Daonity: Protocol Solutions to Grid Security Using Hardware Strengthened Software Environment*

Wenbo Mao[1], Fei Yan[2], Chuanjiang Yi[3], and Haibo Chen[4]

[1] Hewlett-Packard Laboratories, China
[2] Wuhan University
[3] Huazhong University of Science and Technology
[4] Fudan University

Abstract. A central security requirement for grid computing can be referred to as behaviour conformity. This is an assurance that ad hoc related principals (users, platforms or instruments) forming a grid virtual organisation (VO) must each act in conformity with the rules for the VO constitution. Existing grid security practice has little means to enforce behaviour conformity and consequently falls short of satisfactory solutions to a number of problems.

Trusted Computing (TC) technology can add to grid computing the needed property of behaviour conformity. With TC using an essentially in-platform (trusted) third party, a principal can be imposed to have conformed behaviour and this fact can be reported to interested parties who may only need to be ad hoc related to the former. In this paper we report the Daonity system, a TC enabled emerging work in grid security standard, to manifest how behaviour conformity can help to improve grid security.

Keywords: Trusted Computing (TC), Trusted Computing Group (TCG), Grid Computing, Grid Security, Behaviour Conformity, Remote Platform Attestation, Secure Multi-party Computation, Secure Virtualisation.

1 Introduction

A computational grid [5,9,11] can be regarded as a next generation distributed computing system comprising a number — possibly large — of physically separated resources, each subject to their own various security, management and usage policies, to combine to a federated computing environment called *virtual organisation* (VO). The name "grid" follows analogously from tapping electricity supplied by the power grid, meaning that computational resources nowadays

* An Open Grid Forum Project (`https://forge.gridforum.org/projects/tc-rg/`) for developing a grid security standard, led by HP Labs China and participated by Wuhan University, Huazhong University of Science and Technology, Fudan University and Oxford University.

B. Christianson et al. (Eds.): Security Protocols 2007, LNCS 5964, pp. 204–221, 2010.

can and should also be tapped from super computers and data centres *elsewhere*. Early versions of computational grids were more or less confined to a high performance computing setting in which a grid VO comprises of one user plus a number of computational resource providers and/or data centres. Grid computing has now evolved to a more general setting of federated computing which supports sharing of resource and data not only for high performance computing but also involving science collaborations [5]. In the general federated computing setting, a VO of principals who are (may be plural number of) users, computing platforms or devices may be working on a number of common tasks and therefore having similar requirements on resource utilities.

A grid VO may be very dynamic, called into being for a single, short-lived task. In the most general setting, a VO of users and resource providers is geographically distributed and in different trust and management domains. These domains can span governmental, industrial and academic organisations. This implies, even demands, that strong security mechanisms be in place so that the grid services can be used in a secure and accountable manner. At an abstract level of description, two essential characteristics of grid security are:

System behaviour conformity. Because typical grid resources — infrastructure, applications, instrument or data — have critically high importance and value, a grid security strategy should be based mainly on attack prevention. While entity authentication is an important means for controlling access to resources and can also achieve attacker identification after an attack, it does not provide an effective means of attack prevention. This is better achieved with a behaviour conformity mechanism: an entity and its supporting computing system is attested that they have a restricted (and desirable) behaviour which cannot (easily) lead to any serious damage.

Group-oriented security. Resource sharing in a grid VO is, by definition, a group-oriented activity; a grid security solution must support such capabilities. Many accounts of grid design describe use scenarios entailing research data being shared by a group of scientists, large scientific instruments which must be operated by a group of users at the same time, or *ad hoc* collaborations such as a conference discussion among a group of entities (who therefore need to be served with a shared conference key). A useful (and difficult) case of group-oriented security is in the form of *secure multi-party computation* (SMPC) where proprietary data are input to a VO's common computational task in such a manner that no member of the VO should gain access to data input by any other participant after the joint computation.

Several aspects of grid security are well-explored: the use of public key cryptography with PKI identity and attribute certificates is quite well explored (and ongoing) for assuring identity of users, servers, and potentially software itself. These may be supported by a range of policy decision tools to enable authorisation mechanisms. Most grid applications entail code written in one place being executed in another. The problem of potentially malicious code and a trusted host is met by techniques such as sandboxing, code signing, or virus checking,

or simply through strong accounting so that if the code's execution causes substantial cost, its owner is required to pay substantial sums.

The dual of the last problem — trusted code required to run on a potentially malicious host — is harder to address. The possession of a host identity certificate is no guarantee that its administrators are not interfering with the execution of software, observing its inputs and outputs, or simply not offering the promised quality of service. Techniques of code obfuscation may make reverse engineering of software arbitrarily hard but for practical purposes it is unsafe to distribute code and assume that no one will be able to break or subvert it. Theoretical approaches from cryptography and/or statistics hold promise, but are hard to integrate with existing code, or require substantial overheads in order to work.

