Security Issues in Virtual Grid Environments

Jose L. Muñoz, Josep Pegueroles, Jordi Forné, Oscar Esparza, and Miguel Soriano

Technical University of Catalonia (UPC)*
Telematics Engineering Department (ENTEL)
1-3 Jordi Girona, C3 08034 Barcelona (Spain)
{jose.munoz,josep,jordi.forne,oscar.esparza,soriano}@entel.upc.es

Abstract. Computational Grids (or simply Grids) enable access to a large number of resources typically including processing, memory, and storage devices. Usually, Grids are used for running very specific applications (most of them related to some kind of scientific hard problem); however, not much attention has been paid to commercial Grid applications. The massive use of such commercial services will depend on fulfilling their special security, usability and quality of service requirements. In this sense, Virtual Private Grid (VPG) provides a way of dynamically create a virtual grid environment with dedicated network resources. In this paper VPG is compared with related work such as the Grid over VPN (GoVPN), the Grid Community (GC) and the Ad-hoc Grid (AG) and the security challenges for VPGs are analyzed.

1 Introduction

A Grid is a collection of heterogeneous computing resources including processing, memory, and storage devices, all of which are geographically and organizationally dispersed. According to Foster et al. [6] "Computational Grids have emerged as an important new field, distinguished from conventional Distributed Computing by its focus on large-scale resource sharing, innovative applications, and *in some cases, high performance orientation*".

The main focus of current generation Grids is the transparent access to resources; on the contrary offering high performance services to a wide range of non-expert users has not been among the main priorities of the Grid. Current testbeds are either demo systems, with few users and little functionality, or scientific/engineering systems, with specialized users and high QoS demands. Virtual Private Grid (VPG) provides a way of dynamically create a virtual grid environment with dedicated network resources. In this paper VPG is compared with related work such as the Grid over VPN (GoVPN), the Grid Community (GC) and the Ad-hoc Grid (AG) and the security challenges for VPGs are analyzed.

The rest of the paper is organized as follows. Section 2 provides an overview of the related work. Section 3 describes the VPG security architecture requirements and gives some hints on managing privacy, a new relevant requirement for VPGs. Finally, we conclude in Section 4.

* This work has been supported by the Spanish Research Council under the projects DISQET (TIC2002-00818) and ARPA (TIC2003-08184-C02-02).

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3038, pp. 174–178, 2004.

2 Related Work

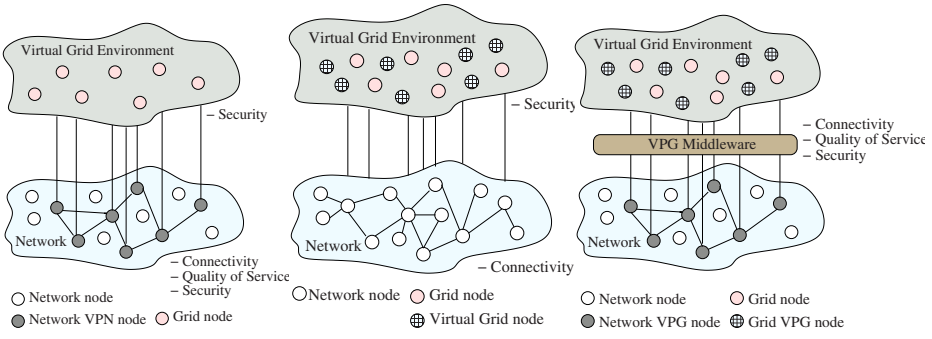
Grid over VPN (GoVPN) is a possible solution for building a virtual grid environment with dedicated resources. In GoVPN, first a VPN is created and then a Grid is deployed over it. The private environment is created at the network level so that users outside this environment do not have network connectivity to the resources. There are several mechanisms that VPNs use to enable private data going through a public network, the most important among them are authentication, privacy and tunneling. IPSec, TLS and SSH are ways of performing these mechanisms. On the other hand, technologies such as RSVP and DiffServ may be used to warrant QoS to VPN tunnels. Some practical approaches of GoVPN are [9], where the authors use SSH for building a "shell-like Grid application", and [4,3], where it is proposed a "virtual private grid file system". GoVPN has several advantages: it can provide warranted QoS at the network level, it relies on existing technologies and it allows seamless integration of legacy applications. However, GoVPN has also some drawbacks. Allowing the Grid level to become aware of changes in the network level and vice versa is tedious because Grid and network level are independently deployed. This fact limits GoVPN to pretty static environments. Moreover, each level (Grid and network) performs its own security services, leading to either duplicated schemes and/or mismatching security policies.

