Enhancing Grid Security Using Workflows, Trusted Computing and Virtualisation

Po-Wah Yau[1] **and Allan Tomlinson**[1]
[1]Information Security Group, Royal Holloway, University of London, Egham, Surrey, TW20 0EX, UK

Abstract— *This paper highlights the need to meet both Grid user and resource provider security requirements, describing the rationale for securing Grid workflows: a set of tasks arranged into a logical order to process a Grid user's dataset. An overview of a secure protocol using Trusted Computing technology is provided, which is further enhanced with platform virtualisation hardware and software. The proposed scheme allows the selection of trustworthy resource providers and gives confidentiality and integrity protection to the workflow, the Grid user's processes and data. The scheme also detects any problems during workflow execution, collecting information that can be used for process provenance.*

Keywords: security, workflows, trust, virtualisation

1. Introduction

Grid computing [1], [2] is a distributed computing model that uses technological and social collaboration to solve data or computation-intensive problems. This is achieved by providing organised management and access to heterogenous resources: these include CPU cycles, databases, application software, computationally steerable machinery and storage space; all made accessible via Grid resource providers.

In general, a two-layer administrative domain hierarchy exists. The top layer is an umbrella administrative domain called a Virtual Organisation (VO). The bottom layer consists of separate autonomous domains that host Grid-enabled resources. This separation of domains can be at many levels, for example, groups in a department, departments in an organisation, organisations in a collaborative project etc. Authorised members of the VO will be able to access VO member resources, even though they reside in different domains.

As implied above, access control was an immediate priority during the inception of Grid computing. Grid middleware, such as those based on the Globus Toolkit [3], use the Grid Security Infrastructure (GSI) [4]. This model focuses on the authentication and authorisation of Grid users and their processes. The area has expanded to include identity and credential management [5], [6], [7]. The end result is an architecture that allows Grid resource providers to enforce both local and VO security policies.

Grid computing is relatively mature in academia [1], [8] and is attracting interest from industry, although widespread adoption has been limited. This is, to some extent, due to the security requirements of potential Grid users. Even academics are wary of the potential risk of exposing high-value data to a third party. However, this situation is changing. This paper looks at using Grid workflows as a means of meeting some of the main user security requirements. This is achieved at two levels: managerial and technical. The managerial level involves decision making at the workflow design phase, and this includes methods that can be applied immediately. The technical level focuses on using additional technology to provide security for workflow execution where required.

The rest of this paper is organised as follows. In section 2 we introduce Grid workflows and explain how they can be used to manage risk to a Grid user's data. Section 3 contains an overview of Trusted Computing primitives, which are used to secure workflow execution as part of a protocol described in section 4. This section presents an overview of a scheme previously proposed in [9] and proposes several enhancements. Finally, conclusions are made in section 5.

2. Using Grid workflows

The main use of Grid computing is to process data to solve large problems, which often involve multiple tasks. Thus, it is likely that most Grid users will be following some form of workflow, either implied or explicitly specified. 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 [11]. 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. Note that input data could include applications and software libraries necessary to complete a particular task.

The complexity involved in managing workflow tasks, their relationships and execution has led to research into automated tools [10]. There are two main types of workflow tools: low-level and high-level [11]. Low-level workflow tools are those in which users specify the workflow in terms of Grid jobs and the physical resources that they will run upon. Such workflows were certainly used in the early stages of Grid workflow adoption. However, the trend in research has been to separate away the details of physical

Grid jobs, allowing users to create abstract workflow specifications. These are then 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.

With both types of tools, there is a need to select the appropriate Grid resource providers and schedule jobs to be submitted to them. This is a difficult task, so a WRB is designed to 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 [11]. 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.

2.1 Secure workflow design

Regardless of whether or not a Grid workflow tool is used, a workflow can be designed to decrease the risk to a user's dataset. Much work has already been carried out into building in fault tolerance into Grid workflows [11], [12]: similar due diligence should be made in terms of security. This requires a cycle of several iterations of design and assessment.

The main security principle that should be followed when designing the Grid workflow is separation of duty [13], [14], to to limit the cost should a single or small number of jobs be compromised. This may involve breaking workflow tasks into smaller sub-tasks to be processed by different service providers. However, there are two issues that could make this separation of duty difficult.