In recent years, increased computer security has been the goal of many efforts made by the computing industry. Among the many ideas, we are specifically focusing on the Trusted Computing (TC) initiative by the industrial standard body, Trusted Computing Group [28]. The purpose of TCG is to develop, define, and promote open, vendor-neutral specifications for trusted computing. It begins with a simple idea: integrating to a platform a low-cost tamper-resistant hardware module to enable and manage data and digital identities more securely within the platform's environment, protecting them from external software attack and physical theft. The TCG work has so far been developed with sufficient innovations to achieve its goal. These include hardware building block and software interface specifications across multiple platforms and operating systems' environments. TCG's open specifications (versions 1.1b and 1.2, available at the "Downloads area" of [28]) not only define reasonable notions of trust and security, but also provide concrete mechanisms to achieve protections by means of policy and trusted environment conformance.

Many authors have remarked on the suitability of these systems for distributed computing or even grid computing but the details are sketchy. Recently, as the TCG technology — hardware modules and the related device drivers — is becoming available, it is timely to consider how it may in practice assist in some grid application scenarios. We observe that the TCG mechanisms for policy and trusted environment conformity can provide a needed role in grid security. This is particularly suitable for our two grid security characteristics listed above. In this paper we propose an innovative approach to grid security from Trusted Computing effort.

1.1 Organisation of This Paper

The remainder of this paper is organised as follows. In §2 we consider grid security requirements. In §3 we overview the current grid security solutions and identify their inadequacy with respect to our two characteristics for grid security. In §4 we overview the Trusted Computing technology. In §5 we consider Trusted Computing technology as the complementary solution to the identified problems in the grid security. The technical presentation of this paper ends in §6 where we provide discussions on issues in the Daonity system implementation and some TCG technology realisation issues.

2 Grid Security Requirements

The US Department of Energy (DoE) Office of Advanced Scientific Computing Research published a report which provides a good summary of the requirements for grid security [5]. The grid requires a security infrastructure with the following properties:

I) Ease of use by users.
II) Conformation with the VO security needs while at the same time working well with site policies of each resource provider site.
III) Provisions for appropriate authentication and encryption of all interactions.

In this paper, we shall refer to this set as the "DoE Grid Security Requirements." We hold the view that DoE Grid Security Requirements II and III are compatible with our two characteristics for grid security. Below we provide more clarifications on this view by refining the grid security requirements.

In the general setting of a grid VO, principals are distributed in different trust and management domains which can span governmental, industrial and academic organisations. These principals are also ad hoc related to one another. This is because (i) a VO usually does not have a reliable control over a principal as a real organisation does over its employees and assets, (ii) these principals need not maintain a responsible relationship to one another as ones should in a real organisation, and (iii) a VO is dynamic, usually comes up into being, grows, diminishes or terminates, in a un-predetermined manner.

Despite the ad hoc and dynamic properties, grid computing needs strong security services. In addition to usual security services for conventional distributed computing to protect mainly owned or organisationally controlled assets against external adversaries, a principal in grid computing also has interest on a platform which is out of the principal's ownership or organisational control, and the needed protection is often against the very owner of the platform. Here are a few typical grid security problems.

Security for grid user. Most grid applications entail code written in one place being executed in another. A host platform's owner should not be able to compromise a guest user's security. For example, a guest algorithm running on a host may need protection, in data confidentiality and integrity, for the guest's input to the algorithm and the output result to be returned back to the guest. The protection may need to have a strength against even a privileged entity (e.g., superuser) at the host.

Security for grid resource provider. A guest user should not be able to compromise security, e.g., to cause damage to data or devices, at a resource provider. The protection may need to be sufficiently strong against a collusion among a group of VO users.

Conformable VO policy. However ad hoc a VO may be, it still needs conformable policy. For example, a VO policy may be that, any participant must not be able to disseminate certain VO owned data outside the VO. The difficulty

here is the *conformity* of the policy to be maintained despite the ad hoc nature of the VO. For example, even with little control over its members, a VO must still be able to remove a member without letting VO data be taken away.

Auditability. Any misuse of resource by users, and compromise to users' data and/or computations possibly by a privileged entity at a resource provider, must be detected in a undeniable manner.

Thus, to protect a user's interest on a platform which maybe beyond the user's organisational control is the distinct nature of grid security. We can summarise here a threat model for grid security.

Threat Model for Grid Security
VO participants are collaboration partners as well as potential adversaries to one another. A participant has interest needing protection in computing environments which are under the control of the other participants.

We shall use "partner-and-adversary" to name this threat model. With this threat model grid security encounters subtle problems.

As will be studied and analysed in detail in §3, existing and mainstream security practice for Grid security, in fact, mainly that supported by Grid Security Infrastructure (GSI) [10,21] for a standard Grid middleware Globus Toolkit [14], is essentially a result of direct applications of the standard public-key authentication infrastructure (PKI). The implied trust model in the direct application of PKI for the VO in GSI is the following. An unknown principal will be deemed trustworthy if it has been introduced by a trusted third party (TTP). It is hoped that the introduced principal will behave in a responsible manner since it should try its *best effort* to honor the introduction of the TTP. Note, however, this is a hope. We remark that in this introduction based trust model a TTP is usually positioned *outside* the system of partners. For example, if a protocol involves Alice and Bob who needs a TTP's service, the TTP is usually not an active or inline participant in the protocol; in particular, the TTP is usually not placed inside the platforms of the protocol participants. Unfortunately, the introduction based trust model actually does not suit grid security very well. Clearly, for grid security facing partner-and-adversary threats, Alice can have little control whether or not the proxy credentials will be misused. In order to mitigate the potential loss or misuse of the proxy credentials, GSI stipulates a policy that a proxy credential has a short lifetime of 12 hours. This is obviously a rather coarse policy and greatly limits the power of grid computing. We can say that the VO constructed in the current GSI is only suitable for a collegial environment in which partners are colleagues or friends alike. As will be analysed and discussed in §3, GSI, as straightforward applications of public-key authentication infrastructure, falls short of satisfactory solutions to a number of Grid computing problems.

