Innovations for Grid Security from Trusted Computing

Wenbo Mao[*], Hai Jin[†], and Andrew Martin[‡]

[*]Hewlett-Packard Laboratories, Bristol
Filton Road, Bristol BS34 8QZ, UK
wenbo.mao@.hp.com

[†]Huazhong University of Science and Technology
Wu Han 430074, China
hjin@mail.hust.edu.cn

[‡]Oxford University Software Engineering Centre
Wolfson Building, Parks Road, Oxford. OX1 3QD. UK
Andrew.Martin@comlab.ox.ac.uk

7th June 2005

Abstract

A central problem for grid (or web) services is how to gain confidence that a remote system is performing according to its published description. In particular, issues of confidentiality and data integrity rely at present on 'best efforts' or weak social trust mechanisms. These are insufficient for a large class of problems, but emerging trusted platform technologies offer great potential to improve this situation.

The Trusted Computing (TC) initiative developed by the Trusted Computing Group (TCG) takes a distributed-system-wide approach to the provisions of integrity protection for resources. The TC's notion of trust and security can be described as conformed system behaviours of a platform environment such that the conformation can be attested to a remote challenger. We consider that such a notion of integrity protection of resources naturally suits the security requirements for Grid computing or science collaborations. We identify and discuss in this paper a number of innovations that the TC technology could offer for improving Grid security.

Keywords

Trusted Computing (TC), Trusted Computing Group (TCG), Grid Computing, Grid Security.

1 Introduction

A computational Grid [14, 17, 19] is a distributed computing system comprising a number—possibly large—of physically separated resources, each subject to their own various security, management and usage policies. It is intended to support a variety of users who may be working on a number of common tasks and have similar resource requirements. A system of such collaborators and resource providers may be described as forming a virtual organisation (VO). Some such VOs 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.

Therefore, two essential features of Grid security are:

System behaviour conformation 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 conformation 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 a conference discussion among a group of entities (who therefore need to be served with a shared conference key).

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, the Trusted Computing Group [28]. The purpose of the TCG is to develop, define, and promote open, vendor-neutral specifications for trusted computing. It begins with a simple idea: using a cheap tamper-resistant hardware module to enable and manage data and digital identities more securely, 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 environments. The 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. Since such hardware — and suitable 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 conformation can provide a needed role in Grid security. This is particularly suitable for our two Grid security characteristics. In this paper we propose an innovative approach to Grid security from Trusted Computing effort.

The remainder of this paper is organised as follows. In §2 we consider Grid security requirements, illuminating these by some use cases in §3. In §4 we overview the current Grid security solutions and identify their inadequacy with respect to our two characteristics for Grid security. In §5 we overview the Trusted Computing technology. In §6 we consider Trusted Computing technology as the complementary solution to the identified problems in the Grid security. Finally in §7 we provide discussions on issues of the TC implementation and deployment.

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 [14]. 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 the sequel, 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. More clarifications will be provided in the remainder of this paper.

3 Use cases

By way of illustration, we record some realistic security requirements of some Grid applications.

3.1 climate*prediction*.net clients

The climate*prediction*.net project seeks to use the best available climate models to produce large *ensemble-based* forecasts of the development of the world climate over the next 50 years. The approach taken is to distribute this model to tens of thousands of participants across the globe, inviting them to run it for a period of about six weeks, returning the results when complete to one of a number of project upload servers.

This project clearly has much in common with `SETI@home`, and indeed, both are now implemented using the BOINC [2] platform. It differs from `SETI@home` in the scale of the task—work units in the latter complete in a matter of hours rather than weeks, and their file transfer requirements are measured in kilobytes rather than megabytes.

Both projects must address the same problem, however: it is quite feasible for participants to return data which appears to come from running the downloaded software, but in fact arises from a different source. Either the participant has modified it (and some `SETI@home` participants have done this, seeking to help the project or to boost their personal ratings for work units completed) or they have completely fabricated the results (for a host of possible reasons). climate*prediction*.net has implemented a hashing checksum to guard against casual tampering or creation of results — but has declined to enter an unwinnable ‘arms race’ against the determined hacker [12].

In both projects, the impact of such behaviour might be substantial: false positives or negatives in the search for anomalous signals in `SETI@home`’s case; biased statistics in the climate*prediction*.net case. The first project has been very successful in attracting participants, and so frequently has far more compute power available than it has data to process. As a result, searches can be duplicated several times over among participants, and non-matching results simply discarded.

