

Grid Security: Next Steps

Andrew Martin and Po-Wah Yau

May 2007 - Final draft v1.4

Abstract

One of the more mature instances of a service-oriented architecture is the model known as Grid Computing. Computational Grids and Data Grids are becoming commonplace in certain sectors, yet the style of security they implement is suitable only for a fairly small subset of possible user communities. Using some case studies and experience, we describe the existing Grid security models, explain why they represent shortcomings for some applications, and describe some emerging architectures, Trusted Computing and virtualisation, which help address the problems.

1 Introduction

Since Foster [15] and others described and popularised the concept of a Grid, one of the central challenges for this style of computing has been finding the means to put in place effective security measures. As the style known as ‘Grid’ is an instance of a service-oriented architecture, many of the issues which arise here also have a wider application in modern distributed systems.

On one level, the objective of any Grid middleware is to present a collection of distributed components as a single machine, so that disparate users (and resources) gain a largely uniform view of the ‘system’. The result is ideally much like a modern multi-disc, multi-processor workstation, wherein the user and application

software need take no interest in which processor or which disc will contain or process their data: the Grid is an operating system for a vast distributed computer (see Figure 1).

It is little wonder, then, that the main issues of security that have been addressed by Grid architects to date have to do with achieving authentication and authorization of users and their programs — and, to a lesser extent, of the systems on which they will execute or be stored [4, 6, 49]. This is no small task: the actors (users, processes, resources) in the Grid may reside within different administrative domains, and the usage patterns of the Grid entail supporting rapidly-formed dynamic ‘virtual organizations’. Moreover, many of the uses of Grids entail non-interactive jobs, wherein some or all of the user’s capability must be delegated to some kind of software agent.

The challenges raised by this kind of security have been quite thoroughly explored by a range of current Grid technologies [7, 33, 36, 44, 47]. These appear to have addressed immediate security requirements of authorisation and authentication: whether they have adequate *usability* properties is a matter of more debate and some issues are reviewed in Section 2.1. That section also considers the strong security requirements which arise when significant pieces of access control functionality are concentrated in a single place.

Substantially different architectures arise in the ‘high throughput’ computing models such as *BOINC* [2] and *Condor* [42]. The lat-

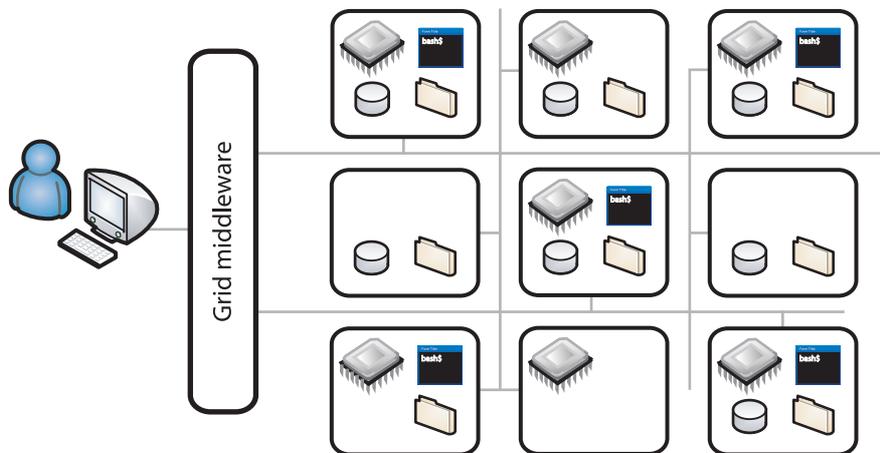


Figure 1: The Grid concept.

ter is well-suited to recruiting spare cycles on desktop computers within an organization; the former to ‘volunteer’ computing projects such as SETI@home and *climateprediction.net*. In these models, the compute nodes are substantially simpler, and the consequent security requirements are also different. We review one such case study in Section 2.2.

Significant classes of application have simply not embraced the Grid style of working at all: even if ‘functionality’ proof-of-concept tools have been produced, practical security concerns have prevented their deployment. Sending one’s software, or data, or experimental parameters outside one’s well-controlled security domain is clearly not something to be undertaken without suitable guarantees of integrity and confidentiality: we explore this in Section 3.

Section 4 draws some unifying themes and technical challenges from these case studies, and Section 5 provides an overview of Trusted Computing and virtualisation — these emerging technologies may help to fill some of the gaps in the existing pattern of Grid security architectures, and Section 6 presents a selection of current proposals. There are some open

issues with trying to fully realise the synergy of Grid, Trusted Computing and virtualisation for security purposes: some thoughts to the road ahead are given in Section 7. We end the paper with some concluding remarks in Section 8.

2 Case Studies

The two case studies in this section provide details of security concerns that have been addressed during the development of Grid technology. The first is an overview of the Grid Security Infrastructure (GSI) [17, 16, 49], a de facto architecture that has been adopted by many Grid implementations, which deals with the requirements of authentication and authorisation. The second, *climateprediction.net*, reveals different security issues that relate to protecting hosts, and to the reliability of results. Both studies reveal issues with Grid computing that we expand upon in Sections 3 and 4.

2.1 Globus-based Grids

The Globus toolkit [12], currently in its fourth version¹, provides a Grid middleware implementation that is widely used in the scientific community [14]. The security model for Globus, and indeed many other Grid middleware technologies, is based on the Grid Security Infrastructure (GSI), first proposed by Foster et al. [17].

The goal of GSI is clear — to provide services for Grid resource providers to enforce both local and VO security policies using existing authentication and authorisation protocols. This authentication must be durable across multiple hosts, in different security domains, and for scalability, possibly require the identification of users (possibly from different, unknown domains).