Then what is exactly a desirable security mechanism we need for a computing environment with a partner-and-adversary threat model? We will need to place

a TTP *right inside* the computing platform owned by the participant to protect the interest of the other participant(s).

3 Current Grid Security Solutions

3.1 Authentication

The Grid Security Infrastructure (GSI) [10] and MyProxy [20] are two important elements of many current grid security solutions.

The GSI, which is the security kernel of the Globus Toolkit [14], provides a set of security protocols for achieving mutual entity authentication between a user (actually a user's proxy which is a client-side computing platform) and resource providers. Entity authentication in the GSI protocols involves straightforward applications of the standard SSL Authentication Protocol (SAP) suite [12]. These standard applications can be considered as a "plug-and-play security solution." They achieve quick deployment and ease of use. As a result, the grid security protocols in the GSI are two-party mutual authentication techniques. Each party has a public-key based cryptographic credential in the formulation of a certificate under the standard public-key authentication infrastructure PKI X.509 [17]. The use of the standard PKI in grid security is not only suitable for the VO environment, but also has an important advantage: single sign-on (SSO). The latter means that each user only needs to maintain one cryptographic credential. As always, any security solution must not demand the user to invoke sophisticated operations or tools.

Using PKI requires each user to hold a private key as their cryptographic credential. This can be a demanding requirement for many users without a secure computing platform in their locality. MyProxy provides a lightweight solution. It uses an online credential repository which can deliver temporary grid credentials to the end user. This is achieved via simple user authentication mechanisms such as password. This can be enhanced via a one-time password such as through a SecureID card.

The combination of the GSI and MyProxy provides a credible solution to the DoE Grid Security Requirement I. The two-party authentication protocols of the GSI, however, do not provide an adequate solution to group oriented grid security applications. For example, consider the DoE Grid Security Requirement III: the GSI cannot easily achieve a common key for a VO-wide encrypted communication.

3.2 Authorisation

The grid authorisation landscape is far more varied. Products such as Akenti [26], Community Authorisation Service [22], VOMS [1] and PERMIS [8] take a variety of approaches. Most make further use of X.509 certificates for identity or other attributes. Typically, it is up to a virtual organisation to construct an authorisation regime which enables it to meet the security requirements and policy of resource providers. These services are related to DoE Security Requirement II.

3.3 Secured Communications

For a host of reasons, it is seen as desirable to achieve integrity or confidentiality of data and control communications in grid contexts. Although some have proposed using Virtual Private Networks for such a purpose, others have argued [19] that this is inappropriate. More commonly, transport level security (TLS/SSL) is employed. This has the benefit of being ubiquitous and highly interoperable, and supported by readily available hardware accelerators, but is emphatically a point-to-point solution.

Web Services Security (WSS) [4] is potentially much more flexible, and in principle more efficient (since only selected elements of the communication are encrypted) — though present implementations do not realise this. WSS takes a message level security approach by performing encryption at the Web Services layer, such as the XML messages. These solutions also make use of X.509 PKI. Observe that the services these latter solutions provide are orthogonal to DoE Grid Security Requirements.

Given the above, we can call the current grid security solutions "plug-and-play PKI" for a conventional client-server environment. It is clear that two-party protocols based grid security solutions neither directly nor effectively support a group-oriented security. Additionally, they do not have a inherent means for realising behaviour control for a remote user and its client system environment. For example, WSS can achieve message encryption between a resource provider and a user. However, there is no way for a stakeholder in the resource provider to know whether or not the remote client environment is compromised (perhaps by a malicious code) even though it knows that such a compromise is equivalent to the nullification of the channel encryption service.

4 Trusted Computing

In 1999 five companies — Compaq, HP, IBM, Intel and Microsoft — founded the Trusted Computing Platform Alliance (TCPA). In 2003 TCPA achieved a membership of 190+ companies, when it was incorporated to Trusted Computing Group (TCG) [28] as a vendor-neutral and not-for-profit organisation for promoting industrial standards for Trusted Computing technologies. TCG takes a distributed, system-wide approach to the establishment of trust and security. It defines a concrete concept of Trusted Computing (TC). We may consider TC as the desired and conformable system behaviour which is not only established and maintained in a platform environment, but can also be attested to a remote challenger.

The following four notions are at the core of the TC technology:

Trusted Platform Module (TPM): Each platform has a TPM which is a tamper-resistant hardware module uniquely integrated to a platform for conformed operation and secure storage. It is designed to perform computations which cannot be subverted by the platform owner, including the system administrator. These computations include some public key cryptographic operations (decryption and digital signature generation using a private key in the TPM), platform system status measurement, and secure storage.

