Daonity—Grid Security with Behaviour Conformity from Trusted Computing*

Wenbo Mao
Hewlett-Packard Labs, China
Beijing 100022, China
wenbo.mao@hp.com

Fei Yan
Computer School
Wuhan University
Wuhan 430072, China

Chunrun Chen
College of Computer Science
Huazhong University of
Science and Technology
Wuhan 430074, China

ABSTRACT

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

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

1. COMPUTATIONAL GRID

A computational grid [1, 2, 3] can be regarded as a next generation distributed computing system comprising a number—possibly large—of physically separated resources, each subject to their own various security, management and usage policies, to combine to a federated computing environment called *virtual organisation* (VO). The name "grid" follows analogously from tapping electricity supplied by the power grid, meaning that computational resources nowadays can and should also be tapped from super computers and data centres *elsewhere*. Early versions of computational grids were more or less confined to a high performance computing setting in which a grid VO comprises of one user plus a number of computational resource providers and/or data centres. Grid computing has now evolved to a more general setting of federated computing which supports sharing of resources and data not only for high performance computing but also involving science collaborations [1]. In the general federated computing model, a VO of principals who are (may be plural number of) users, computing platforms or devices may be working on a number of common tasks and therefore having similar requirements on resource utilities.

*A full version of this paper and a systematic description of the Daonity system plus the open source code are available at `forge.gridforum.org/projects/tc-rg/daonity/`

2. GRID SECURITY: THREAT MODEL AND CURRENT PRACTICE

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

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

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

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

Conformable VO policy However ad hoc a VO may be, it still needs conformable policy. For example, a VO policy may be that, any participant must not be able to disseminate certain VO owned data outside the VO. The difficulty here is the *conformity* of the policy to be maintained despite the ad hoc nature of the VO. For example, even with little control over its members, a VO must still be able to remove a member without letting VO data be taken away.

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

Thus, to protect a user's interest on a platform which maybe beyond the user's organisational control is the distinct nature of grid

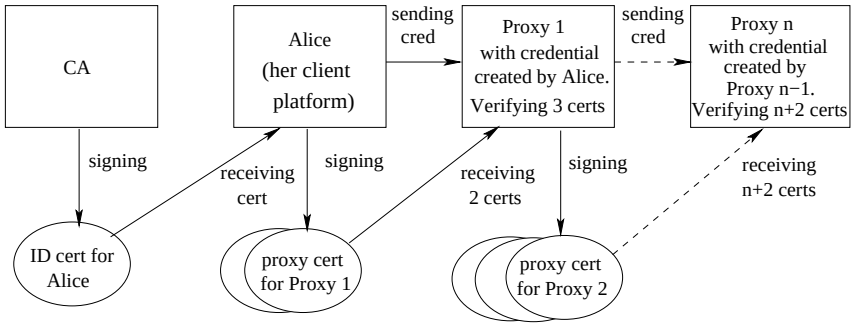

Figure 1: A typical VO construction in GSI

security. We can summarise here a threat model for grid security.

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

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

Existing and mainstream grid security practice, in fact, mainly that supported by Grid Security Infrastructure (GSI) [7] for a standard grid middleware Globus Toolkit [5], is essentially a result of direct applications of the standard public-key authentication infrastructure (PKI). Fig 1 depicts a typical VO structure in GSI. This VO is initiated by a user Alice who is assumed to have an identity certificate issued by a system-wide known grid certification authority (CA). Alice creates the VO by recruiting a member (named Proxy 1 in Fig 1). Further enlargement of the VO, if necessary, is proxy-authorised to be carried out by Proxy 1 (i.e., without Alice's involvement), and likewise with respect to subsequent proxies until the VO becomes sufficiently large (e.g., with $n + 1$ members in the case of Fig 1). In order for the enlargement to be performed in a streamline fashion without complex interactions among many members, GSI applies PKI to form a proxy certification chain: Alice creates a key pair and certifies the public part for Proxy 1 who in turn creates a key pair and certifies the public part for Proxy 2 (recruited by Proxy 1), ..., and so on. This way, a new member can verify, without interaction with Alice, that it is indeed Alice who has authorised the organisation of the VO.