These requirements and others have led to the adoption of mutual authentication protocols that employ X.509 identity certificates, thus relying on a public-key infrastructure (PKI), of which trust relationships are well understood [43]. If either principal participating in the mutual authentication protocol is not satisfied then the transaction will not proceed. Here, satisfaction is the ability to verify the chain of trust from the principal to the respective roots of trust, i.e. certification authority (CA), and trusting the CA used. As the global collection of projects and organisations implementing Grid solutions has grown, a collaboration of mutually-trusting and mutually-auditing certificate authorities has organised itself to manage a *Grid Policy Management Authority*² (Grid PMA).

Following the ethos of Grid computing, other authentication protocols can be used, for example, Kerberos [31]. This requires credential translation services and additional policy rules

to be specified (the WS-Trust specification [34] contains details on how to specify how different credentials can be ‘trusted’). Regardless of the type of credentials used, there is a common theme — to provide entity authentication. In the rest of this section, we discuss Grid PKI, which requires that users and resources register their public keys with a trusted CA to each acquire an X.509 certificate — this is their long-term credential.

A Grid user’s software client will typically use the user’s certificate and corresponding private key (pass-phrase protected) as part of SSL/TLS [11] based mutual authentication with another Grid entity, most likely a resource provider or resource broker. If it is a resource provider, then upon successful authentication, the resource provider will create a temporary local account for the user, which is controlled according to the local security policy.

Since some computations are subject to batch submission and/or are themselves very long running, it is necessary for their users to delegate some capability to Grid software to act on their behalf when they are not present. This capability also provides a single-sign-on service, which is appealing for Grid usability. The Globus architecture uses a system that involves *proxy certificates* [47] to satisfy the above requirement.

The user’s client generates an asymmetric key pair and a certificate on-the-fly, with a (relatively) short-lived lifetime appropriate to the task at hand. The certificate is signed with the user’s long-term private key, so its validation path involves the user’s long-term certificate as well as that certificate’s issuer — the short-term certificate is a proxy for the user’s identity. The private key corresponding to this certificate can be kept online, in a location accessible to the Grid software (and without pass-phrase protection) so that when the Grid software needs to undertake a task on the user’s behalf, it can do so using the temporary key

¹See www.globus.org/toolkit for more information.

²www.gridpma.org

pair and proxy certificate for authentication — decreasing the risk of compromising the user’s long term credentials.

Of course, we expect that the user’s long-term private key must be password protected and stored in a secure place: this raises significant challenges for useability [28]. The concept of the proxy is that, because it is short-lived, its private key can be less heavily protected. Members of the Grid PMA have described constraints upon how users must store their private keys (see, for example, the UK e-Science Certificate Policy which specifies that private keys must be protected using a pass-phrase of at least fifteen characters [21]).

One possible solution to the problem of secure usable storage for keys, and also to providing users with access to their keys whilst mobile, is to store certificates in a central repository. This has been discouraged (or banned outright) by the policies described above, because, although a well-designed repository retains only encrypted keys, the store represents too much of a valuable asset that is too much of an attack target.

An acceptable compromise has been the use of an online credential repository tool like *MyProxy* [33] to hold medium-lifetime (say, one month) proxy credentials, which are used, in turn, to create and sign further proxy credentials which are short-lived (say, one day). This is an example of a delegation chain, where the user’s original credentials are being delegated, first to MyProxy, which then delegates to any entity that can provide the correct authentication information and satisfies the MyProxy security policy. MyProxy and its protocols are constructed in such a way that the private keys held by the tool never need to be exported: credentials are delegated to MyProxy so that MyProxy can also delegate on behalf of a user, perhaps to one of the user’s jobs that needs to request further Grid resources.

The delegation of trust is an extremely important concept in Grid computing. What we have discussed hitherto is that trust is built upon successful entity authentication, i.e. corroborating an identity, but this implicitly requires trusting the Grid entities involved to show due diligence to protect user credentials and also not to abuse those credentials. As discussed later in Section 3, this situation may not be acceptable to many potential Grid users.

2.2 *climateprediction.net*

The *climateprediction.net* project arose from an observation by Allen [1] — surprising at the time — that modern home PCs were equipped with enough power to permit them to run a credible climate model, previously the preserve of supercomputers. Since climate models involve a large number of underdetermined parameters, and assumptions about initial conditions, an accepted model for their use is to run an *ensemble* of models, each employing a different mix of parameters and conditions, and use these to form a *Monte Carlo* simulation producing a probabilistic forecast of future climate.

The project has delivered such models to hundreds of thousands of volunteers worldwide, and is steadily processing the results returned to make predictions [41]. Such a style of working (lately using the BOINC [2] platform) is of course just one corner of the broad range of architectures coming under the heading of ‘Grid computing’ — but for our present purposes it has features in common with many of the others, as we shall see.

Leaving aside any possible problems with the scientific methodology or accuracy of the executable model, the greatest threat to the project was that it should finish with insufficient data to make useful forecasts [40]. To this end, one of the greatest security priorities was that participants should join in large num-

bers, and be retained: there must be no taint of compromise to participants' privacy or the integrity of their machines. To encourage retention, we used web-based community tools and visualization tools. We employed code signing to give participants confidence that the package being installed was the one we intended, and relied upon the good name of the research organisations involved to assure participants that this was indeed safe.

The converse problem — the question of whether climate data returned to the centre truly arose from a run of the model — is much harder to tackle. We could have, of course, deployed a signature key with the climate model, and have it sign the results, but this would have not been difficult to reverse engineer, and sign bogus results also. To avoid creating an 'arms race' we used a simple (published but non-standard) hash for this task.

Why would anyone be interested in returning bogus results? Two motivations suggest themselves:

1. Those who participate in volunteer projects often like to vie with each other for positions on a 'leader board' recording how much CPU time they have donated: by returning bogus data, it might be possible to rise up the leader board without the tedium of actually running the climate model.³
2. The topic of climate prediction is politically sensitive. It is quite feasible that some interested party — potentially a very well-funded (perhaps covertly funded) party — would have an interest in adulterating the results, either to give

³In fact, the only significant 'security incident' to date on the project was related to this: someone created a trojan which installed the model on the PCs of unsuspecting users, and recruited them to his 'team'. This was an eventuality our *a priori* security analysis did not anticipate.

an alternative outcome, or perhaps more likely, simply to cast significant doubt upon the accuracy of the results.

