure for collecting provenance information itself, could be flawed and a mechanism is required to detect when this is the case. Audit informa-

tion will also assist the debugging of workflows, as confidence in the resource providers will help to eliminate a large potential source of errors.

3 Trusted Computing

A trusted platform is one that behaves in a particular manner for a specific purpose. Such a platform can be built following the TCG's Trusted Platform Module (TPM) specifications[1]. These specifications describe a tamper-resistent device with cryptographic coprocessor capabilities. This device provides the host platform with a number of services including: special purpose registers for recording platform state; a means of reporting this state to remote entities; and asymmetric key generation, encryption and digital signature capabilities. TC also encompasses new processor designs [10] and OS support [18] which facilitate software isolation. These concepts are examined in more detail elsewhere — see, for example [16, 17]. For the purposes of this paper we examine four TC-related concepts: integrity measurement, TPM keys, sealing and platform attestation.

Integrity Measurement: An integrity measurement is the cryptographic hash of a platform component (i.e. a piece of software executing on the platform) [18]. For example, the integrity measurement of a program can be calculated by computing a cryptographic digest of a program's instruction sequence, its initial state and its input. Integrity measurements are stored in special purpose registers within the TPM called Platform Configuration Registers (PCRs).

TPM Keys: A TPM can generate an unlimited number of asymmetric key pairs. For each of these pairs, private key use and mobility can be constrained. Key use can be made contingent upon the presence of a predefined platform state (as reflected in the host platform's TPM PCRs). Additionally, a private key can be migratable, nonmigratable or certifiable migratable.

A non-migratable key is inextricably bound to a single TPM instance, and is known only to the TPM that created it. A certificate for a non-migratable key and its security properties may be created by the TPM on which it was generated. A certifiable migratable key (CMK) can be migrated but also retains properties which can be certified by the TPM on which the CMK was generated. When a CMK is created, control of its migration is delegated to a migration (selection) authority. In this way, controlled migration of the key is made possible, so that an entity other than the TPM owner helps to decide where the CMK can be migrated. This ensures that the certified security properties of the key are maintained.

Sealing: This is the process by which data is encrypted and associated with a set of integrity measurements representing a particular platform configuration. The protected data can only be decrypted and released for use by a TPM when the current state of the platform matches the integrity measurements to which the data was sealed.

---

[1] https://www.trustedcomputinggroup.org/specs/TPM/

Platform Attestation: Platform attestation enables a TPM to reliably report information about the current state of the host platform. On request from a challenger, a TPM provides signed[2] integrity measurements reflecting (all or part of) the platform's software environment. The challenger can use this information to determine whether it is safe to trust the platform and its software environment. This involves validating the received integrity measurements against a set of values it believes to be trustworthy, possibly provided by a trusted third party such as a software vendor.

However, there are potential issues surrounding the binary representation of software components — such a representation is static and inflexible; program behaviour has to be inferred; upgrades and patches are difficult to deal with; and revocation is problematic [9]. In order to overcome these problems, the concept of property-based platform state representation has been proposed [9, 20], in which a platform's state is represented by a set of high-level security properties. Using such techniques, migratable and certifiable migratable keys can be generated such that private key use is bound to properties, data can be sealed to properties, and the TPM can attest to platform properties, rather than specific software integrity measurements.

Application of TC to Grid Security: A number of authors have considered how Trusted computing could be applied to Grid Computing [4, 5, 15, 24]; the main goal of much of this prior art is to prevent or detect resource provider misbehaviour. Mao et al. [3] propose Daonity, a system which establishes a relocatable key enabling controlled group sharing of encrypted content. Löhr et al. [13] propose a scheme in which resource providers publish *attestation tokens*, which contain public keys from non-migratable TPM key pairs and the platform states to which private key use is bound. Each token is signed by the TPM to prove that it was produced by an authentic TPM.