The first is where tasks involve the use of resources where the number of providers is low, for example, the querying of specialist databases or the use of unique computationally steerable equipment. This is made even more difficult by the fact that access time to such resources may be limited. Secondly, there will be tasks that will be fundamental to the security of the user's data, and these will often be the tasks that produce the final results. In both cases, these tasks should be protected using stronger security policies, requiring them to be processed by Grid resource providers that offer stronger security guarantees, such as those using Trusted Computing (more on this later in section 4). For workflow results when the cost of compromise is too high to rely on third party processing, it may be necessary to direct sensitive tasks for in-house processing.

A risk assessment of the current design proposal should be conducted, which can include the use of formal risk analysis tools, for example CRAMM[1]. In this case, the primary assets would be job data (input, intermediate and output data) and the workflow itself.

The goal of the risk assessment is to determine the value of each asset, their security requirements and subsequently the risk and cost of compromise. The results of this assessment are then fed back into the next iteration of the workflow design cycle. The main security requirements for the above assets are as follows:

- Confidentiality of job data: ensuring that unauthorised principals cannot read job data;
- Integrity of job data: ensuring that unauthorised modifications to job data are detected;
- Confidentiality of workflow data: ensuring that unauthorised principals cannot read workflow data; and
- Integrity of workflow data: ensuring that unauthorised modifications to workflow data are detected.

The first two requirements are prominent because users are outsourcing their data to, potentially untrustworthy, third parties for processing. Integrity is as, perhaps more, important than confidentiality because violation of this requirement will lead to invalid results, a loss of reputation for the user, the Grid resource provider and perhaps Grid computing as a whole.

The last two requirements reveal how important it is to concentrate on protecting the workflow as well as job data. With proper workflow design, 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, especially those that are performing high-value work. Workflows are often large and dynamic, and their integrity is fundamental to acquiring valid results.

2.2 Workflow security requirements

Trust in the WRB is critical, because 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. Therefore, it would be sound practise to use multiple WRBs to execute different parts of the workflow, again looking to enforce separation of duty.

Resource providers might be selected based on direct experience and/or other indirect metrics, such as reputation or trust measurements based on provenance services [15], [16], [17]. 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.

[1] www.cramm.com

Many observers have commented on the vulnerabilities surrounding Grid middleware and the subsequent risk of job execution compromise [18], [19], [20]. 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. This issue is especially prominent at the beginning of a workflow, as undetected compromise would lead to error propagation throughout the rest of the workflow, wasting resources on processing incorrect data.

Finally, audit 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 information 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[2] Trusted Platform Module (TPM) specifications [21], [22], [23]. These specifications describe a tamper-resistent device which provides the host platform with a number of security services. These include: 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. Trusted Computing (TC) also encompasses new processor designs [24] and OS support [25] which facilitate software isolation. These concepts are examined in more detail elsewhere [26], [27]. For the purposes of this paper we describe 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) [25]. 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).

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.

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[3] 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.

4. Securing Grid workflows

This section describes a protocol which uses Trusted Computing to provide additional security guarantees for Grid workflows. Full details of the protocol can be found in [9]. In the following text we provide an overview of the protocol, which provides the following security services:

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. An 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. Our protocol allows a Grid user to submit all, or part of, a workflow for execution on TC-enabled platforms, to gain strong assurances that the protected workflow has executed correctly, and that the data was protected from malicious entities.

4.1 The protocol

Once a user has designed a workflow according to the principles outlined in section 2.1, a trusted WRB is then selected. This could be either an in-house WRB, or a WRB discovered by other means. For example, a trusted resource broker verification service could determine the trustworthiness of an external WRB.

The workflow of Grid tasks that the user wishes to protect is passed to the WRB, together with an encompassing security policy. The WRB maps the workflow tasks to a

[2] www.trustedcomputinggroup.org

[3] Using a private attestation signing key.

set of jobs, which 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.

4.1.1 Key distribution

Consider a sequence of jobs $a_0, a_1, ..., a_n$ that make up a user's submitted 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). Details on how a resource provider can obtain a private key are given in [9]. 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.

4.1.2 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:

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

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 [28], [29] — $g_{K_i}(a_i||r_i)$ 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 a public verification key used to verify messages from RP_{i-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} could be either a resource provider from an unprotected part of the workflow, or the WRB itself. In the former case, the state α_{i-1} would not be sent. In the latter, if the WRB is TC enabled, then α_{i-1} would be the platform state of the WRB itself which is useful for auditing purposes.