A mitigation for the problem of bogus results can be achieved by sending identical tasks to different participants. This works well if there are sufficient participants, and is essential in a *search* task (like SETI@home, or a search for prime numbers). For a *simulation* task, if the threat is not too severe, the value gained from performing more simulations (rather than duplicating them) may outweigh the impact of a few rogue results [29].

To date, we have not detected (using a statistical sample of genuine duplicates [23]), any attempts to seed the project with bogus data. Anyone doing so for political ends would presumably have announced the fact following the publication of the first significant results [41].

We discuss similar concerns in different projects and Grid architectures in Section 4 below.

3 Roads not travelled

Perhaps the largest class of examples in this area comes from projects which have *not* embraced a Grid style of working. By the nature of these, it is hard to report upon them: not least because, if a concrete decision has been made, it usually relates to security concerns, which the parties concerned are not willing to discuss. Nevertheless, we can use the case studies above to identify the open security issues that contribute to the lack of Grid utilisation.

The *climateprediction.net* project anticipated that its sponsors — those who had provided software models or funding — would be willing to participate in running its software and so contribute to its success. In fact, organisational policies prohibiting (perhaps wisely) participation in volunteer computing efforts

made this very difficult: although some instances of the system specialized to in-house work were successfully deployed, these inherently lose many of the benefits of a simple global pool of participants. This model of Grid computing utilises resources that may not actually be owned by the participants (the users who have downloaded the software to operate on machines they do not ‘own’) whereas the converse is true in the model of Grid computing described in the first case study — Globus-based Grids. Here, the risk of running foreign, potentially malicious, code is controlled by applying traditional and well-understood (but not necessarily vulnerability-free) access control mechanisms to the user’s temporary local account at the resource.

So while a resource provider has at least some means of protecting their assets, what of the Grid user and their data? In a number of engineering contexts, the benefit has been recognised by using a Grid-style solution to achieve rapid large-scale modelling by recruiting resources across a number of collaborating organisations: but realistic deployments have often not taken place due to security concerns. If Grid computing is to be used, then it will be used to process large amounts of data with potentially high intellectual property value, and sending this ‘outside the firewall’ is simply a step too far.

Likewise, many other domains — financial sector, multimedia (rendering animated images, for example), medical applications — have seen huge potential for distributing calculations, but similar concerns exist. Medical applications have particularly strong additional legal requirements: any data that contains personal identifiable information must be anonymised. Moreover, after data has been used, it must be reliably deleted so that it is not in danger of being retained or correlated with other data without authorisation.

Consider also biomedical applications, such

as simulations for drug discovery. In a fiercely competitive industry, where even queries upon public databases must be buried in large quantities of noise (unwanted queries to hide the presence of the important one), the resistance to using Grid computing is understandably very great. Whereas the risk profile associated with containing those calculations within a single data centre is well-understood, the consequences of sending certain data to less highly-assured systems (still less, systems outside the organisation) are simply judged too dangerous to be worthwhile.

This touches on another point, that is, the trust that is placed in Grid services such as database queries. While Grid computing can be used to leverage computation time, its other use is for the federation of data provided by different, often disparate, information services [48]. The threat of falsifying results in the *climateprediction.net* Grid, discussed in section 2.2, also apply here, but the techniques needed to trust and gain confidence in returned results will be different in Globus-based Grids. In the case of the latter, a user will have no control over the implementation of the service it is using, other than to negotiate Service Level Agreements (SLAs) [10] which only provide an administrative control.

4 Unifying Themes

We have seen that many of the functional security requirements of the scenarios above are met by the Grid technologies available. Such an analysis could have considered in more detail the protocols enabling message level and transport level encryption, the construction of security tokens, policy expressions and other technologies for achieving authorization decisions and enforcement. These certainly provide the means to describe and enforce ‘hygienic’ separation and sharing, according to pol-

icy, among Grid users.

However, the assumption that systems and their administrators are benign was long ago nullified. For example, it has not been considered wise for users to set up connections between separate security domains which do not require authentication upon use, since by doing so one would also give access to the one domain by the administrators of the other. This problem looms very large in a Grid of many, many interconnected domains.

Some unifying themes, then, are:

- A Grid is, by necessity, a large and elaborate structure, with many administrators who will be motivated by a variety of different interests. Present models of Grid architecture and security do not expose explicitly who is being trusted to do what: these implicit trust relationships are often insufficient.

Resource brokers are entities that act on behalf of the user (using delegated credentials) to discover and use services that match user requirements. Whilst resource brokers are necessary to evolve the usability and potential of Grid computing, they are a powerful position to compromise the users that they are meant to serve.

Resource brokers rely on service registries/directories to which both brokers and service providers subscribe to. Brokers will trust that registries are up-to-date and contain accurate information. In turn registries trust that service providers accurately describe service details, such as security policy.

Just as the World Wide Web has evolved, from a simple information sharing medium where each site was broadly trustworthy on its own terms, into a home of many competing concerns and not a little fraud, we may expect a Grid to do the

same.

- There has been much emphasis so far on authentication of users, with substantial more recent interest in distributed authorization regimes and policy combining, necessary to support different credential types [36, 44]. It is commonplace for these to use an X.509 host certificate to authenticate themselves to users and other services, but the protocols surrounding the establishment of such identity are far from well-suited to dynamic provisioning of capacity — even though this is somehow inevitable in a production Grid environment. To improve the current architecture, Basney et al. [5] propose a trust negotiation stage during SLA negotiation, where both parties present a set of CAs that they trust, and X.509 based authentication is successful if there is at least one CA in the intersection of the two sets.

