GridSec: Trusted Grid Computing with Security Binding and Self-defense Against Network Worms and DDoS Attacks*

Kai Hwang, Yu-Kwong Kwok, Shanshan Song, Min Cai Yu Chen, Ying Chen, Runfang Zhou, and Xiaosong Lou

Internet and Grid Computing Laboratory, University of Southern California, 3740 McClintock Ave., EEB 212, Los Angeles, CA 90089-2562, USA
{kaihwang, yukwong, shanshas, mincai, rzhou, cheny, chen2, xlou}@usc.edu
http://GridSec.usc.edu

Abstract. The USC GridSec project develops distributed security infrastructure and self-defense capabilities to secure wide-area networked resource sites participating in a Grid application. We report new developments in trust modeling, security-binding methodology, and defense architecture against intrusions, worms, and flooding attacks. We propose a novel architectural design of Grid security infrastructure, security binding for enhanced Grid efficiency, distributed collaborative IDS and alert correlation, DHT-based overlay networks for worm containment, and pushback of DDoS attacks. Specifically, we present a new pushback scheme for tracking attack-transit routers and for cutting malicious flows carrying DDoS attacks. We discuss challenging research issues to achieve secure Grid computing effectively in an open Internet environment.

1 Introduction

Over the last few years, a new breed of network worms like the *CodeRed, Nimda, SQL Slammer*, and *love-bug* have launched widespread attacks on the Whitehouse, CNN, Hotmail, Yahoo, Amazon, and eBay, etc. These incidents created worm epidemic [8] by which many Internet routers and user machines were pulled down in a short time period. These attacks had caused billions of dollars loss in business, government, and services. Open resource sites in information or computational Grids could well be the next wave of targets. Now more than ever, we need to provide a secure Grid computing environment over the omni-present Internet [6].

---

* The paper was presented in the *International Workshop on Grid Computing Security and Resource Management* (GSRM'05) in conjunction with the *International Conference on Computational Science* (ICCS 2005), Emory University, Atlanta, May 22-25, 2005. The research reported here was fully supported by an NSF ITR Grant 0325409. Corresponding author: Kai Hwang, USC Internet and Grid Computing Lab, EEB 212, Los Angeles, CA 90089. E-mail: kaihwang@usc.edu, Tel.: (213) 740-4470. Y.-K. Kwok participated in this project when he was a visiting associate professor at USC on sabbatical leave from HKU.

V.S. Sunderam et al. (Eds.): ICCS 2005, LNCS 3516, pp. 187 – 195, 2005.

Network-centric computing systems manifest as Grids, Intranets, clusters, P2P systems, etc. Malicious intrusions to these systems may destroy valuable hosts, network, and storage resources. Network anomalies may appear in many Internet connections for *telnet, http, ftp, smtp, Email,* and *authentication* services. These anomalies cause even more damages. Internet anomalies found in routers, gateways, and distributed hosts may hinder the acceptance of Grids, clusters, and public-resource networks [10]. Our work is meant to remove this barrier from Grid insecurity. This article reports our latest research findings in advancing security binding and building self-defense systems tailored for protecting Grid resource sites.

- Architectural design of the Grid security infrastructure in Section 2
- Security binding for trusted resource allocation in Grid job scheduling [12] in Section 3.
- The CAIDS distributed IDS and alert correlation system in Section 4
- The salient features of a DHT (*distributed hash table*) overlay [1, 13] for supporting distributed worm containment [1, 8] in Section 5
- A real-time pushback scheme to combat DDoS (*Distributed Denial of Service*) attacks [2, 3, 9] in Section 6.