4 Securing Grid Workflows

We now describe how Trusted Computing may be used to provide the following security services to Grid workflows:

1. Trusted Resource Provider Selection;
2. Confidentiality of job information;
3. Integrity of job information; and
4. Audit data for process provenance.

Job information can include a job script, any executables, and input and output data. TC can be used to provide strong assurances to the Grid user that a workflow has executed correctly, and that the data was protected from malicious entities.

4.1 Assumptions

In order to fully utilise TC in Grid computing, the supporting TC architecture must be integrated into Grid environments. The proposal in this paper operates on the following assumptions:

---

[2] Using a private attestation signing key.

Trusted Computing prevalence: There exists a Workflow Resource Broker (WRB) that is equipped with a trusted platform, as described in section 3. A subset of Grid resource providers will also have trusted platforms installed; the scheme only uses such providers to process workflow jobs.

Resource broker verification service: This is provided by a trusted third party, and will be used to determine whether or not a WRB is trustworthy. This is achieved by verifying the platform state attested to by a WRB against known trusted states.

Public keys: All entities involved will have a certified copy of the chosen WRB's public signature verification key. Conversely, the WRB will have the public signature verification keys of all entities.

The underlying assumption is that trusted platforms exist within a Grid network, supported by the TC infrastructure. With major backing from hardware and software vendors, TC is becoming more pervasive, which will lead to the greater availability of TC supporting entities such as migration authorities.

4.2   The Scheme

A user relies on a trusted resource broker verification service to determine the trustworthiness of a WRB, using platform attestation (see section 3). A workflow specification tool is used to create an abstract workflow of Grid tasks that is passed to the WRB, together with an encompassing security policy. The WRB maps the workflow tasks to a set of jobs, that are scheduled for submission to selected resource providers meeting the user's security requirements. To achieve this, the WRB may have to translate high-level user requirements into low-level platform state requirements. Workflow execution is then protected using TC services, as we next describe.

Key Distribution   Consider a sequence of jobs $a_0, a_1, ..., a_n$ that make up a user's workflow. For each job a_i, the WRB matches the user's high-level security requirements to a private key SK_i, whose use is contingent on the selected resource provider's platform satisfying low-level state information α (see section 3). A resource provider could use either of the following two methods to obtain a private key in our framework:

1. The private keys can be created *a priori* or dynamically by the WRB as certifiable migratable keys for each of the jobs in the user's workflow, with the WRB specifying itself as the migration (selection) authority. The WRB specifies the states to which the private keys are bound prior to their migration to the selected resource providers.
2. The resource providers themselves each create a non-migratable private key bound to a specific platform state; this state and the corresponding public key are advertised as part of an attestation token [12]. The WRB pulls the attestation tokens from a service register and uses them to select appropriate resource providers.

The result is that the WRB can seal data that a resource provider can only access when it is in a trusted state. This allows the workflow to be protected, as described below.

Protecting the Workflow Once the private keys have been provisioned, the WRB creates a symmetric key K_i for each job a_i, and generates a set of information to send to each chosen resource provider RP_i:

$$\text{WRB} \rightarrow \text{RP}_i : ID_W||r_i||g_{K_i}(a_i||r_i)||e_{PK_i}(K_i)||IP_{i+1}||PK_{i+1}||$$
$$ID_{RP_{i-1}}||VK_{RP_{i-1}}||\alpha_{i-1}||\sigma \tag{1}$$

where:

- ID_W contains the identifiers of the workflow and the WRB;
- r_i is a random nonce chosen by the WRB;
- g is the generation-encryption function of an agreed authenticated encryption scheme [6, 11] — $g_{k_i}(a_i||r_i)$ generates the ciphertext and message authentication code for the concatenation of the job and nonce;
- $e_{PK_i}(K_i)$ is the key K_i encrypted using RP_i's public key PK_i;
- IP_{i+1} is the address to which any job output should be sent — this could be either the WRB or the next resource provider in the workflow RP_{i+1}, either for storage or further processing;
- PK_{i+1} is the public key used to encrypt job output;
- $ID_{RP_{i-1}}$ is the identifier of the preceding resource broker;
- VK_{i-1} is the public verification key used to verify messages from RP_{i-1} (see section 4.1);
- α_{i-1} is the platform state that RP_{i-1} had to be in in order to process a_{i-1};
- σ is the digital signature of the WRB on the entire message.