The protocols involved in X.509-based authentication imply a symmetric trust relationship, that is, if either the user or the Grid entity is not satisfied then the Grid transaction will not proceed. However, as noted in Section 3, a trust asymmetry exists with the authentication of systems, and the software running on them, which have received far less attention.

- The middleware supporting Grid work is necessarily ‘fat’ and hard to assure. It is inevitable that it will contain numerous points of vulnerability. By its networked and distributed nature it offers a natural large surface to any would-be attacker: whether their motivation is the subversion of the resources of the Grid, the theft of the data and software being used, or an attack upon the integrity of calculations.

For example, we are aware of a (so far unpublished) piece of analysis which demon-

strated significant vulnerabilities in the BOINC software platform — which would have permitted an attacker to run arbitrary code on the PCs of any nominated subset of BOINC (equivalently, SETI@home) participants.⁴

- The issue of complex middleware is exacerbated as more features are added, which in themselves may introduce more security vulnerabilities. For example, currently, any jobs running as part of a user-designed workflow are loosely coupled, so they are independent of one another. The Message Passing Interface (MPI) library for parallel programming has been adapted for the Grid environment, to enable closely coupled jobs that can communicate with each other while running [13, 22] — the potential for malicious jobs to send confidential data becomes even greater. The fact that Grid computing has moved towards Web Services, a technology which is sometimes promoted as unhindered by firewalls, does little to allay fears of unauthorised misuse.
- A ‘genuine’ Grid architecture would make the choice of underlying systems quite transparent: the data and computations should be agnostic about their selection; the resources entirely commoditised. If those systems are differentiated — by, for example, the kind of security offered, or the trustworthiness of their administrators — this abstraction fails.
- Few, if any, Grid security threats have materialized so far. We have observed

⁴The vulnerabilities had been present but apparently undetected for quite some time: they were quite visible in the freely-available source code for BOINC. This is an interesting counterexample to the ‘many eyes’ security argument for open source software, not least because the SETI@home participants are among the most technically savvy computer users on the planet. The crucial vulnerabilities have now been patched.

a little ‘noise’ around the edge of public participation Grid style activities — but these are relatively large, high-profile activities. The large-scale managed production Grids have seen very few security incidents (but some very evident vulnerabilities). However, it is the vulnerabilities that determine commercial confidence in using multi-domain Grids.

- The compute nodes of a Grid architecture embody an inherent mutual contract: there is an expectation that they will accurately perform the computations requested (or flag an error if this is not possible), and in return there is an expectation that they jobs sent to them will not damage or compromise them. This is a classic duality in distributed security: that of trusted code on an untrusted host (from one perspective) and untrusted code on a trusted host (from another).

We have observed that for many applications, existing security issues mean that deployment is not feasible at present: a consequent natural question is to ask where we should invest effort in research and development for the greatest benefit. The rest of this paper explores a promising possible direction.

5 Trustworthy components

The fundamental requirement revealed is the need to establish trust that provides assurances to the Grid user. The ability to trust that Grid services are correct invocations of their service descriptions is a goal of the work on provenance [20, 32]. Here, an instrumentation and monitoring architecture is used to enable a user to trace how a particular result has been derived. Thus, information such as the workflow of service providers used, parameters, decisions are recorded to create an audit trail that is

non-repudiable. It is this the ability to create this audit trail that builds a user’s confidence and trust in a Grid service, and often provides input into reputation models [35, 37]. While essential in their own right, even provenance systems rely on trusting key entities to operate correctly.

The requirements that can be derived from the discussion above are reminiscent of several of the design goals of the technologies of *Trusted Computing*⁵ [30]. These amount to an initiative from hardware and software vendors to adapt the commodity PC architecture to facilitate new security primitives that make use of a hardware root-of-trust called the Trusted Platform Module (TPM) [45] — a tamper-resistant chip that is integrated into a device’s motherboard. The features of Trusted Computing include:

- A strong means for the system to measure its own integrity at start-up, and in consequence, a means to achieve a strong guarantee of integrity for any (combination of) software component(s);
- One or more cryptographically strong identities — capable of attesting that software integrity to a third party;
- Support for the secure storage of a hierarchy of keys, intimately bound to the platform.

The TPM contains a set of Platform Configuration Registers (PCRs) to store securely a representation of the host platform’s boot process. This consists of a series of platform ‘integrity measurements’, essentially the hash digest of some software component on the platform. A trusted platform with a TPM can undergo an authenticated boot process — at each stage of the boot process a measurement is

⁵See also www.trustedcomputinggroup.org for the home of the Trusted Computing Group.

taken of the components required for the subsequent stage, before control is passed to those measured components [18] (see Figure 2).

The platform can attest to its configuration by presenting the platform measurements, digitally signed by the TPM, to a requesting principal. The ‘trust’ in Trusted Computing is the assertion that a platform is in a specific configuration. Whether this configuration is one that is ‘trustworthy’ for Grid computing is an issue we address later. To avoid confusion, we will state that a platform will meet ‘conformed behaviour’ instead of ‘trusted’ as used in Trusted Computing parlance.

Perhaps most critically, platform measurements can be used to ‘seal’ data: encrypted so that it can be accessed only by a platform in a particular state — running a particular software stack and application.

A related technology for creating trusted platforms is virtualisation — providing the ability to run multiple ‘virtual machines’ on one physical platform, managed by a ‘hypervisor’ or Virtual Machine Monitor (VMM), as shown in Figure 3. While the benefits of virtualisation for Grid computing has already been commented upon [24], they also offer several security properties, e.g. process isolation, which are seen as complementary to Trusted Computing. An example of the union of the two technologies is Intel’s LaGrande virtualisation technology [18], providing hardware support to create secure (and measurable) compartments for virtual machines to operate in.