2 GridSec Security Infrastructure Architecture

Our GridSec security architecture is designed to be a wide-area defense system that enables high degree of trust [7] among the Grid sites in collaborative computing over the Internet. As illustrated in Fig. 1, GridSec adopts DHT-based overlay architecture

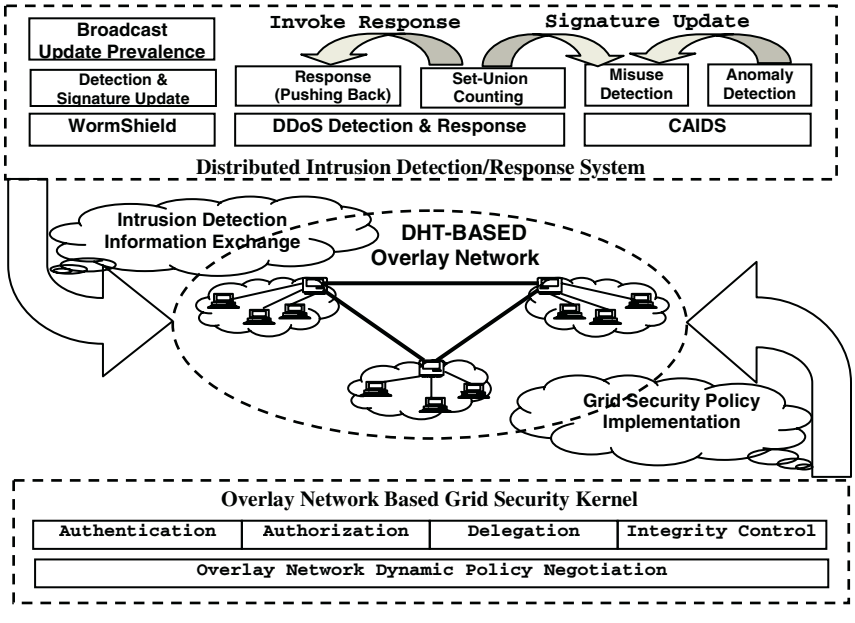


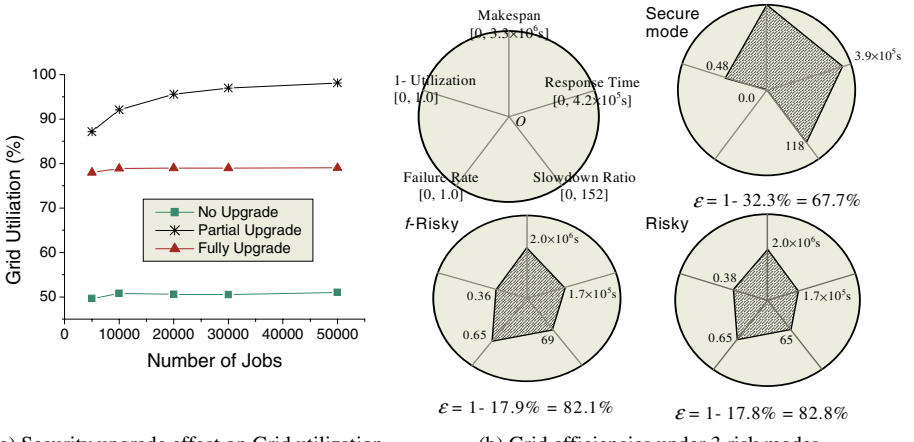
Fig. 1. GridSec infrastructure for building self-defense capabilities to protect Grid sites

as its backbone. As a virtual communication structure lay logically on top of physical networks, our overlay network maintains a robust virtual inter-networking topology. Through this topology, trusted direct application level functionalities facilitates inter-site policy negotiation and management functions such as authentication, authorization, delegation, policy exchange, malicious node control, job scheduling, resource discovery and management, etc.

The GridSec system functions as a *cooperative anomaly and intrusion detection system* (CAIDS) [6]. Intrusion information is efficiently exchanged by the overlay topology with confidentiality and integrity. Each local IDS is autonomous, and new algorithms can be added easily due to the high scalability of the overlay. Each node may work as agent for others and varies security models/policies can be implemented. As shown in Fig. 1, currently available functional blocks include the WormShield [1], CAIDS [6] and DDoS pushback scheme [2]. We are currently integrating our newly developed worm and flooding defense algorithms into the GridSec *NetShield* system .