The second project, with its much greater resource requirements and commitment, has a lower number of participants, and cannot afford to duplicate model runs on a large scale. Happily, though, because the entire modelling effort is a statistical one, individual out-lying results are not a significant problem, provided no systematic bias is introduced. For climate*prediction*.net, then, it suffices to duplicate a small random sample of runs, and to use a pairwise comparison to estimate the accuracy of the whole ensemble. [9]

For both projects, a much more robust solution would be to gain some kind of assurance that the software running was the intended software, with the intended inputs, and that the results returned are those created by the software, unaltered.

3.2 Grid data/compute nodes

Clearly, any kind of subversion which may arise in the climate*prediction*.net host could equally well arise in a more tightly-coupled grid node also. With the system administrator's connivance or otherwise, the host might appear correctly to process grid computing jobs, but might falsify results, or might retain a record of 'interesting' inputs or outputs.

Many instances of such concerns are already known:

3.2.1 climate*prediction*.net upload servers

The climate*prediction*.net upload servers are themselves donated resources from sympathetic academic departments across the world. Because the whole dataset is massive and cannot easily be collected in once place, it is stored, and must be processed, at these donated nodes. However, no genuine guarantee can be had of their ability or willingness to enforce data integrity (storing results without perturbation), or to compute derived results accurately.

3.2.2 Bioinformatics and databases

A similar concern arises in the bioinformatics arena, where much work relies upon queries against common databases. Information about what those queries are is commercially sensitive — it will indicate an area of study to a competitor — and so although those databases are public they will often be copied and held (at substantial cost) in-house, because their system administrators are not trusted to refrain from harvesting query information. An alternative approach is to run 100 queries at a time, only one of which is a genuine request, in order to confound anyone looking for patterns. Neither is a very efficient scheme.

3.2.3 Sensitive code

A further deployment scenario arises in the Integrative Biology project, where the security of data is not a major concern, but the code being run represents the fruit of much research effort — and so is valuable intellectual property. The heart modeller whose career has been devoted to making a particular model may wish to run it on a high-performance computer but does not want to run the risk of the administrator (or, worse still perhaps, another user) of that machine taking a copy of the code.

In each case, one would wish either to certify the whole software stack ahead of time, or to subject it to audit, and to then have a remotely-checkable guarantee that the audited software is that still being run. We would wish to do this in a platform-independent manner, since in an ideal grid context we should neither know nor care which particular host is running our software or hosting our data.

3.3 Medical Informatics

Increasing complexity of electronic patient records raises a requirement for forms of mandatory access control. Different parts of each patient's records should be accessible by varying sets of people: administrative staff, nurses, general practitioners, specialists. For purposes of epidemiology and other research, anonymised or pseudonymised records should be available. Although attempts are being made to keep the access rules simple, the current societal interpretation of issues of consent mean that some detailed fine-grained rules are required. If an element of a patient record is stored in an encrypted form, the decrypted form should only be available to individuals in selected roles, and where suitable facilities for record-keeping and audit exist.

If follows that a reasonable requirement might be for a clinician receiving sensitive information to be prevented from passing it to a third party. At the moment, no strong mechanisms exist to enforce this. As a result, the UK NHS, for example, uses separated networks, which drives up costs, and can nevertheless not give strong guarantees of separation (because the network is so large that we cannot reasonably imagine it is homogeneous or completely audited).

Such issues of *end-to-end security* [11] are not at all addressed by present Grid solutions; nor is it clear how they might.

4 Current Grid Security Solutions

4.1 Authentication

The Grid Security Infrastructure (GSI) [18] and MyProxy [23] are two important elements of many current Grid security solutions.

The GSI, which is the security kernel of the Globus Toolkit [21], 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 [20]. 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 [22]. 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.

4.2 Authorization

The Grid authorization landscape is far more varied. Products such as Akenti [27], Community Authorization Service [24], VOMS [1] and PERMIS [5] 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 authorization regime which enables it to meet the security requirements and policy of resource providers. These services are related to DoE Security Requirement II.

4.3 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 [8] 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 [13] 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. WS-S 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, WS-Security 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.

5 Trusted Computing

In 1999 five companies—Compaq, HP, IBM, Intel and Microsoft—founded the Trusted Computing Platform Alliance (TCPA). In 2003, the TCPA achieved a membership of 190+ companies. TCPA was then succeeded by the Trusted Computing Group (TCG) [28]. The 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): This is a tamper-resistant hardware module 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. Each platform has a TPM.