Note that in the case of RP_0, RP_{i-1} would be the WRB. Thus, the state α_{i-1} sent to RP_0 would be the platform state of the WRB; this information can be used for auditing purposes (see below).

Executing the workflow The following is the process of workflow execution at an arbitrary RP_i, after receiving message 1 (see section 4.2). We assume that each message also contains both the identifier of its originator and a digital signature.

$$\text{RP}_{i-1} \rightarrow \text{RP}_i : ID_W||\text{ready} \tag{2}$$
$$\text{RP}_i \rightarrow \text{RP}_{i-1} : ID_W||C(r_{RP_i}) \tag{3}$$
$$\text{RP}_{i-1} \rightarrow \text{RP}_i : ID_W||\alpha_{i-1}(r_{RP_i})||g_{K_i'}(R(a_{i-1}))||e_{PK_i}(K_i') \tag{4}$$
$$\text{RP}_i \rightarrow \text{WRB} : ID_W||r_{RP_i}||\alpha_{i-1}(r_{RP_i}) \tag{5}$$

For the above interaction, the following are the steps taken by RP_i to execute a_i upon receiving message 2:

1. Verify σ from message 1.
2. Use the private key SK_i to decrypt the symmetric key K_i.

3. K_i is passed to the appropriate Grid application, which decrypts a_i and verifies its data integrity.
4. Generate a random nonce r_{RP_i} and send an attestation challenge $C(r_{RP_i})$ to RP_{i-1} (message 3);
5. Compare the response $\alpha_{i-1}(r_{RP_i})$ from message 4 with α_{i-1} from message 1;
6. The results of the comparison are sent to the WRB for auditing (see message 5). If the check has failed, then RP_i waits for further instructions from WRB, which raises an exception.
7. Otherwise, the symmetric key K_i' from message 4 is decrypted and used to recover the results $R(a_{i-1})$ of the previous job. K_i' would have been generated by RP_{i-1} (see step 9).
8. Job a_i is processed using $R(a_{i-1})$.
9. Once a_i has completed, RP_i creates a fresh symmetric key K_{i+1}', generates $g_{K_{i+1}'}(R(a_i))$ and encrypts the key $e_{PK_{i+1}}(K_{i+1}')$.

This process iterates through the workflow until the very last job is executed, when the last RP_n will respond to an attestation challenge by the WRB. Note that, in a centralised workflow system, message 2 could be sent by the WRB.

5 Security Analysis

Establishing trust in the WRB is a fundamental precursor to our scheme. It cannot be expected that a standard Grid user will be able to interpret attestation integrity measurements, hence we require a trusted third party to perform this task on behalf of the user. From this, we have a basis for determining if the results of a workflow can be secured and, indeed, trusted.

Part of this trust is formed from assurances that workflow jobs were executed correctly and not compromised in any way. This requires protection in two directions. In the forward direction, it is necessary to ensure that only trusted resource providers are selected to process workflow jobs. These jobs, together with input and output data, should have confidentiality and integrity protection so that only authorised resource providers can process them. In the reverse direction it is essential to determine whether or not the selected resource providers were compromised when processing their allocated jobs. The rest of this analysis focuses upon how well the proposed scheme provides these security services, with references back to the messages and steps described in sections 4.2 and 4.2.