Grid Community (GC) [11] and **Ad-hoc Grid (AG)** [10] both create the private grid environment using only grid-level mechanisms. In GC each site grants coarse-grained access of its resources to a community account requiring the existence of a Community Authentication Server (CAS). AG deploys a similar scheme but allows more spontaneous and short-lived collaborations. However, privacy and QoS at the network level are not addressed by any of them.

The **Virtual Private Grid (VPG)** provides a solution for building private grid environments comprising both network and grid levels. In this sense, VPG can be understood as a middleware interacting with these two layers. Hence, the middleware has to include two modules: one for managing network resources and another controlling security. Section 3 will further describe the requirements of the latter module, the VPG Security Architecture (VPG-SA). Figure 1 shows the different mechanisms for building the private grid environment with dedicated resources.

3 VPG Security Architecture

The VPG Security Architecture (VPG-SA) must provide the basis for a dependable grid operation. Among other general security requirements [5] as Authentication, Authorization, Assurance, Accounting, etc. the VPG-SA must support all the particular aspects of current generation Grids [7]. These security topics are the following: (1) The VPG must grant access to resources distributed over multiple administrative domains. (2) The inter-domain security solution used for VPG must be able to *interoperate with the local security solutions* encountered in individuals domains, enforcing the support for *multi-domain trust and policies*. (3) A user should be able to authenticate once and subsequently initiate computations that acquire resources without the need for further user authentication (*Single sign-on*). (4) Multi-domain support and single sign-on will need



(a) Grid over VPN (b) Grid Community and Adhoc Grid (c) Virtual Private Grid

Fig. 1. Technologies for building a virtual Grid

the usage of a common way of expressing the identity of a principal such as a user or a resource, that is a *Uniform credentials/certification infrastructure*.

Moreover, it must be taken into account that the VPG runs over an untrusted network and that *data privacy* must be assured at the Grid level. Usually, applications running inside a VPG require to send a computational result to a group of nodes for use during the next computation. Nevertheless, privacy management for large groups can be expensive and it can harmfully affect the VPG performance. Currently grid implementations and toolkits provide some specific security features, mainly for resource access control, user authentication, and hop-by-hop encryption of communication links. The issue of end to end privacy has not been supported in any of these works.

End to End Privacy. The use of multicast at the network layer and the data encryption in the grid level is the efficient way of managing privacy in VPG. The encryption will use a common key shared by all the nodes in the virtual grid environment. The key should be known by all nodes in the VPG and must be updated every time the group of nodes changes. This is the only way of achieving perfect forward and backward secrecy. The creation of the VPG multicast channel, the key stablishment, and the key update, have to be assumed by the VPG-SA.

VPG creation. In VPG each node is considered a member of a multicast group. When the virtual grid environment has to be created, a secure multicast group has also to be set up. A node will be considered member of the group if it knows the common secret shared by all the group (weak authentication). The VPG-SA has the responsibility of delivering the secret key to every initial member. Each member has to be previously authenticated and it is mandatory that all the VPG nodes support multicast connections. The only way of performing this action is by means of a unicast connection with each one of the initial nodes of the virtual environment. As long as this action only takes place once, bandwidth and computation issues are not very relevant.

VPG management. If only unicast connections were used, the VPG key update would use many bandwidth and computation resources. In this sense, multicast along

with logical key tree based schemes [8,1,2] can be used to improve performance of security management in the Grid. The VPG-SA must use two types of keys: Session Encryption Keys (SEK) and Key Encryption Keys (KEK). The SEK is the key that each node in the grid has to know for weak authentication and encrypting the exchanged data. KEKs are keys that are used to decrease the bandwidth requirement for VPG management. The knowledge of these keys will allow nodes in the grid to get the updated SEK, that is, a VPG-SA can exclude or include nodes in the Grid by using these keys.

Addition of a node to the VPG environment. The VPG-SA maintains a logical binary-tree structure including all the nodes in the grid. Each node of the grid is logically placed in a leaf of the tree. Each junction in the tree structure will contain a different KEK. Every node in the grid must know its own key plus all the KEKs located in the junctions from his leaf to the tree root. When a node wishes to leave the VPG all the KEKs that it knew and the SEK must be updated, so the VPG-SA has to compute the new KEKs. After that, it has to notify the changes to the remaining nodes. In order to do that, it sends a multicast message containing the updated KEKs. This message is encrypted so that each node can only decrypt the data that concerns to it. This is achieved by encrypting each new updated KEK using the not-updated key under its junction. This mechanism can be understood seeing Figure 2.