6 Trusted Grids

Proposals that use Trusted Computing for Grid computing are rare, probably because the potential has not yet been fully realised (a situation that this article is meant to help rectify). Mao, Martin, et al. [26] discuss several possibilities such as using a TPM to store user

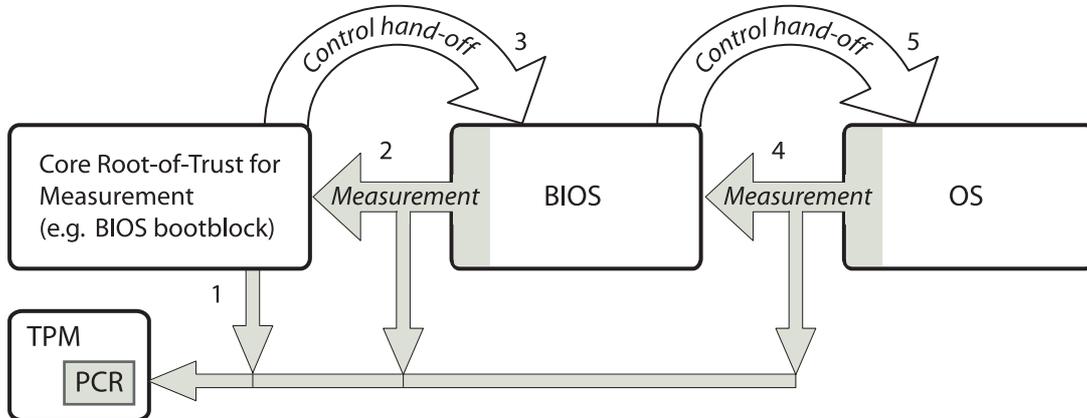


Figure 2: An example of an authenticated boot process.

credentials, or enforce VO policy as part of a distributed firewall. In this section, we outline state of the art that has been proposed since.

Cooper and Martin [8] propose a small, simple security manager component that contains and controls the VMM, and Grid services and user jobs are executed in their own VMs, whose isolation is guaranteed by the VMM. The principle is that by keeping the security manager component small, then it will be easier to gain high assurance of its security properties, in this case, by remote attestation. Instead of

delegating credentials to a Grid job (which in this architecture is contained in a VM), the credentials are delegated to the attested security manager to control and protect. The credentials are used to verify the integrity of a user job, retrieve job data from a storage element, verify job data integrity (if required). Data confidentiality can also be achieved: if the security manager can attest to a platform configuration that is satisfactory to the storage element (possibly according to user policy), then both entities perform secure key exchange/establishment of a symmetric key that the storage element uses to encrypt the job data.

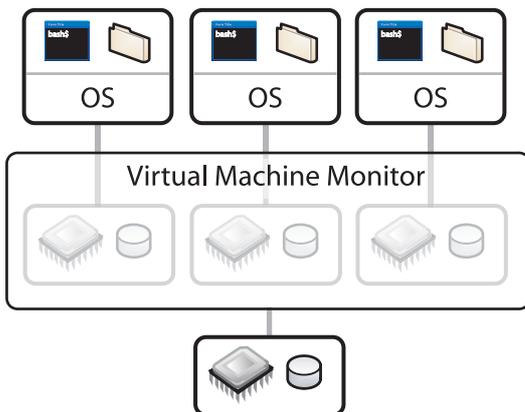


Figure 3: A virtualisation setup.

Löhr et al. [25] also propose that Grid jobs run in VM compartments, and that the virtualisation layer is attested to when searching for service providers. In addition, they propose that each service provider publish an *attestation token*, which contains the public key of a non-migratable private key held within the TPM of the worker node being advertised, and the PCR measurements of the platform configuration that the private key is ‘sealed’ to. Also included are the TPM’s signature on the token, to prove that it was produced by an authentic TPM, and the corresponding public key to ver-

ify the signature⁶.

Grid users compile a set of attestation tokens and select one that represents a configuration that they are willing to trust. Since this decision can be made offline, the overhead is negligible. Moreover, direct invocation of the TPM is unnecessary, which is appealing because TPM commands cannot generally be processed in parallel and the TPM could otherwise become a bottleneck.

Once a service provider (or more specifically, an advertised TPM-enabled worker node) has been selected, the user encrypts a session key using the provider's public key, retrieved from its attestation token. Therefore, only the provider in the attested configuration (and believed to be 'trustworthy' by the user) can access the session key. This implicitly provides freshness, because if the platform's state has been changed since its attestation token was produced, then the token is no longer fresh and cannot be used.

Both the solutions mentioned above require the full TC architecture to operate correctly. In the interim, the requirement of proxy delegation — and an element of something more sophisticated, a relocatable key to allow controlled group sharing of encrypted content — has motivated the *Daonity* project [27], also reported by Chen et al. elsewhere in this technical report. The Daonity system implements the credential migration capability of TPMs, allowing a VO group key to be securely transported to VO members that can attest to the state of their TPM-enabled platforms⁷.

The mechanisms of Daonity may be developed further to offer 'Digital Rights Management' (DRM) for Grid data. DRM has tended

to be associated with media companies wishing to protect music and movies against unwanted copying, but the same ideas apply naturally to the data and software being processed in a Grid context [9]: we would find it desirable to place policy-based access controls on those data, regardless of the Grid node and domain in which they find themselves. The technologies of Trusted Computing can enable this, by releasing data and keys only to suitably-attested platforms.

Finally, we mention two implemented systems that make use of TC capabilities where available — the Secure Audit Web Service (SAWS) [50] and the Secure Hardware Enhanced MyProxy (SHEMP) [28]. SAWS was originally implemented to provide a secure auditing function for the PERMIS authorisation framework [7], but has since been extended into a web service where the SAWS server ideally runs on a TPM-enabled platform. The TPM is used to store audit record sequence numbers, and private decryption and signing keys (of asymmetric key pairs).