Core Root of Trust for Measurement (CRTM): At the 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.’ The 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 OS's (a platform can run many copies of OS's, see §6.4) and one PCR for a family of specific applications. The stored platform environment status is maintained until system reboot.

Remote Platform Attestation: Using cryptographic challenge-response mechanisms, a remote entity can evaluate whether a platform's system has desired and conformed behaviour. Remote platform attestation is the most significant and the most innovation element in the TC technology. With this capability, a remote stakeholder can be assured, with confidence, of the desired and conformed behaviour of a platform.

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 [3, 4]. The 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 [10] 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 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.

6 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.

6.1 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. A TPM can be configured to hold keys in a 'non-migration' mode which will never reveal any private key (up to the tamper-resistance level for which a TPM is designed). Keys can also be configured as 'migratabe', wherein key material can be explored with the owner's consent. 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.

6.2 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 [15] 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 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 certificates used in the current Grid solutions are mainly limited to 'identity certificates': they are not suitable for distributed firewall use which needs refined policies. We can call a certificate for a distributed firewall use a 'property certificate.'

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

6.3 Attestation of Policy Conformation 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. Likewise, a participating (a giver) client may also have similar concern with respect to a VO.

TC's notion of remote platform attestation is a ready solution for this sort of service. Each user has 'Attestation ID Keys' (AIKs) which can sign PCRs in a TPM. For details, see 'Root of Trust for Storage' in §5: a PCR is a securely stored (in the TPM) cryptographic checksum of a specific executable and the secure storage is current for a session since the platform was booted. Therefore, a digitally signed PCR value, which is verifiable upon a challenge by a stakeholder using a public AIK, provides an assertion that the current instantiation of the platform is running the specific executable. Notice that a PCR stores the system integrity data in an accumulative formulation which does not limit the number of executables being stored, and there are plenty of PCRs (minimum of 24) in a TPM, hence complex site policies can be defined by combining PCRs.

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.

6.4 Securely Virtualised OS's and Services as 'Vaults'

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 [16] PC utilisation's are between 10 to 20 percent. A similar situation also applies to the servers environment, e.g., [26].

Using the notion of a virtual machine, 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 OS's. Relations between these OS's can be configured to satisfy various access control policies. It is thus realistic to suppose that large chunks of underutilised platform resources (enterprise PCs and servers) can be organised to provide services for other users (or applications). It is obvious that a stringent security policy conformation is necessary. One basic feature of such policies is that a virtualised OS or service should function like a 'vault' which confines its users and processes to certain behaviours which cannot affect the rest of the system. For example, when buggy code used by a prime PC user is gone dead, the rest of the system services should continue serving uninterrupted.

The TC technology can provide a strong guarantee for a stringent policy conformation. Secure OS and service 'vaults' can be loaded with respective PCRs measured and stored in the TPM. IT security administrators in a resource contributing enterprise can challenge a system in the realm from time to time for policy attestation to make sure the proper functioning of the 'vaults.'

6.5 Group-oriented Security

Combining the distributed firewall technique of §6.2 with the remote platform attestation technique in 6.3, 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 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 language) 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 this environment can be securely conducted.

7 Trusted Computing Implementation and Deployment Status

The 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 the MS Cryptographic API (MS-CAPI) and PKCS#11 [25]. If an application developer has experience writing with MS-CAPI or PKCS#11, 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, the TCG organisation 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 the 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.

7.1 Known Challenges

As noted by [6] and [7] 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. [6] 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. [7] 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.

8 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 conformation, 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.

Acknowledgements

Greg Astfalk reviewed an early draft of this paper and provided insightful comments and suggestions. Graeme Proudler provided help with the TC technology background. Paul Vickers provided help with the Grid computing background. Nigel Edwards and Dirk Kuhlmann provided insights in the notion of secure virtualisation.

References

[1] R. Alfieri, Roberto Cecchini, Vincenzo Ciaschini, Luca dell'Agnello, Ákos Frohner, A. Gianoli, Károly Lörentey, and Fabio Spataro. Voms, an authorization system for virtual organizations. In F. Fernández Rivera, Marian Bubak, A. Gómez Tato, and Ramon Doallo, editors, *European Across Grids Conference*, volume 2970 of *Lecture Notes in Computer Science*, pages 33–40. Springer, 2003.