The implied trust model in the direct application of PKI for the VO in Fig 1 is the following. An unknown principal will be deemed trustworthy if it has been introduced by a trusted third party (TTP). It is hoped that the introduced principal will behave in a responsible manner since it should try its *best effort* to honor the introduction of the TTP. Note, however, this is a hope. We remark that in this introduction based trust model a TTP is usually positioned *outside* the system of partners. For example, if a protocol involves Alice and Bob who needs a TTP's service, the TTP is usually not an active or inline participant in the protocol; in particular, the TTP is usually not placed inside the platforms of the protocol participants. Unfortunately, the introduction based trust model actually does not suit grid security (including the typical case in Fig 1) very well. Clearly, for grid security facing partner-and-adversary threats, Alice can have little control whether or not the proxy credentials will be misused. In order to mitigate the potential loss or misuse of the proxy credentials, GSI stipulates a policy that a proxy credential has a short lifetime of 12 hours. This is obviously a rather coarse policy and greatly limits the power of grid computing. We can say that the VO constructed in Fig 1 is only suitable for a collegial environment in which partners are colleagues or friends alike.

Then what is exactly a desirable security mechanism we need for a computing environment with a partner-and-adversary threat model? We will need to place a TTP *right inside* the computing platform owned by the participant to protect the interest of the other participant(s).

3. SOLUTION: TRUSTED COMPUTING

We consider that *Trusted Computing* (TC) technology [12] developed by Trusted Computing Group (TCG) forms a practical and readily available technical means to serving our need for countering partner-and-adversary threats in grid security. TCG is an important industrial initiative for improving computer security by means of a hardware supported security architecture. TCG uses a tamper-protection hardware module called *Trusted Platform Module* (TPM) which is integrated into a computing platform. With the tamper-protection property of the TPM, TCG in fact assumes a platform owner a potential adversary with respect to the rule of a federated computing system in which the platform is involved, and tries to prevent this party from by-passing or breaching the rule. In contrast to the conventional security mechanisms against external, or less privileged, adversaries, the owner of a platform usually is in a privileged position, i.e., a stronger adversary and thereby it is harder to prevent it from wrongdoing.

With hardware protected cryptographic capabilities, the TPM which is integrated into a computing platform is effectively an in-platform TTP which is there to protect the rule of fair play for all participants, whether the owner of the platform or a guest user. For a federated computing system, the TCG technology can not only improve security in a conventional sense (because of the enhancement with hardware) such as strong protection on cryptographic key material, but also with more innovation to allow conformed behaviour of a platform and the owner/users to be measured by the rest of the participants in the federated computing system.

4. DAONITY IN ONE USECASE: A CONFORMED POLICY FOR A VO

Daonity is a research and standard development project in Open Grid Forum (OGF) [13] to develop TC based solutions to strengthen GSI. Planned as a contribution to the open standard, the Daonity system will be in open source as a component for the ever evolving Globus Toolkit. In the time of writing this extended abstract we have completed the design, specification and implementation for the first version of the system. We plan to make the first release of Daonity to OGF in September 2006.

We will report in a full paper a number of TCG innovations which form Daonity's contributions to improving GSI. Here in this

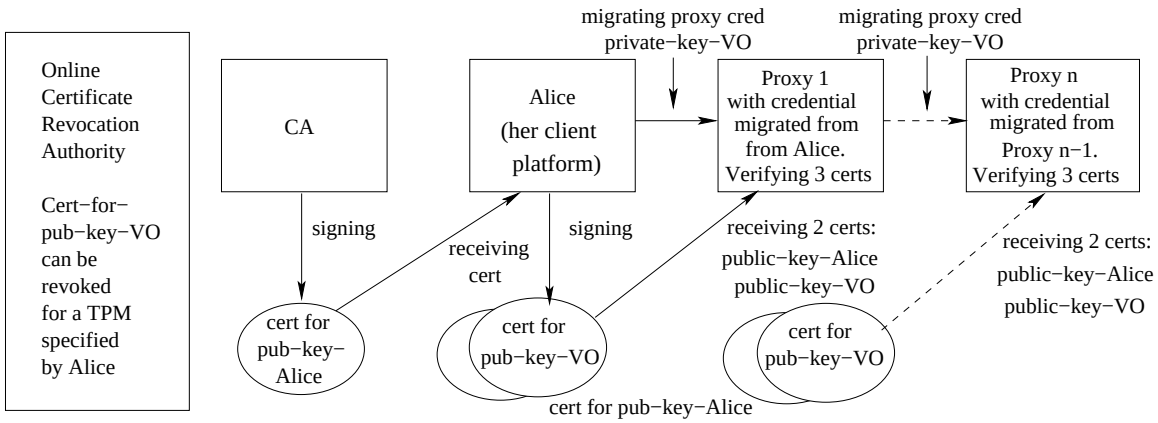

Figure 2: A VO in Daonity created using TCG credential migration

extended abstract we shall only describe one scenario of the TCG enabled security solution to grid security. This is a grid VO with conformed behaviour. This VO is dynamic in its lifecycle of creation, enlargement and shrinkage. This VO must maintain a what we shall refer to as "no-dissemination" policy (to be specified in a moment). This policy stipulates with sufficient generality a very desirable property for grid computing which the current GSI cannot achieve. Let us now describe this VO in three steps: system requirements and assumptions, creation of the VO and adding members to it, and its dynamic property (ease of removal of a member).