The SHEMP project, reported by Marchesini and Smith [28], employs a trusted platform to strengthen the MyProxy tool described in Section 2, applying the idea of MyProxy to a more generalised setting outside of Grid computing. They outline the ideas behind *key usage policies* which users can specify when uploading their credentials to credential repositories, for example, creating a rule that proxy credentials can only be created for a TPM-enabled platform.

7 Next steps

Attestation, as currently specified, provides a binary result which may not be feasible in full generality — this has been identified by several authors. Yoshihama et al. [51] propose a web service architecture to support and interpret

⁶A supporting infrastructure required to verify TPM signatures is typically assumed. More information can be found in [46].

⁷The Daonity system operates with Globus Toolkit 4, is open source and is available at www.hpl.hp.com/personal/Wenbo_Mao/daonity

attestation measurements (which may or may not be due to TC). An alternative approach is taken by Shi et al. [39] who propose the Binding Instructions aNd Data (BIND) system, which is designed for fine-grained remote attestation of selected code that is important to the verifier. Finally, we mention the work by Sadeghi and Stübke [38] and Haldar, Chandra and Franz [19] who argue that basing attestation on the ‘properties’ or semantics of a system is more intuitive than looking at specific hardware and software configurations.

A concern that has been echoed by Löhr et al. [25] and others is an issue common to authentication and access control mechanisms — the high assurance that attestation provides is up to the point in time that attestation occurs, with no guarantees about what happens after attestation. Mao et al. [26] suggest performing remote attestation both before and after process execution to detect if a system has changed, but perhaps there may be other, preventative methods.

Haldar et al. [19] also propose extending platform integrity measuring to virtual machines. The inclusion of virtualisation in this collection of capabilities is crucial: modern software stacks are sufficiently complex that we cannot hope to attest every aspect of their code, still less their configuration, in a changing environment. The code required for a particular task, or Grid computation, however, is sufficiently simple that we can conceive of being able to construct a purpose-built virtual machine for the particular job, and verify when it is launched that it has not been tampered with. In such a situation, the VMM becomes highly trusted, and we must therefore make it as simple as possible, to reduce its vulnerability potential. In consequence, a Grid architecture to make full use of such capabilities requires significant redesign from existing solutions.

To these three major areas of work — strengthened central services, DRM, strongly

isolated and attested execution environments — we may add many more mixed modes and intermediate steps in developing trusted Grid components. Simple identity management for hosts and Grid software (BOINC, Condor, etc.) will be valuable in its own right — work in peer-to-peer networks may be relevant here (for example, see [3]). Also useful will be the ability to use virtualisation to isolate the one piece of functionality from another — so that a BOINC virtual machine may be distributed to volunteer participants and entirely isolate them from the task they are participating in, and also isolate that task entirely from the desktop user’s work.

8 Conclusion

We have demonstrated that there are some substantial security properties missing from current Grid technologies, and shown how these can be an impediment to the take-up of such ideas. The powerful abstraction of the Grid idea — that users may not know where their data is stored, nor where their computation has been run — is at once a great strength but also a very significant security challenge. The resource owners’ dual problem — wanting to offer a service, but not to be harmed by the users’ choice of software and data — is also of interest, but better addressed by existing technologies.

The paper has attempted to summarise some of the trust and security requirements which emerge from known examples, and to point towards ways in which trusted computing technologies are beginning to address those requirements. There is a clear danger that in adopting new technologies we shall simply move the problem, since, for example, keys still need to be employed for signing: the attacker’s task changes from needing to extract the key to trying to persuade the tool to sign something

when it should not.

Nevertheless, insofar as Trusted Computing appears poised to make a substantial contribution to improved security in a variety of internet-based scenarios, it seems that new technologies — if they are employed wisely — have much potential to extend and expand the model of grid and utility computing. The improvements come not merely from improved security but, as a result, by facilitating architectures and modes of working which have simply not been possible in the past.

References

- [1] M. Allen. Do-it-yourself climate prediction. *Nature*, 401(6754):642, Oct 1999.
- [2] D. P. Anderson. BOINC: A system for public-resource computing and storage. In *Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing (GRID'04), Pittsburgh, USA, November 8, 2004*, pages 4–10. IEEE Computer Society, Nov 2004.
- [3] S. Balfe, A. D. Lakhani, and K. G. Patterson. Trusted Computing: Providing security for peer-to-peer networks. In G. Caronni et al., editor, *Proceedings of the Fifth International Conference on Peer-to-Peer Computing (P2P '05), Konstanz, Germany, August 31-September 2, 2005*, pages 117–124. IEEE Computer Society, Aug-Sep 2005.
- [4] J. Basney, M. Humphrey, and V. Welch. The MyProxy online credential repository. *Software: Practice and Experience*, 35(9):801–816, Jul 2005.
- [5] J. Basney, W. Nejdl, D. Olmedilla, V. Welch, and M. Winslett. Negotiating trust on the Grid. In C. Goble, C. Kesselman, and Y. Sure, editors, *Proceedings of Seminar Seminar 05271— Semantic Grid: The Convergence of Technologies, Dagstuhl, Germany, July 3-8, 2005*, pages 50–60. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, Jul 2005.
- [6] D. Chadwick. Authorisation in Grid Computing. *Information Security Technical Report*, 10(1):33–40, Jan 2005.
- [7] D. Chadwick, A. Otenko, and E. Ball. Role-based access control with x.509 attribute certificates. *IEEE Internet Computing*, 7(2):62–69, March 2003.
- [8] A. Cooper and A. Martin. Towards a secure, tamper-proof grid platform. In *Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid, Singapore, May 2006*, pages 373–380. IEEE Press, May 2006.
- [9] A. Cooper and A. Martin. Towards an open, trusted digital rights management platform. In *Proceedings of the ACM workshop on Digital rights management (DRM '06), Alexandria, Virginia, USA, October 30, 2006*, pages 79–88. ACM Press, Oct 2006.
- [10] K. Czajkowski, I. Foster, and C. Kesselman. Resource and service management. In Ian Foster and Carl Kesselman, editors, *The Grid 2: Blueprint for a New Computing Infrastructure*, chapter 18, pages 259–283. Morgan Kaufmann Publishers, 2nd edition, 2004.
- [11] T. Dierks and C. Allen. The TLS protocol version 1.0. Rfc 2246, Internet and Engineering Task Force, Jan 1999.
- [12] I. Foster. Globus Toolkit version 4: Software for service-oriented systems. In *Pro-*