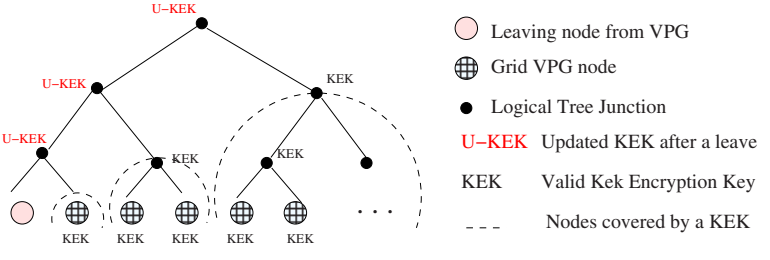


Fig. 2. Example of logical key in VPG-SA

Algorithm efficiency considerations. VPG-SA could be simpler by just updating the SEK and sending it to all the remaining nodes. This will imply as many unicast connections as remaining nodes the grid has, but it cannot be afforded neither in terms of bandwidth nor latency.

Consider a grid made of 1000 nodes. Every time a change in the composition of the grid happened, the VPG-SA would have to send 1000 messages before another node wanted to join or leave the VPG. In case this could not be achieved, a rekeying process would be aborted by the beginning of another rekeying process and the system would collapse. Using key trees, the number of messages to be sent are reduced to the order of the logarithm of the number of VPG nodes, so the VPG-SA reduces the rekeying time. A specially suitable algorithm for Grid environments was proposed by some of the authors in [12] and more detailed explanation about this topic can be found in [2].

4 Conclusions

Currently, not many attention has been paid to commercial Grid applications. The massive use of such commercial services will depend on fulfilling the special security, usability and quality of service requirements. In this sense, the VPG provides users a way of dynamically create their own Grid environment with dedicated network resources.

This paper compares VPG with GoVPN, GC, and AG. Basically, VPG operates in both grid and network levels while the others are focused in just one of them. This fact allows VPG to better solve security issues, in particular privacy which has been poorly addressed by most of the current solutions. In these sense, VPG-SA includes enhanced functionalities for efficiently creating and managing VPGs that support end to end privacy.

References

1. D. Ballenson, D. McGrew, and A. Sherman. Key Management for Large Dynamic Groups: One Way Function Trees and Amortized Initialization, 2000. Internet Draft. Work in Progress.
2. R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas. Multicast security: A taxonomy and some efficient constructions. In *INFOCOMM'99*, 1999.
3. R. Figueiredo, P. Dinda, and J. Fortes. A case for grid computing on virtual machines. In *International Conference on Distributed Computing Systems (ICDCS)*, pages 550–559. IEEE Computer Society, May 2003.
4. Renato J. Figueiredo. Vp/gfs: an architecture for virtual private grid file systems. Technical Report TR-ACIS-03-001, University of Florida, May 2003.
5. I. Foster and C. Kesselman. *The Grid. Blueprint for a new computing infrastructure*. Morgan Kaufmann, 1999.
6. I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the Grid. *International Journal of Supercomputer Applications*, 15(3):200–222, 2001.
7. Ian T. Foster, Carl Kesselman, Gene Tsudik, and Steven Tuecke. A security architecture for computational grids. In *ACM Conference on Computer and Communications Security*, pages 83–92, 1998.
8. H. Harney and E. Harder. Logical Key Hierarchy Protocol, 1999. Internet Draft. Work in Progress.
9. Kenji Kaneda, Kenjiro Taura, and Akinori Yonezawa. Virtual private grid: a command shell for utilizing hundreds of machines efficiently. *Future Generation Computer Systems*, 19(4):563–573, 2003.
10. Markus Lorch and Dennis Kafura. Supporting secure ad-hoc user collaboration in grid environments. In *Grid Computing*, volume 2536 of *LNCS*, pages 181–193. Springer-Verlag, November 2002.
11. Laura Pearlman, Von Welch, Ian Foster, and Carl Kesselman. A community authorization service for group collaboration. In *IEEE Third International Workshop on Policies for Distributed Systems and Networks*, pages 50–59, June 2002.
12. Josep Pegueroles, Wang Bin, Miguel Soriano, and Francisco Rico-Novella. Group rekeying algorithm using pseudo-random functions and modular reduction. In *Grid and Cooperative Computing GCC2003 LNCS Springer-Verlag*, 2003.