4.1 Requirements and Assumptions

In Daonity we use TPMs in the platforms of the VO participants to enforce security policy of the VO over the VO members. We have the following system requirements and assumptions.

i) The default case of our VO has the same structure as that in Fig 1. The creation of the VO is initiated by a leader (Alice) and its enlargement is also in the proxy manner, i.e., Alice authorises a proxy to recruit a new member without interacting back for her approval. Of course, if an application demands explicit approval by Alice then interactions will be needed in the recruiting time.

ii) No VO member will be able to disseminate electronically any VO data outside the VO. (To disseminate electronically means to communicate electronically using convenient tools such as email, ftp, copy to removable storage media, etc).

iii) Removal of a VO member can only be done by the VO leader and must be non-interactively. With non-interaction, refusal (non-cooperation) by a member will not cause service stoppage. A removed member must not be able to take away any VO internal data.

iv) The platform of each VO member has a TPM. Each TPM has a certified *attestation identity key* (AIK). This is a public key with the private component being always protected by the TPM. A TPM can be attested to a remote querier using a cryptographic protocol which uses AIK. For clarity in exposition, we will not explicitly specify the TCG CA, the AIK mechanism and the attestation protocol, but assume that these are all in place upon identification and acceptance of a TPM-platform.

v) The system also needs to use a Grid CA. This CA is in the position of the CA in Fig 1 and Fig 2 and known to all the prospective VO members.

vi) We will make use of a TCG standard protocol: *credential migration*. This protocol allows a *migration authority* (MA) of a cryptographic credential (a private key) in a TPM to move the credential to another TPM. We assume that, for the credential which has been migrated from Alice's TPM to other TPMs, Alice remains being the MA for that credential.

4.2 Creation and Enlargement of the VO

Fig 2 illustrates the creation and enlargement of our VO. Apart from using an "Online Certificate Revocation Authority" (its role see §4.3), the rest part of this VO is very similar to the VO in Fig 1. The VO's creation is also initiated by Alice, and each enlargement step proxy-authorised to a proxy without interacting back to Alice. Now with the use of the TCG credential migration protocol (which is a TCG standard protocol and not detailed in this abstract), the use of chained proxy certificates is avoided. A single proxy credential, which we shall name "private-key-VO," will be created in the TPM of Alice's platform and proxy-authorised to migrate to each of the TPMs of the principals participating the VO. The matching public-key-VO is certified by Alice. Now at any step of new member recruitment, the joining new member verifies only a small constant number of certificates to deem the bona-fide-ness of the request originated from Alice. With the VO credential in TPMs, a coarse policy of 12 hour VO lifetime in GSI is now unnecessary.

Once the VO is fully built, Alice (is notified with the full information of the participating members) can broadcast within the VO a session key which is encrypted using public-key-VO. Subsequent discussions within the VO can be protected using the session key.

With the VO credential residing inside TPMs, a much stronger protection on the credential is achieved. Desirable policy conformity follows as a result. In our usecase, the system will disallow any VO member to try to save a cleartext copy of the VO discussions in the persistent storage. This materialises our non-dissemination policy. It is not hard to imagine other conformable policies. Below let us manifest the dynamic property of this VO.

4.3 Dynamic-ness of the VO

In our VO usecase, a TPM has the following conformed behaviour on using the VO credential: the credential is usable in each instance of its use only after the TPM has obtained an explicit approval from the "Online Certificate Revocation Authority" (OCRA). The OCRA can be instructed by Alice to enlist a revocation of using the matching credential for public-key-VO on a given TPM. Once a revocation of usage with respect to a TPM is enlisted in OCRA, the TPM will no longer use the credential anymore. This way, Alice can remove a member out of the VO without letting it take away VO data. Notice that this removal procedure is non-

interactive: no need to obtain the consent from the member to be removed. Non-interactive removal prevents possible refusal and so the VO functions cannot be stopped by a non-cooperative party.

The need for OCRA (an online server) to be frequently checked by TPMs seems to be a downside of policy conformity in this usecase. Depending on applications, Alice may in some cases make an explicit instruction to the TPMs in the VO to avoid real-time checking with OCRA for a period of time.

5. IMPLEMENTATION STATUS

We have planned to make Daonity an open-source system. The implementation of Daonity has greatly benefited from the open-source Trusted Software Stack (TSS) package TrouSerS [11], OpenSSL library [8] and the open-source grid middleware package GT4. In fact, apart from the TPM migration component, all other TSS parts of Daonity are readily adapted and modified from TrouSerS, OpenSSL and plugged into GT4.