4.1.3 Executing the workflow

The following is the process of workflow execution at an arbitrary RP_i, after receiving message 1 (see section 4.1.2). We assume that each message also contains both the identifier of its originator and a digital signature.

$$\text{RP}_{i-1} \rightarrow \text{RP}_i \quad : \quad ID_W||\text{ready} \quad (2)$$

$$\text{RP}_i \rightarrow \text{RP}_{i-1} \quad : \quad ID_W||C(r_{RP_i}) \quad (3)$$

$$\begin{aligned} \text{RP}_{i-1} \rightarrow \text{RP}_i \quad : \quad & ID_W||\alpha_{i-1}(r_{RP_i})|| \\ & g_{K'_i}(R(a_{i-1}))||e_{PK_i}(K'_i) \end{aligned} \quad (4)$$

$$\text{RP}_i \rightarrow \text{WRB} \quad : \quad ID_W||r_{RP_i}||\alpha_{i-1}(r_{RP_i}) \quad (5)$$

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 will have been generated by RP_{i-1} (see step 9).
8) Job a_i is processed along with $R(a_{i-1})$.
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 $e_{PK_{i+1}}(K'_{i+1})$.

4.2 Security analysis

Trust in the execution of the submitted workflow 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 jobs, together with input and output data, have confidentiality and integrity protection so that only authorised resource providers can process them. This is achieved through the use of a symmetric key that

only selected resource providers can access if, and only if, their platforms have not been modified. This is because the symmetric key is sealed to a TPM public key that is bound to a specific platform configuration.

In the reverse direction, it is essential to determine whether or not the selected resource providers were compromised when processing their allocated jobs. This is achieved by utilising the platform attestation challenge-response mechanisms introduced by TC. A full analysis of the protocol is given elsewhere [9].

4.3 Enhancements

In this section we propose three further enhancements that improve the management and security of Grid workflow execution. These are platform virtualisation, final key distribution and Grid access device security.

4.3.1 Platform virtualisation

One of the issues with using TC is with attaining security assurances from a binary representation of a particular platform configuration. 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 [30]. While research is ongoing in this area [30], [31], many believe that platform virtualisation offers a solution [20], [32], [33].

Platform virtualisation is the process of emulating a hardware platform as a virtual machine for a guest application or operating system to execute on. The guest system is encapsulated within a virtual disk image, which can be cloned and migrated between virtual machine hosts. Development of this technology initially began as a software process, but recently hardware extensions have been made available to further empower the process in x86 architecture [34], [35]. Therefore, with sufficient hardware resources, it is possible to run several virtual machines simultaneously on the same same host platform, as depicted in figure 1.

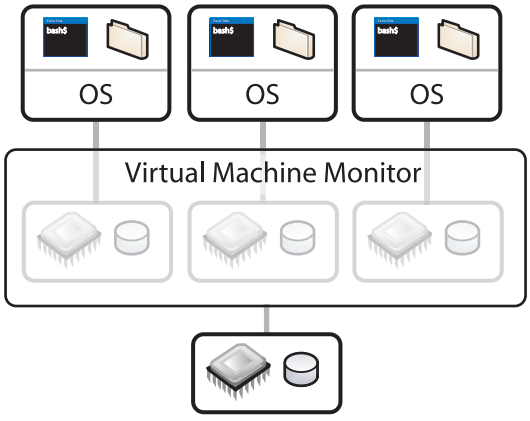

Fig. 1: A virtualisation setup.

The Virtual Machine Manager (VMM) is responsible for managing the virtual machines and access to the physical platform. More importantly, from a security viewpoint, the VMM enforces process isolation both between virtual machines, and between virtual machines and itself.

Another advantage that virtualisation offers is that the codebase for the VMM should be smaller than that for a conventional operating system. This should mean a more reliable method to test and gain assurances for the security of the host platform [20], [32], [33].

One of the benefits of virtualisation, especially for computational tasks, is that the need to target jobs at sites with specific software libraries, even operating system, is eliminated [36]. A Grid user can prepare a virtual disk image, with the software pre-installed, which can then be downloaded by the service provider. In terms of security, this can include rights management software for access control. It is also important to ensure that virtual disk images are themselves secured [32].

The impact of using virtualisation with the scheme described in section 4.1 is as follows. One of the key tasks for the WRB is the selection of resource providers that meet processing and security requirements. By creating and using downloadable virtual disk images, this process becomes easier for the WRB, and as a consequence, less trust has to be placed in the WRB. Also, less trust is required of a service provider to provide the correct software resources.