Trusted Resource Provider Selection: Job protection is achieved using private keys that have been sealed to particular platform states that match the user's security requirements. During job scheduling, the WRB only considers resource providers that can provide trusted platforms in the required states (message 1). This means that if a resource provider deviates from the predefined state, either by accident or due to malicious attack, then that resource provider will be unable to access the private key to decrypt job information (step 2), and intermediate (input) data encrypted by preceding resource providers (step 7).

Confidentiality and Integrity of Job Information: Job information and workflow results are protected using authenticated encryption which provides both confidentiality and integrity services (messages 1 and 4). This requires the use of symmetric keys, which are generated by the WRB and resource providers. In turn, the symmetric keys are encrypted with the private keys procured before workflow execution. This is a standard key management technique, utilising the speed of symmetric cryptography to protect the large quantities of data, and the key distribution advantages of public key cryptography to protect the less bandwidth-intensive symmetric keys.

Process Provenance: Platform attestation is used to reliably collate audit data for process provenance. Before accepting intermediate data from a resource provider, our scheme requires that the resource provider attests to its platform state post-job execution (steps 4–6). This provides three advantages. Firstly, any compromise can be detected immediately — if this occurs near the beginning of the workflow then considerable resources are saved from unnecessarily processing the rest of the workflow using incorrect data. Secondly, it allows the WRB to react by rescheduling the job to another resource provider. Thirdly, a record of attestation results are kept to provide a detailed audit trail.

Since the WRB requires resource providers on which workflows terminate to attest to their platform states, and input data for the workflow resides on trusted storage nodes, our proposal provides a complete audit trail to augment any additional provenance system being used. Thus, the proposed scheme enables the detection of any resource providers that may have compromised workflow results. An additional system will be required to manage the audit information collected, and this will be explored in future work.

6 Final remarks

Grid workflows provide significant advantages when completing highly complex computations if strong assurances that participating entities will behave as expected can be provided. This requires both the judicious selection of trustworthy Grid resource providers, and a means to determine whether or not this trust still holds after job processing. This trust is built using Trusted Computing technology — there exists challenges in the implementation, as discussed in [1], and this will be the focus of future work. We have presented a novel scheme that enables trusted resource provider selection, protects the integrity and confidentiality of jobs within a workflow and provides data for process provenance. The provision of these security services enables Grid users to derive confidence in the execution of their workflows, and from this establish trust in workflow results.

Acknowledgements

References

1. S. Balfe and E. Gallery. Mobile agents and the deus ex machina. In *The IEEE 21st International Conference on Advanced Information Networking and Applications (AINA-07), Niagara Falls, Canada, May 21-23, 2007*, pages 486–492. IEEE Press, May 2007.

2. D. Chadwick. Authorisation in Grid Computing. *Information Security Technical Report*, 10(1):33–40, 2005.

3. H. Chen, J. Chen, W. Mao, and F. Yan. Daonity — Grid security from two levels of virtualisation. *Information Security Technical Report*, 12(3):123–138, 2007.

4. A. Cooper and A. Martin. Towards a secure, tamper-proof grid platform. In *Proceedings of the 6th IEEE International Symposium on Cluster Computing and the Grid, Singapore, May 2006*, pages 373–380. IEEE Press, May 2006.

5. A. Cooper and A. Martin. Towards an open, trusted digital rights management platform. In *Proceedings of the ACM workshop on Digital rights management (DRM '06), Alexandria, Virginia, USA, October 30, 2006*, pages 79–88. ACM Press, Oct 2006.

6. A. W. Dent and C. J. Mitchell. *User's guide to cryptography and standards*. Artech House, 1st edition, 2005.

7. I. Foster and C. Kesselman. *The Grid 2: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, San Francisco, 2nd edition, 2004.

8. I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke. A security architecture for computational grids. In *Proceedings of the 5th ACM conference on Computer and Communications Security, San Francisco, California, United States, November 2–5*, pages 83–92, New York, Nov 1998. ACM Press.