Core Root of Trust for Measurement (CRTM): At platform boot time, the TPM measures the system's data integrity status. The measurement starts from the integrity of BIOS, then that of OS and finally to applications. With CRTM, it is possible to establish a desired platform environment by loading only well behaved systems. This is a strong requirement which is called "secure boot." TCG also permits a slightly weaker measured boot which is called "authenticated boot." In the latter the TPM will permit loading of code which does not pass the measurement but will only securely record the status of that which has passed the measurement for attestation purpose (see below).

Root of Trust for Storage: The measured integrity of an executable is represented by a cryptographic checksum of the executable. This is then securely stored in a TPM. The TPM component called *Platform Configuration Register* (PCR) holds this data in an accumulative formulation. The TPM has a number of PCRs; each of them can be used to accumulate system integrity data for one category of system executables, e.g., one PCR for OSs (a platform can run many copies of an OS, see §5.5) and one PCR for a family of specific applications. The stored platform environment status is maintained until system reboot.

Remote Platform Attestation: Remote platform attestation is the most significant and innovative element in the TC technology. Using cryptographic challenge-response mechanisms, a remote entity can evaluate whether a platform's system has desired and conformed behaviour. With this capability, a remote stakeholder can be assured, with confidence, of the desired and conformed behaviour of a platform. In §5.4 we will provide a concrete protocol specification to manifest the functionality of platform attestation.

We notice that with a platform having the above behaviour, the TC technology has met resistances by being interpreted as providing for monopoly control over the use of software; trusted computing has its detractors [2,3]. TCG considers this a misinterpretation because a TCG platform should be able to execute any software in the "authenticated boot" condition (see CRTM above). Others argue [24] that market forces, combined perhaps with light-touch regulation and scrutiny, will help to keep the world sane. We may also observe that faulty software abounds and will help to keep the market from becoming completely controlled by any single party.

At any rate, we are able to avoid this controversial issue here. In the attempted TC application to grid security there should be much less disagreement since grid computing either requires behavioural compliance from an individual user as a condition for using remote resources, or implies federation and cooperation among a group of users.

5 Trusted Computing for Grid Security

We believe that TC technology can offer good solutions to grid security problems for which current grid security solutions do not play a role. Specifically, we

argue that TC technology addresses particularly well the DoE Grid Security Requirements II and III in the partner-and-adversary threat model which we have discussed in §2.

5.1 Secure Storage of Cryptographic Credential

Unattended user authentication is an important feature in the grid. This means that a user working in a VO is mainly doing so via their proxy. Work within a VO may involve dynamic sessions of resource allocation and hence require user entity authentication without having the user present. In the GSI, and in MyProxy, this is achieved by having a user client platform be issued a proxy certificate. The cryptographic credential of this certificate (i.e., the private key matching the public key in this certificate) is simply stored in the file system of the platform protected under the access control of the operating system. In this way, the client platform does not need to prompt the user for cryptographic operations. The obvious danger of leaving a private key in the file space is mitigated by stipulating a short lifetime for the proxy certificate. The default lifetime of a proxy certificate in the GSI is 12 hours. Upon expiration, a new proxy certificate must be re-issued. We feel this is an unacceptable security exposure.

With a TCP containing a tamper-resistant TPM, it is natural to store a user's cryptographic credentials in the TPM, or under an encryption chain controlled by the TPM. In TC, each user of a platform can generate many copies of private keys with their matching public keys being certified in the standard X.509 PKI. Thus, even if a platform is under the control of an attacker, the attacker, though in this situation may be able to misuse the user's credential (still in a conformable manner), cannot retrieve any information stored in the TPM. Thus, in a TC enhanced grid security setting, the protection of user secret key credentials can be substantially improved.

5.2 Sharing of Security Resource by Roaming Professionals

In GSI, MyProxy provides a lightweight solution to roaming professionals to obtain grid services ubiquitously [20]. It uses an online credential repository which can deliver temporary grid credentials to the end user. This is achieved via simple user authentication mechanisms such as password. A user shares a password with MyProxy server. Whenever and wherever the user requests for a cryptographic credential by authenticating to the MyProxy server, the server will generate a proxy certificate for the user and this includes the private key. The certificate is sent to the user, with the private key encrypted using the shared password. As we discussed in the previous section, a proxy certificate with a password encrypted private key form a weak security mechanism. GSI prioritises ubiquitous services over strong security.

We should notice the most basic behaviour conformity property of the TPM: prohibition of even the owner of the TPM from accessing certain protected data. Let a TPM have a public key for use by remote users, such that the decryption must only be possible inside the TPM and the result is not easily accessible even

by the TPM owner, for example, the decryption result only exists in a memory location which prohibits the platform owner to access.

Now, a user who is not TPM equipped, perhaps because of a roaming professional whose home-base machine is a desktop, can use other people's TPM resource while obtaining a proper protection of her/his privacy, even from the TPM owner. Such a user may still use a MyProxy server to generate a proxy certificate (needn't be a short-lived one). The MyProxy server should encrypt the certificate using a public key of a given TPM, and make the certificate usable only by the user who should input to the TPM the correct password (also encrypted using the public key). The owner of the TPM equipped platform, if trying to gain an access to the user's proxy, must at least attack the password which the user has used in the protection of the certificate.