The issues surrounding the security properties of a complex platform are also mitigated. If we assume that service providers use well known implementations of VMMs, then the WRB wraps the relevant job and data within a virtual disk image, and this is encrypted with the symmetric key. According to our scheme, this key is now only accessible if the virtual disk image has been loaded, and that the both the VMM and virtual disk image have not been modified.

4.3.2 Final key distribution

As presented in section 4.1, a symmetric key is used to protect job results. It is possible that, for jobs that output final workflow results, these symmetric keys can be divided into several components. Each of these key components can be distributed, as an individual workflow job, to a resource provider for storage separate from where the workflow results are being stored. Purpose-built Grid service providers, dedicated to key storage, could be envisaged. If denial-of-service attacks are a possibility, which would prevent the whole key being recovered, then threshold cryptography techniques can be used so that only a certain number of components are needed to recover the whole key [37].

4.3.3 Grid access device security

Although access control mechanisms are already present, as discussed in section 1, they do not prevent a compromised Grid access device from obtaining sensitive results. Also,

there is evidence that Grid users regularly circumvent the GSI [38].

Therefore, the scheme can be extended to include the Grid access device. The key, or key components, used to protect the results of the final jobs in the workflow would be sealed to a required platform configuration, including a virtual disk image that also includes rights management software. By incorporating this into the workflow itself, the user is adding an extra layer of security to prevent unauthorised access to their results.

5. Conclusions

Well designed Grid workflows provide significant advantages when completing highly complex computations. Moreover, they enable a means of managing risk to a user's dataset. This, however, requires secure workflow execution. In turn, this requires the judicious selection of trustworthy resource providers, and a means to determine whether or not this trust still holds after job execution. Trusted Computing technology can be used to establish and maintain this trust.

We have proposed a scheme that allows trusted resource provider selection; protects the integrity and confidentiality of jobs within a workflow; and provides audit data for process provenance. The provision of these security services enables Grid users to derive confidence in the secure execution of their workflows. This provides a foundation of trust in workflow results that is furthermore strengthened by the benefits of using virtualisation technology.

Acknowledgements

The authors are sponsored by the Engineering and Physical Sciences Research Council (EPSRC) UK e-Science programme of research (EP/D053269).

References

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

[2] I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the Grid: Enabling scalable virtual organizations," *International Journal of High Performance Computing Applications*, vol. 15, no. 3, pp. 200–222, 2001.

[3] I. Foster, "Globus Toolkit version 4: Software for service-oriented systems," in *Proceedings of the IFIP International Conference on Network and Parallel Computing (NPC 2006), Tokyo, Japan, October 2-4, 2006*. Springer-Verlag (LNCS 3779), Oct 2006, pp. 2–13.

[4] 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*. New York: ACM Press, Nov 1998, pp. 83–92.

[5] D. Chadwick, A. Otenko, and E. Ball, "Role-based access control with x.509 attribute certificates," *IEEE Internet Computing*, vol. 7, no. 2, pp. 62–69, March 2003.

[6] J. Novotny, S. Tuecke, and V. Welch, "An online credential repository for the Grid: MyProxy," in *Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10), San Francisco, CA, USA, August 7-9, 2001*. IEEE Computer Society, Aug 2001, pp. 104–111.

[7] S. Tuecke, V. Welch, D. Engert, L. Perlman, and M. Thompson, "Internet X.509 public key infrastructure (PKI) proxy certificate profile," Internet and Engineering Task Force," RFC 3820, June 2004.

[8] T. Doyle, "Meeting the particle physics computing challenge," *PSCA International Public Service Review: Department of Trade and Industry*, no. 8, pp. 88–89, Autumn 2005.

[9] P. Yau, A. Tomlinson, S. Balfe, and E. M. Gallery, "Securing grid workflows with trusted computing," in *Proceedings of the 8th International Conference on Computation Science (ICCS '08), Krakow, Poland, June 23–25, 2008*. Springer-Verlag LNCS 5103, June 2008, pp. 510–519.

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

[11] J. Yu and R. Buyya, "A taxonomy of scientific workflow systems for grid computing," *ACM SIGMOD Record*, vol. 34, no. 3, pp. 44–49, Sep 2005.