- ceedings of the IFIP International Conference on Network and Parallel Computing (NPC 2006), Tokyo, Japan, October 2-4, 2006*, pages 2–13. Springer-Verlag (LNCS 3779), Oct 2006.
- [13] I. Foster and N. T. Karonis. A Grid-enabled MPI: Message passing in heterogeneous distributed computing systems. In *Proceedings of the 1998 ACM/IEEE conference on Supercomputing (CDROM), Orlando, FL, USA, November, 1998*, pages 1–11. IEEE Computer Society, Nov 1998.
- [14] I. Foster and C. Kesselman. *The Grid 2: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, San Francisco, 2nd edition, 2004.
- [15] 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.
- [16] I. Foster, H. Kishimoto, and A. Savva et al. *The Open Grid Services Architecture, Version 1.5*. Open Grid Forum, Jul 2006.
- [17] Ian Foster, Carl Kesselman, Gene Tsudik, and Steven Tuecke. A security architecture for computational grids. In *Proceedings of the 5th ACM conference on Computer and Communications Security, San Francisco, California, United States, November 02–05*, pages 83–92, New York, Nov 1998. ACM Press.
- [18] D. Grawrock. *The Intel safer computing initiative: building blocks for trusted computing*. Intel Press, 2006.
- [19] V. Haldar, D. Chandra, and M. Franz. Semantic remote attestation — A virtual machine directed approach to Trusted Computing. In *Proceedings of the 3rd USENIX Virtual Machine Research & Technology Symposium (VM '04), San Jose, CA, USA, May 6-7, 2004*, pages 29–41. USENIX, May 2004.
- [20] J. Hollingsworth and B. Tierney. Instrumentation and monitoring. In Ian Foster and Carl Kesselman, editors, *The Grid 2: Blueprint for a New Computing Infrastructure*, chapter 20, pages 319–351. Morgan Kaufmann Publishers, 2nd edition, 2004.
- [21] J. G. Jensen. *UK e-Science Certification Authority Certificate Policy and Certification Practices Statement*. Rutherford Appleton Laboratory, 0.7 edition, Jul 2002.
- [22] N. T. Karonis, B. Toonen, and I. Foster. MPICH-G2: A Grid-enabled implementation of the Message Passing Interface. *Journal of Parallel and Distributed Computing: Special Issue on Computational Grids*, 63(5):551–563, May 2003.
- [23] Christopher G. Knight, Sylvia H. E. Knight, Neil Massey, Tolu Aina, Carl Christensen, Dave J. Frame, Jamie A. Kettleborough, Andrew Martin, Stephen Pascoe, Ben Sanderson, David A. Stainforth, and Myles R. Allen. Association of parameter, software and hardware variation with large scale behavior across 57,000 climate models. In *To appear in Proceedings of the National Academy of Sciences of the United States of America*. National Academy of Sciences, 2007.
- [24] I. Krsul, A. Ganguly, J. Zhang, J. A. B. Fortes, and R. J. Figueiredo. VM-Plants: Providing and managing virtual machine execution environments for grid computing. In *Proceedings of the 2004*

- ACM/IEEE conference on Supercomputing, Pittsburgh, PA, USA, November 6-12, 2004*, page 7. IEEE Computer Society, Nov 2004.
- [25] H. Löhr, H. V. Ramasamy, A.-R. Sadeghi, S. Schulz, M. Schunter, and C. Stübke. Enhancing Grid security using trusted virtualization. In *Proceedings of the 1st Benelux Workshop on Information and System Security (WISSEC '06), Antwerpen, Belgium, November 8-9, 2006*. Computer Security and Industrial Cryptography (COSIC), K.U. Leuven, ESAT/SCD, Nov 2006.
- [26] W. Mao, A. Martin, H. Jin, and H. Zhang. Innovations for grid security from trusted computing — protocol solutions to sharing of security resource. In *Proceedings of the 14th International Workshop on Security Protocols, Cambridge, UK, March 2006, to appear*. Springer-Verlag LNCS, March 2006.
- [27] W. Mao, F. Yan, and C. Chen. Daonity — Grid security with behaviour conformity from trusted computing. In *Proceedings of the first ACM workshop on Scalable Trusted Computing, Alexandria, Virginia, US, November 2006*, pages 43–46. ACM Press, Nov 2006.
- [28] J. Marchesini and S. W. Smith. SHEMP: Secure hardware enhanced MyProxy. In *Proceedings of the Third Annual Conference on Privacy, Security and Trust (PST '05), New Brunswick, Canada, October 12-14, 2005*, Oct 2005.
- [29] A. Martin, T. Aina, C. Christensen, J. Kettleborough, and D. Stainforth. On two kinds of public-resource distributed computing. In *Proceedings of the UK e-Science All Hands Meeting 2005 (AHM '05), Nottingham, UK, September 19-22, 2005*, pages 931–936. EPSRC, Sep 2005.
- [30] C. J. Mitchell. *Trusted Computing*, volume 6 of *IEE Professional Applications of Computing*. IEE Press, London, 1st edition, 2005.
- [31] P. C. Moore, W. R. Johnson, and R. J. Detry. Adapting globus and kerberos for a secure ASCII grid. In *Proceedings of the 2001 ACM/IEEE conference on Supercomputing (CDROM), Denver, Colorado, USA, November 10-16, 2001*, pages 21–21. ACM Press, Nov 2001.
- [32] L. Moreau, S. Chapman, A. Schreiber, R. Hempel, O. Rana, L. Varga, U. Cortes, and S. Willmott. Provenance-based trust for Grid Computing: Position paper. Technical report, University of Southampton, May 2004.
- [33] J. Novotny, S. Tuecke, and V. Welch. An online credential repository for the Grid: MyProxy. In *Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10), San Francisco, CA, USA, August 7-9, 2001*, pages 104–111. IEEE Computer Society, Aug 2001.
- [34] Oasis Standards. *WS-Trust 1.3*, March 2007.
- [35] J. Patel, W. T. L. Teacy, N. R. Jennings, and M. Luck. Monitoring, policing and trust for grid-based virtual organisations. In *Proceedings of the UK e-Science All Hands Meeting 2005 (AHM '05), Nottingham, UK, September 19-22, 2005*, pages 891–898. EPSRC, Sep 2005.
- [36] L. Pearlman, V. Welch, I. Foster, C. Kesselman, and S. Tuecke. A community authorization service for group