In this way, TC's conformed behaviour property enables a secure sharing of security resource (the TPM). We notice that, although TPM will not become everywhere available overnight, use of the TPM as a shared resource (can even be remotely shared) in some applications, such as grid security, can indeed happen within a short period of time.

5.3 Distributed Firewall for a VO

In a conventional organisation a firewall plays an effective role in protecting the information assets of the organisation. A conventional firewall relies for its function upon the notions of restricted topology and controlled entry points. More precisely, a firewall relies on the assumption that every entity on one side of the entry point (the firewall) is to be trusted, and any entity on the other side is, at least potentially, an enemy. Because many attacks are achieved via malicious connections which can be shielded by a firewall, firewalls are a powerful protective mechanism.

A grid VO is typically composed of multiple physically distinct entities which are in different organisations who usually do not (entirely) trust each other. There is no longer a notion of a restricted network topology. The current grid security solution does not utilise the notion of firewall based protection. A user (its proxy) enters a VO without bringing in its own computational resource. Such a VO is in a primitive stage: a user only uses resource "out there," rather than also contributing their own resource as well. In fact, many grids have value precisely because every participant becomes a taker as well as a giver. Imagine the augmented value of a medical research collaboration which combines small databases of some limited clinical trials information scattered in various hospitals into global database available for access and search.

Bellovin proposed a notion of distributed firewall [6] which exactly suits the situation of a grid VO. In a distributed firewall, a packet is deemed to be accepted or rejected according to whether it has an acceptable digital signature. The packet's acceptance not only depends on the validity of a signature, but also on the rights granted to the certificate.

At first glance it seems that the current grid security solutions can already achieve a distributed firewall for a VO since these solutions also use public key

cryptography and PKI authentication framework which enable the use of digital signatures. The main problem is that the short lifetime of a proxy certificate of any participant makes the packet-level signature verification a performance burden. We repeat that the acceptance of a signature in a distributed firewall application is not only on the validity of the signature in the conventional sense, it should also be judged on the firewall policy granted to a certificate. The short-lived proxy certificates used in the current grid solutions are mainly limited to "identity certificates": these certificates are not suitable for distributed firewall use which needs refined policies associated to an IP configuration. We can call a certificate for a distributed firewall use a "property certificate."

With TC technology making multiple long-term (node and property) certificates available to each a platform, a grid VO can readily implement a distributed firewall technique.

5.4 Attestation of Behaviour Conformity in a Remote System

A grid stakeholder has legitimate reasons to worry about whether a participating subsystem in a VO conforms to the VO's security policy. For example, consider the need for a remote platform, which is sending in a GridFTP query for some sensitive information, does indeed run the correct version of the GridFTP which will flush the downloaded data from the local memory without saving a local copy in the file system after using the data (or only save an encrypted copy). Likewise, a participating client in a secure multi-party computation (SMPC) task may also have similar concern with respect to its proprietary data input to a VO. In an SMPC, data input to a distributed algorithm (protocol) from each of the participating parties should be confidential to the group in such a manner that the group can jointly compute a result while none of the participant can gain any knowledge about input data from any other participants.

TC's notion of remote platform attestation is a ready solution for this sort of grid services. Now let us describe how platform attestation can convince a remote user conformed behaviour of the platform.

A TPM contains a number of registers called Platform Configuration Registers (PCRs). Each PCR accumulates cryptographic hash checksums of secure applications (software systems) which are currently running on the local platform. Let SA denote a secure application, e.g., part of a protocol for GridFTP or SMPC, and let $H(SA)$ be the hash checksum of SA. Suppose that a remote user Alice initiates the protocol which causes SA to run on a TPM equipped platform (which we denote by TPM-Platform). Since the application is a secure one, Alice concerns whether or not TPM-Platform does run the bona-fide copy of SA. Protocol 1 (in the box) specifies a typical case of platform attestation to allow TPM-Platform to attest to Alice regarding her concern.

Protocol 1: Remote Platform Attestation

1. (In response to Alice's initiation) SA in TPM-Platform generates a public/private key pair SA_{pub}, SA_{pri} and sends them to TPM;

2. TPM creates $H(SA)$ and accumulates it into a PCR in the following formula

$$PCR' \leftarrow PCR \oplus H(SA);$$

TPM applies an "Attestation ID Key" (AIK) to certify (i.e., digitally sign) the information about SA; we denote by

$$Cert_{\text{SA\_Start}} = \text{Sig}_{\text{AIK}}(SA, H(SA), SA_{pub}, PCR', PCR, Ctr);$$

here Ctr is TPM's counter value which increases monotonously in each instance of authenticated boot and cannot be reset (not even by the owner of TPM-Platform);
3. SA sends $Cert_{\text{SA\_Start}}$ to Alice; she verifies the validity of the certificate using (the public) AIK of TPM; it is Alice's responsibility to deem whether or not to accept SA, e.g., by checking if $H(SA)$ is the correct value (which should have already been publicised by another authentication server regarding SA); Alice shall also send a random challenge to SA in TPM-Platform;
4. SA responds by signing the challenge value using SA_{pri};
5. Upon Alice's acceptance of the response, she can be convinced that TPM-Platform does indeed run the bona-fide copy of SA in an authenticated boot session which is identified by Ctr;
6. Upon termination of SA, Alice can ask TPM to issue