The TPM migration component has been completed for the TPM chip version 1.1b manufactured by Infineon Technologies AG on a number of HP platforms. At the time of writing this abstract we are about to make the first release of the open source code of the Daonity system (location see the footnote in the title page of this extended abstract). We have planned our next step of the implementation work to be making Daonity to work for TPMs of all major manufacturers. With Daonity released in open source, we hope that those who appreciate Daonity would help to spread the system onto TPMs of other vendors.

6. RELATED WORK

Apart from GSI, GT and TrouSerS which are obviously closely related works, the following two papers seem to be most relevant to the work of Daonity.

The first is on property-based attestation [10]. This paper argues that the attestation functionality proposed by the existing specification of the TCG can be misused to discriminate certain platforms and the operating systems running on the platform. Therefore the really essential element for an attestation should be properties of the software systems in the platform, not the software systems themselves. We consider that a foolproof version of the Daonity system will need to borrow the idea of property-based attestation. However researchers are now close to a consensus that the real issue with attestation is a need to work with operating systems together. We shall discuss this point further in §7.

The second is the work of SHEMP (Secure Hardware Enhancement for MyProxy) [6]. This is a system which hardens MyProxy, the on-line PKI credential management servers, using a hardware trusted computing base. A MyProxy server is an important and attractive target. Strong hardware based protection of the credentials and user passwords which are managed by a MyProxy server is very desirable. The hardware TCB proposed by SHEMP is an IBM cryptographic co-processor.

7. CONCLUSION AND FUTURE WORK

As grid security is becoming a more and more serious need, a number of problems remains un-tackled in the existing practice. We have identified that behaviour conformity is an essential requirement for grid security, or in fact for where a partner-and-adversary like threat model applies. We have described Daonity in one, yet sufficiently general, grid computing usecase: a dynamic VO with a strongly conformed policy enforced by the TCG technology. Clearly no existing grid security mechanism can achieve the conformed behaviour as can Daonity for the manifested usecase. Our usecase therefore demonstrates that Trusted Computing technology can provide suitable and practical solutions to the identified problems in the existing grid security practice.

We remark however that for the first release of Daonity we have only been working on the TSS/TPM and the grid middleware layer without touching the operating system in between. It is in fact possible for a very strong adversary, such as a privileged user, to bypass our form of behaviour conformity in the current version of Daonity, e.g., using some special tool to read a register in CPU for the session key when a VO discussion is being encrypted/decrypted.

New thoughts on virtualisation of operating systems for offering new security services and functionalities have been proposed. The consensus is to work with operating systems, e.g., [4, 9]. We plan to work on future development of Daonity along that direction. For instance, recording in the TPM "odd operations" of a privileged process seems to be a useful auditing service to deter unwanted behaviour while attestation for a "clean" TPM register should be much more meaningful than doing for a complex software system.

Acknowledgments

Greg Astfalk and Andrew Martin reviewed an early draft of this paper and provided insightful comments and suggestions. Zhidong Sheng, Jing Zhan, Liqiang Zhang, Lihua Song, Qi Sun, Miao Zhang, Lei Zhang (of Wuhan Univ.), Weizhong Qiang, Wenbo Yan and Shuming Liu (of HUST) participated in the implementation work. Wei Liu, Huanguo Zhang, Hai Jin and Boris Balacheff provided organisational, coordinative and logistic support to the Project. We would also like to thank anonymous reviewers of 13th ACM-CCS and 1st ACM-STC for important review feedbacks.

8. REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] T. Garfunkel, M. Rosenblum and D. Boneh. Flexible OS support and applications for Trusted Computing. In the 9th Hot Topics in Operating Systems (HOTOS-IX), 2003.

[5] Globus Toolkit 4. `www-unix.globus.org/toolkit/`

[6] J. Marchesini and S. Smith. SHEMP–Secure Hardware Enhancement for MyProxy. Technical Report TR2005-532, Dept of Comp. Sci. Dartmouth College, Feb 2005.

[7] Open Grid Forum. Overview of the GSI `www.globus.org/security/overview.html/`

[8] OpenSSL. `www.openssl.org/`

[9] OpenTC. Available at `www.opentc.net/`

[10] A. Sadeghi and C. Stüble. Property-based attestation for computing platforms: caring about properties, not mechanisms. New Security Paradigm Workshop, 2004.

[11] TrouSerS. The open-source TCG Software Stack. `trousers.sourceforge.net/`

[12] Trusted Computing Group. `www.trustedcomputinggroup.org`.

[13] Trusted Computing Research Group. Open Grid Forum. `forge.gridforum.org/projects/tc-rg/`