- collaboration. In *Proceedings of the 3rd International Workshop on Policies for Distributed Systems and Networks (POLICY'02)*, Monterey, CA, USA, June 5-7, 2002, pages 50–60. IEEE Computer Society, Jun 2002.
- [37] S. Rajbhandari, I. Wootten, A. S. Ali, and O. F. Rana. Evaluating provenance-based trust for scientific workflows. In *Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid, Singapore, May 2006*, pages 365–372. IEEE Press, May 2006.
- [38] A.-R. Sadeghi and C. Stübke. Property-based attestation for computing platforms: Caring about properties, not mechanisms. In *Proceedings of the 2004 Workshop on New Security Paradigms (NSPW '04)*, Nova Scotia, Canada, September 20-23, 2004, pages 67–77. ACM Press, Sep 2004.
- [39] E. Shi, A. Perrig, and L. V. Doorn. BIND: A fine-grained attestation service for secure distributed systems. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy, Oakland, CA, USA, May 8-11, 2005*, pages 154–168. IEEE Press, May 2005.
- [40] D. Stainforth, A. Martin, A. Simpson, C. Christensen, J. Kettleborough, T. Aina, and M. Allen. Security principles for public-resource modeling research. In *Proceedings of the 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'04)*, Modena, Italy, June 14-16, 2004, pages 319–324. IEEE Computer Society, Nov 2004.
- [41] D. A. Stainforth, T. Aina, C. Christensen, M. Collins, N. Faull, D. J. Frame, J. A. Kettleborough, S. Knight, A. Martin, J. M. Murphy, C. Piani, D. Sexton, L. A. Smith, R. A. Spicer, A. J. Thorpe, and M. R. Allen. Uncertainty in the predictions of the climate response to rising levels of greenhouse gases. *Nature*, 433(7024):403–406, Jan 2005.
- [42] D. Thain, T. Tannenbaum, and M. Livny. Distributed computing in practice: the condor experience. *Concurrency and Computation: Practice and Experience*, 17(2-4):323–356, Feb 2005.
- [43] Thompson and Olson et al. *CA-based Trust Model for Grid Authentication and Identity Delegation*. Open Grid Forum, Oct 2002.
- [44] M. R. Thompson, A. Essiari, and S. Mudumbai. Certificate-based authorization policy in a PKI environment. *ACM Transactions on Information and System Security*, 6(4):566–588, Nov 2003.
- [45] Trusted Computing Group. *TPM Main Part 1 Design Principles Specification Version 1.2 Revision 94*, March 2006.
- [46] Trusted Computing Group. *TCG Specification Architecture Overview Revision 1.3*, March 2007.
- [47] S. Tuecke, V. Welch, D. Engert, L. Perlman, and M. Thompson. Internet X.509 public key infrastructure (PKI) proxy certificate profile. Rfc 3820, Internet and Engineering Task Force, June 2004.
- [48] S. Venugopal, R. Buyya, and K. Ramamohanarao. A taxonomy of Data Grids for distributed data sharing, management, and processing. *ACM Computing Surveys*, 38(1):1–53, Mar 2006.
- [49] V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor,

- C. Kesselman, S. Meder, L. Pearlman, and S. Tuecke. Security for grid services. In *Proceedings of the Twelfth International Symposium on High Performance Distributed Computing (HPDC-12), Seattle, Washington, June 22-24, 2003*, pages 48–57. IEEE Press, Jun 2003.
- [50] W. Xu, D. Chadwick, and S. Otenko. A PKI-based secure audit web service. In *Proceedings of the IASTED International Conference on Communication, Network and Information Security (CNIS '05), Phoenix, USA, November 14-16, 2005*. IASTED, Nov 2005.
- [51] S. Yoshihama, T. Ebringer, M. Nakamura, S. Munetoh, and H. Maruyama. WS-Attestation: Efficient and fine-grained remote attestation on web services. In *Proceedings of the IEEE International Conference on Web Services (ICWS'05), Orlando, FL, USA, July 11-15, 2005*, pages 743–750. IEEE Computer Society, Jul 2005.

portal which includes links to some of the proposals and applications mentioned in this paper.

Acknowledgments

The ideas described here have arisen from discussions with a broad range of collaborators, not least among them Boris Balacheff, Shane Balfe, Andy Cooper, Wenbo Mao, Allan Tomlinson and David Wallom. Any good ideas here are probably due to them; any bad ideas remaining are, of course, our own.

The second author is being funded by the Engineering and Physical Sciences Research Council (EPSRC) UK e-Science programme of research (EP/D053269). The site www.distributedtrust.org contains more details of this project.

For more information on the use of Trusted Computing in Grid Computing, see www.trustedgridcomputing.org — this is a