$$Cert_{\text{SA\_End}} = \text{Sig}_{\text{AIK}}(SA, H(SA), Ctr);$$

having seen Ctr in $Cert_{\text{SA\_End}}$ unchanged from that in $Cert_{\text{SA\_Start}}$, Alice can further be convinced that the authenticated boot session of TPM-Platform has been maintained during the whole execution of SA; this assures Alice that SA has been running and then properly terminated in the correct (trusted) session in TPM-Platform.

The TC innovation in remote platform attestation provides a powerful solution to the integrity protection of resources. Integrity protection of resources is a serious problem which the current grid security techniques cannot solve.

5.5 Securely Virtualised OSs and Services as "Vaults"

Using the notion of a virtual machine (VM) [15], an area of memory in a computing system can be isolated from the rest of the system to provide a simulated computer as if it were a separate computer. One piece of hardware can even enable multiple general-purpose OSs. Relations between these OSs can be configured to satisfy various access control policies. Moreover, on a TPM-platform, an access control policy for a VM as an object of other software systems (maybe other VMs) on the same platform can be conformed and attested to a remote user of the VM by applying Protocol 1 in the preceding section. Let's use "attested VM" to name a VM which has attested to a remote user an access control policy the user desires. Garfunkel et al. [13] consider that an attested VM can be

a "lock down" OS (which they name "closed-box VM"). Such a lock down OS may only permit to run a given list of secure applications. Again, these secure applications can also have behaviour conformity features which can be attested to a remote user. Thus, a lock down OS can serve a remote user a "vault" like service over a foreign platform. The "vault" on the platform is not even accessible by the platform's owner. The remote user (Alice) can send her data encrypted by a public key of the "vault" to input to an secure application running on the "vault" and then obtain the computation result which sent back to her encrypted under her public key. This achieves a secure guest computation, and on top of it SMPC (see §5.4) is practical.

Secure guest computation is very relevant to grid services. In many enterprise organisations it is typical that many PCs run continuously while not being used for extensive periods of time, e.g., outside working hours. Also, in many organisations typical uses of a PC involve word-processing like jobs which require minimal resource utilisation by the prime PC user. According to studies by Microsoft [7] typical PC utilisation is between 10 and 20 percent. A similar situation also applies to the servers environment, e.g., [25]. With secure guest computation, it is realistic to suppose that large chunks of underutilised platform resources (enterprise PCs and servers) can be organised to provide services for external users (or applications). It is obvious that a stringent security policy conformity is necessary. A "vault" service can achieve exactly the needed stringency to protect the interest of the external users. For example, when faulty code used by a prime PC user crashes or hangs, the rest of the system services should continue serving uninterrupted.

5.6 Group-Oriented Security Using Credential Migration

Combining the distributed firewall technique of §5.3 with the remote platform attestation technique in 5.4, we can imagine a realisation of a group-oriented security for a VO. As in the case of a physical group, in a VO there also needs to be an entity acting as the group manager or a stakeholder. The group manager is responsible for defining and managing the group security policies. These policies can be tailored to the setup of each site. The group security policy definition, setting up and management can be achieved using the distributed firewalls technique by letting the manager play the role of a property certification authority who issues property certificates to the group members. The group policy enforcement is then achieved by the group manager challenging and verifying the property attestation with each member of the VO.

For example, upon satisfaction of an attestation according the VO security policy and the remote site policy, the manager could release a group session key to the attested remote environment and this group session key plays the role of the *Security Association* (in IPSec) for that entity to penetrate the distributed firewall (i.e., to secure each packet both in data integrity and in message confidentiality). Thus, conference discussions in the VO can be securely conducted and confined within the VO.

In Daonity, we achieve the agreement and distribution of a VO Security Association within the VO members using TCG's standard protocol *Credential Migration*. This protocol allows a *migration authority* (MA) of a cryptographic credential (a private key) in a TPM to move the credential to another TPM. In TCG, credential migration is designed for allow a user to retain her/his security parameters when changing (e.g., upgrading) platform. The essence of Daonity is to make an extensive use of this TCG functionality. When a user Alice creates a VO, she creates the VO credential in her TPM, and then migrates the private key of the credential to the TPMs of the other VO members.

6 Implementation Status and Known Challenges

We have planned to make Daonity an open-source system. The implementation of the Daonity system has greatly benefited from the open-source Trusted Software Stack (TSS) package TrouSerS [27], and the open-source grid middleware package GT4. In fact, apart from the TPM migration component, all other TSS parts of Daonity are readily adapted and modified from TrouSerS, and plugged into GT4.

The TPM migration component has been completed for the TPM chip version 1.1b manufactured by Infineon Technologies AG on a number of HP platforms. We have made the first release of the open source code of the Daonity system (available at [29]). With TCG's TSS soon to become available for TPMs of all complying vendors, we have planned to add the migration part to TCG's TSS and so Daonity will become usable over TPMs of those vendors.