[12] G. Kandaswamy, A. Mandal, and D. A. Reed, "Fault tolerance and recovery of scientific workflows on computational grids," in *In proceedings of the Eighth IEEE International Symposium on Cluster Computing and the Grid (CCGRID 2008), Lyon, France, May 19-22, 2008*, T. Priol, L. Lefevre, and R. Buyya, Eds. IEEE Press, May 2008, pp. 777–782.

[13] R. A. Botha and J. H. P. Eloff, "Separation of duties for access control enforcement in workflow environments," *IBM Systems Journal*, vol. 40, no. 3, pp. 666–682, 2001.

[14] J. H. Saltzer and M. D. Schroeder, "The protection of information in computer systems," *Proceedings of the IEEE*, vol. 63, no. 9, pp. 1278–1308, Sept 1975.

[15] 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*. IEEE Press, May 2006, pp. 365–372.

[16] Y. L. Simmhan, B. Plale, and D. Gannon, "A survey of data provenance in e-Science," *ACM SIGMOD Record*, vol. 34, no. 3, pp. 31–36, Sep 2005.

[17] S. Song, K. Hwang, and Y.-K. Kwok, "Risk-resilient heuristics and genetic algorithms for security-assured grid job scheduling," *IEEE Transactions on Computers*, vol. 55, no. 6, pp. 703–719, Jun 2006.

[18] 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*. IEEE Press, May 2006, pp. 373–380.

[19] 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.

[20] A. Martin and P.-W. Yau, "Grid security: Next steps," *Information Security Technical Report*, vol. 12, no. 3, pp. 113–122, 2007.

[21] *TPM Main Part 1 Design Principles Specification Version 1.2 Revision 94*, Trusted Computing Group, March 2006.

[22] *TPM Main Part 2 TPM Data Structures Version 1.2 Revision 94*, Trusted Computing Group, March 2006.

[23] *TPM Main Part 3 Commands Specification Version 1.2 Revision 94*, Trusted Computing Group, March 2006.

[24] Intel, "LaGrande Technology Architectural Overview," Intel Corporation, Tech. Rep. 252491-001, Sept. 2003.

[25] M. Peinado, P. England, and Y. Chen, "An Overview of NGSCB," in *Trusted Computing*, ser. IEE Professional Applications of Computing Series 6, C. J. Mitchell, Ed. London, UK: The Institute of Electrical Engineers (IEE), April 2005, ch. 7, pp. 115–141.

[26] C. J. Mitchell, *Trusted Computing*, ser. IEE Professional Applications of Computing. London: IEE Press, 2005, vol. 6.

[27] S. Pearson, Ed., *Trusted Computing Platforms: TCPA Technology in Context*. Prentice Hall, 2003.

[28] A. W. Dent and C. J. Mitchell, *User's guide to cryptography and standards*, 1st ed. Artech House, 2005.

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

[30] 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*. USENIX, May 2004, pp. 29–41.

[31] 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*. ACM Press, Sep 2004, pp. 67–77.

[32] C. Gebhardt and A. Tomlinson, "Secure virtual disk images for grid computing," in *In proceedings of the Third Asia-Pacific Trusted Infrastructure Technologies Conference, Wuhan, Hubei, China, October 14-17, 2008*, W. Mao, Ed. IEEE Press, October 2008, pp. 19–29.

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

[34] *AMD64 Virtualization Codenamed ŞPacificaŤ Technology, Secure Virtual Machine Architecture Reference Manual Publication No. 33047, Revision 3.00*, Advanced Micro Devices Inc., April 2005.

[35] G. Neiger, A. Santoni, F. Leung, D. Rodgers, and R. Uhlig, "Intel virtualization technology: Hardware support for efficient processor virtualization," *Intel Technology Journal*, vol. 10, no. 3, pp. 167–178, August 2006.

[36] I. Krsul, A. Ganguly, J. Zhang, J. A. B. Fortes, and R. J. Figueiredo, "VMPlants: Providing and managing virtual machine execution environments for grid computing," in *Proceedings of the 2004 ACM/IEEE conference on Supercomputing, Pittsburgh, PA, USA, November 6-12, 2004*. IEEE Computer Society, Nov 2004, p. 7.

[37] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[38] B. Beckles, V. Welch, and J. Basney, "Mechanisms for increasing the usability of grid security," *International Journal of Human-Computer Studies*, vol. 63, no. 1-2, pp. 74–101, July 2005.