9. V. Haldar, D. Chandra, and M. Franz. Semantic remote attestation — A virtual machine directed approach to Trusted Computing. In *Proceedings of the 3rd USENIX Virtual Machine Research & Technology Symposium (VM '04), San Jose, CA, USA, May 6-7, 2004*, pages 29–41. USENIX, May 2004.

10. Intel. LaGrande Technology Architectural Overview. Technical Report 252491-001, Intel Corporation, Sept. 2003.

11. International Organisation for Standardization. *ISO/IEC 19772: Information technology – Security techniques – Authenticated encryption*, 2007.

12. H. Löhr, H. V. Ramasamy, A.-R. Sadeghi, S. Schulz, M. Schunter, and C. Stüble. Enhancing Grid security using trusted virtualization. In *Proceedings of the 1st Benelux Workshop on Information and System Security (WISSEC '06), Antwerpen, Belgium, November 8-9, 2006*. COmputer Security and Industrial Cryptography (COSIC), K.U. Leuven, ESAT/SCD, Nov 2006.

13. H. Löhr, H. V. Ramasamy, A.-R. Sadeghi, S. Schulz, and C. Stüble. Enhancing grid security using trusted virtualization. In *Autonomic and Trusted Computing (ATC 2007), Hong Kong, China, July 11-13, 2007*, pages 372–384. Springer-Verlag (LNCS 4610), Jul 2007.

14. W. Mao, A. Martin, H. Jin, and H. Zhang. Innovations for grid security from trusted computing — protocol solutions to sharing of security resource. In *Proceedings of the 14th Int. Workshop on Security Protocols, Cambridge, UK, March 2006, to appear.* Springer-Verlag LNCS.

15. A. Martin and P.-W. Yau. Grid security: Next steps. *Information Security Technical Report*, 12(3):113–122, 2007.

16. C. J. Mitchell. *Trusted Computing*, volume 6 of *IEE Professional Applications of Computing*. IEE Press, London, 2005.

17. S. Pearson, editor. *Trusted Computing Platforms: TCPA Technology in Context*. Prentice Hall, 2003.

18. M. Peinado, P. England, and Y. Chen. An Overview of NGSCB. In C. J. Mitchell, editor, *Trusted Computing*, IEE Professional Applications of Computing Series 6, chapter 7, pages 115–141. The Institute of Electrical Engineers (IEE), London, UK, April 2005.

19. S. Rajbhandari, I. Wootten, A. S. Ali, and O. F. Rana. Evaluating provenance-based trust for scientific workflows. In *Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid, Singapore, May 2006*, pages 365–372. IEEE Press, May 2006.

20. A.-R. Sadeghi and C. Stüble. Property-based attestation for computing platforms: Caring about properties, not mechanisms. In *Proceedings of the 2004 Workshop on New Security Paradigms (NSPW '04), Nova Scotia, Canada, September 20-23, 2004*, pages 67–77. ACM Press, Sep 2004.

21. Y. L. Simmhan, B. Plale, and D. Gannon. A survey of data provenance in e-Science. *ACM SIGMOD Record*, 34(3):31–36, Sep 2005.

22. S. Song, K. Hwang, and Y.-K. Kwok. Risk-resilient heuristics and genetic algorithms for security-assured grid job scheduling. *IEEE Transactions on Computers*, 55(6):703–719, Jun 2006.

23. I. J. Taylor, E. Deelman, D. B. Gannon, and M. Shields, editors. *Workflows for e-Science: Scientific Workflows for Grids*. Springer, 2007.

24. P.-W. Yau and A. Tomlinson. Using trusted computing in commercial grids. In B. Akhgar, editor, *Proceedings of the 15th International Workshops on Conceptual Structures (ICCS 2007), Sheffield, UK, July 22-27, 2007*, pages 31–36. Springer-Verlag, Jul 2007.

25. J. Yu and R. Buyya. A taxonomy of scientific workflow systems for grid computing. *ACM SIGMOD Record*, 34(3):44–49, Sep 2005.