In the remainder of this section we explain the methodology of the implementation work.

TCG has defined the security subsystems in such a manner so as to allow cryptographic applications to evolve easily from basic hardware protection mechanisms, such as key hardening, to more advanced capabilities, such as platform attestation and key backup and recovery services. The TCG whitepaper "Writing TCG Enabled Trusted Applications" (at the "Downloads area" of [28]) provides an overview of the strategies that application developers may employ in developing TCG-aware client applications.

The TCG Software Stack (TSS) provides trust services that can be used by enhanced operating systems and applications. The TSS uses cryptographic methods to establish security services and trust relationships, allowing applications to maintain privacy, protect data, perform owner and user authentication, and verify operational capabilities of the platform.

The TCG Crypto Service Providers (CSPs) provide features that are commonly associated with cryptographic functionality. A TCG-enabled platform typically supports both PKCS#11 [23] and the MS Cryptographic API (MS-CAPI). If an application developer has experience writing with PKCS#11 or MS-CAPI, it is relatively easy to provide basic TCG enabled capabilities. For most applications, the application developer may harden RSA asymmetric private key operations by simply calling the new CSP that is provided with TPM-enabled platforms. While there may occasionally be a subtle user experience

difference based on different vendors' TSS and CSP, TCG is working to develop common interfaces and actions that may, over time, facilitate a common user experience, independent of the platform.

In order to utilise the enhanced capabilities of TCG-enabled platforms, the application developer must use the SDKs provided by the TPM manufacturer or OEM to expose the advanced trustworthy capabilities. An application developer may take advantage of a trusted platform's attestation capabilities by modifying their applications to require and verify the proper credentials provided by an attestation server. Eventually, most of the TPM and platform vendors will support the necessary credentials for attestation to function properly. Interoperability and compliance testing is being put in place and all the platform vendors have committed to supporting this mandatory aspect of the TCG specifications. Attestation servers are available from multiple vendors, including Verisign and Wave Systems, and some of these server products can assist in bridging the capability requirements of the platform's current limitations.

TCG-enabled PC platforms with TPM version 1.1b, both in desktop and notebook machines are now widely available from several computing systems manufactures. These include Dell, Fujitsu, HP, IBM and Intel (TCG "Fact Sheet," available at the "Downloads area" of [28]). These commercial-off-the-shelf products offer key storage for securing users' cryptographic credentials.

6.1 Known Challenges

We list a number of known challenges and attribute them to be problems in realisation of some TCG technologies.

As noted by [16] and [18] the remote attestation envisaged above is disappointingly fragile. There are many elements contributing to the runtime environment of a given piece of code. Operating systems, dynamic libraries, virtual machines, configuration files, etc. may all be upgraded or patched, leading to an explosion in the number of environments to be certified. In a realistic production grid, this will certainly be the case. Although we may hope to limit the scope of this heterogeneity as much as possible (because other behaviours may change as a result of differences, not merely security properties) the number of likely variants is probably too great to manage. A benefit of the grid environment is the notion of a Grid Information Service (GIS), which might reasonably hold information about system configuration, and — if trusted — could hold relevant attestation information also.

Haldar et al. [16] propose *semantic attestation* wherein a "Virtual Trusted Machine" is attested using the TPM mechanisms, and then the programs running upon the virtual machine — Java or .NET perhaps — are attested by their *behaviour* rather than their binary properties (so that semantically neutral changes may be made at any time).

Marchesini et al. [18] describe a case study in which three gross levels of change frequency are envisaged: the operating system kernel is "long-lived" and attested by the TPM mechanisms; intermediate software (in their case, the code of an Apache server) is dubbed "medium-lived" and perhaps certified by a CA for the

sake of a community; and detailed software (web pages etc.) is "short-lived" and protected by an encrypted file system, with periodically-updated hashes covering its integrity.

Some combination of these features would seem ideal for a grid or web services context. We might determine that in a dedicated web services host, the environment up to the virtual machine is stable enough to offer TPM attestation; the individual services might be assured in other ways. Conversely, many grid applications will not run inside a virtual machine (although their controlling logic may) since they must exploit native processor performance as totally as possible — for these, other solutions will be necessary.

The challenge, then, for grid and TC is to find means of integration which will support the significant components of grid infrastructure in as seamless a manner as possible. It is necessary to support the whole lifecycle behaviour: provisioning and commissioning grid nodes, deploying software, authorising users and (critically) groups to perform particular actions, and so on. Support for fine-grained mandatory access control will require integration with the authorisation services discussed. Service descriptions will need to support the best that semantic grid services have to offer; grid information services will need to record configuration information for attestation purposes.

7 Concluding Remarks

As grid security is becoming a more and more important topic, a number of problems remains untackled by the current grid security solutions. We have identified group-oriented security and distributed system behaviour conformance as among the essential requirements for grid security while being indifferently supported by the current grid security solutions. We have argued that trusted computing technology, thanks to its inherent properties of group-oriented security and system behaviour conformity, can provide suitable solutions to the identified grid security problems.