[2] D. P. Anderson. BOINC: A system for public-resource computing and storage. In *Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*, Pittsburgh, PA, November 2004.

[3] Ross Anderson. TCPA/Palladium frequently asked questions, 2003.

[4] Bill Arbaugh. Improving the TCPA specification. *IEEE Computer*, pages 77–79, August 2002.

[5] D. W. Chadwick. RBAC policies in XML for X.509 based privilege management. In *Proceedings of SEC 2002*, 2002.

[6] Vivek Haldar, Deepak Chandra, and Michael Franz. Semantic remote attestation — a virtual machine directed approach to trusted computing. In *VM'04*. USENIX, 2004.

[7] John Marchesini, Sean Smith, Omen Wild, and Rich MacDonald. 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.

[8] Andrew Martin and Carl Cook. Grids and VPNs are antithetical. In Howard Chivers and Andrew Martin, editors, *Workshop on Grid Security Practice and Experience*, 2004. CHECK THE REFERENCE.

[9] Andrew Martin, David Stainforth, and Neil Massey. Verification of results in climate*prediction*.net. forthcoming, 2005?

[10] David Safford. Clarifying misinformation on TCPA, October 2002.

[11] Jerome H. Saltzer, David P. Reed, and David D. Clark. End-to-End Arguments in System Design. *ACM Transactions in Computer Systems*, 2(4):277–288, November 1984.

[12] David Stainforth, Andrew Martin, Andrew Simpson, Carl Christensen, Jamie Kettleborough, Tolu Aina, and Myles Allen. Security principles for public-resource modeling research. In *IASTED* , 2002.

[13] B. Atkinson, et. al. Specification: Web Services Security (WS-Security), Version 1.0, 05 April 2002.

[14] R. Bair (editor), D. Agarwal, et. al. (contributors). National Collaboratories Horizons, Report of the August 10-12, 2004, National Collaboratories Program Meeting, the U.S. Department of Energy Office of Science.

[15] S. Bellovin. Distributed Firewalls, *;login:,* November 1999, pp 39-47.

[16] W.J. Bolosky, J.R. Douceur, D. Ely and M. Theimer. Feasibility of a service distributed file system deployed on an existing set of desktop PCs. In Proceedings of International Conference on Measurement and Modeling of Computer Systems, 2000, pages 34-43.

[17] I. Foster and C. Kesselman. *The Grid: Blueprint for a New Computing Infrastructure*, chapter 2: computational Grids, pages 15–51. Morgan Kaufmann, San Francisco, 1999.

[18] I. Foster, C. Kesselman, G. Tsudik and S. Tuecke. A security architecture for Computational Grids, 5th ACM Conference on Computer and Communications Security, pages 83–92, 1998.

[19] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the Grid: Enabling scalable virtual organizations. *International Journal of High Performance Computing Applications*, 15(3):200–222, 2001.

[20] A.O. Freier, P. Karlton, and P.C. Kocher. The SSL Protocol, Version 3.0. INTERNET-DRAFT, draft-freier-ssl-version3-02.txt, November 1996.

[21] Globus Toolkit. Available at `www-unix.globus.org/toolkit/`.

[22] ITU-T. Rec. X.509 (revised) the Directory — Authentication Framework, 1993. International Telecommunication Union, Geneva, Switzerland (equivalent to ISO/IEC 9594-8:1995.).

[23] J. Novotny, S. Teucke and V. Welch. An Online Credential Repository for the Grid: MyProxy, Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10), IEEE Press, August 2001.

[24] L. Pearlman, V. Welch, I. Foster, C. Kesselman and S. Tuecke. A Community Authorization Service for Group Collaboration, Proceedings of the 3rd International Workshop on Policies for Distributed Systems and Networks, p 50, 2002.

[25] RSA Security. PKCS#11 v2.20: Cryptographic Token Interface Standard. 28 June 2004. Available at `www.rsasecurity.com/pub/pkcs/pkcs-11/v2-20/pkcs-11v2-20.pdf`.

[26] `www.serverwatch.com/`.

[27] M. Thompson, A. Essiari and S. Mudumbai. Certificate-based Authorization Policy in a PKI Environment, ACM Transactions on Information and System Security (TISSEC), 6(4), 566-588 (2003).

[28] `www.trustedcomputinggroup.org`.