As we are still in an early stage of problem identification and solution search, the suggested approaches should be considered as initial input to substantial further investigations, which should include not only their plausibility, but also their alignment with the current grid security solutions. Nevertheless, as hardware and software support for TC is gradually becoming available, it is timely to consider how such tools can be used to maximum effect in enhancing trust and security in grid environments.

Acknowledgments

Greg Astfalk reviewed an early draft of this paper and provided insightful comments and suggestions.

References

1. Alfieri, R., Cecchini, R., Ciaschini, V., dell'Agnello, L., Frohner, Á., Gianoli, A., Lörentey, K., Spataro, F.: VOMS, an authorization system for virtual organizations. In: Fernández Rivera, F., Bubak, M., Gómez Tato, A., Doallo, R. (eds.) Across Grids 2003. LNCS, vol. 2970, pp. 33–40. Springer, Heidelberg (2004)
2. Anderson, R.: TCPA/Palladium frequently asked questions (2003)
3. Arbaugh, B.: Improving the TCPA specification. In: IEEE Computer, pp. 77–79 (August 2002)
4. Atkinson, B., et al.: Specification: Web Services Security (WS-Security), Version 1.0, (April 05 2002)
5. Bair, R. (ed.), D. Agarwal, et al (contributors). National Collaboratories Horizons, Report of the August 10-12, National Collaboratories Program Meeting, the U.S. Department of Energy Office of Science (2004)
6. Bellovin, S.: Distributed Firewalls. ;login: pp. 39-47 (November 1999)
7. Bolosky, W.J., Douceur, J.R., Ely, D., Theimer, M.: Feasibility of a service distributed file system deployed on an existing set of desktop PCs. In: Proceedings of International Conference on Measurement and Modelling of Computer Systems, pp. 34–43 (2000)
8. Chadwick, D.W.: RBAC policies in XML for X.509 based privilege management. In: Proceedings of SEC 2002 (2002)
9. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. In: Computational Grids. ch. 2, pp. 15–51. Morgan Kaufmann, San Francisco (1999)
10. Foster, I., Kesselman, C., Tsudik, G., Tuecke, S.: A security architecture for Computational Grids. In: 5th ACM Conference on Computer and Communications Security, pp. 83–92 (1998)
11. Foster, I., Kesselman, C., Tuecke, S.: The anatomy of the Grid: Enabling scalable virtual organizations. International Journal of High Performance Computing Applications 15(3), 200–222 (2001)
12. Freier, A.O., Karlton, P., Kocher, P.C.: The SSL Protocol, Version 3.0. INTERNET-DRAFT, draft-freier-ssl-version3-02. txt (November 1996)
13. Garfunkel, T., Rosenblum, M., Boneh, D.: Flexible OS support and applications for Trusted Computing. In: The 9th Hot Topics in Operating Systems, (HOTOS-IX) (2003)
14. Globus Toolkit 4, `http://www-unix.globus.org/toolkit/`
15. Goldberg, R.: Survey of virtual machine research. IEEE Computer Magazine 7, 34–45 (1974)
16. Haldar, V., Chandra, D., Franz, M.: Semantic remote attestation — a virtual machine directed approach to trusted computing. In: VM 2004, USENIX (2004)
17. ITU-T. Rec. X.509 (revised) the Directory — Authentication Framework, International Telecommunication Union, Geneva, Switzerland (equivalent to ISO/IEC 9594-8:1995) (1993)
18. Marchesini, J., Smith, S., Wild, O., MacDonald, R.: Experimenting with TCPA/TCG hardware, or: How I learned to stop worrying and love the bear. Technical Report TR2003-476, Department of Computer Science, Dartmouth College, Hanover, New Hampshire (December 2003)
19. Martin, A., Cook, C.: Grids and VPNs are antithetical. In: Chivers, H., Martin, A. (eds.) Workshop on Grid Security Practice and Experience (2004)

20. Novotny, J., Teucke, S., Welch, V.: An Online Credential Repository for the Grid: MyProxy. In: Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10), IEEE Press, Los Alamitos (August 2001)
21. Open Grid Forum. Overview of the GSI, http://www.globus.org/security/overview.html
22. Pearlman, L., Welch, V., Foster, I., Kesselman, C., Tuecke, S.: A Community Authorization Service for Group Collaboration. In: Proceedings of the 3rd International Workshop on Policies for Distributed Systems and Networks, p. 50 (2002)
23. RSA Security. PKCS#11 v2.20: Cryptographic Token Interface Standard (June 2004), http://www.rsasecurity.com/pub/pkcs/pkcs-11/v2-20/pkcs-11v2-20.pdf
24. Safford, D.: Clarifying misinformation on TCPA (October 2002)
25. Servers Unilization, http://www.serverwatch.com/
26. Thompson, M., Essiari, A., Mudumbai, S.: Certificate-based Authorization Policy in a PKI Environment. ACM Transactions on Information and System Security (TISSEC) 6(4), 566–588 (2003)
27. TrouSerS. The Open-Source TCG Software Stack, http://www.trousers.sourceforge.net/
28. Trusted Computing Group, http://www.trustedcomputinggroup.org
29. Trusted Computing Research Group, Open Grid Forum, http://www.forge.gridforum.org/projects/tc-rg/