3 Security-Binding for Trusted Resource Allocation

The *reputation* of each site is an aggregation of four major attributes: *prior job execution success rate, cumulative site utilization, job turnaround time,* and *job slowdown ratio.* These are behavioral attributes accumulated from historical performance of a site [12]. The defense capability of a resource site is attributed to *intrusion detection, firewall, anti-virus/worm,* and *attack response capabilities.* Both site reputation and defense capability jointly determine the *trust index* (TI) of a resource site. In [12], we have suggested a novel fuzzy-logic approach to generating the local trust index from the above-mentioned attributes.



(a) Security upgrade effect on Grid utilization

(b) Grid efficiencies under 3 risk modes

Fig. 2. Effects on Grid utilization and Grid efficiency by security enhancement

On the other hand, user jobs provide their *security demand* (*SD*) from resource site. A *trusted resource allocation* (TRA) scheme must satisfy a *security-assurance*

condition: *TI* ≥ *SD* during mapping jobs to resource sites. We evaluate the effectiveness of a TRA scheme by considering five performance metrics: *makespan*, *job failure rate*, *site utilization*, *response time*, and *slowdown ratio*, etc. We report in Fig. 2(a) the effects of trust integration towards total Grid utilization. *No upgrade* policy corresponds to resource allocation without trust integration. While *full upgrade* and *partial upgrade* entail a full scale and a resource-constrained trust integration, respectively. Higher Grid utilization is observed after the integration process.

This security-binding scheme is effective in mapping large-scale workloads in NAS and PSA benchmark experiments [12]. New performance metrics are developed to assess the effects of trust integration and secure allocation of trusted resources to enormous Grid jobs. Our secure binding scheme scales well with both job number and Grid size. Trusted job outsourcing makes it possible to use open Grid resources with confidence and calculated risks. We consider three risk conditions in remote job executions, namely, *conservative* mode, *f-risky* and *risky* mode representing various levels of risk the jobs may experience.

The cumulative Grid performance of these three modes is shown in Fig.2(b) by three 5-D Kiviat diagrams under 3 risk conditions. The five dimensions correspond to five performance metrics. The smaller is the shaded polygon at the center of the Kiviat diagram, the better is the *Grid efficiency,* defined by $\varepsilon = (1 - A_{shaded} / A_{circle})$. This implies that more efficient Grid has *shorter makespan* and *response time* and *lower slowdown, failure rate,* and *under-utilization rate* (1- *utilization rate*). Our NAS simulation results shows that it is more resilient for the global job scheduler to tolerate job delays introduced by calculated risky conditions, instead of resorting to job preemption, replication, or unrealistic risk-free demand.

4 Distributed Intrusion Detection/Alert Correlation

The CAIDS we built [6] can be deployed at various Grid sites to form a *distributed IDS* (DIDS) supported by alert correlation sensors. These sensors are scattered around the computing Grid. They generate a large amount of low-level alerts. These alerts are transmitted to the alert correlation modules to generate high-level intrusion reports,

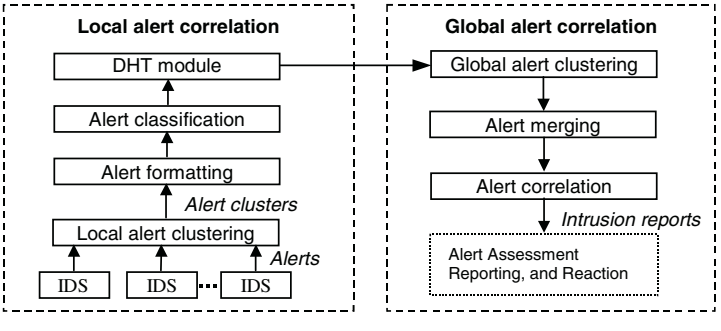


Fig. 3. Alert operations performed in local Grid sites and correlated globally

which can provide a broader detection coverage and lower false alarm rate than the localized alerts generated by single IDS. Figure 3 shows the alert operations performed by various functional modules locally and globally [4].

Similar to a major earthquake, one large attack has a series of after attacks. The global alert correlation is to detect the relationship among the attacks. We need a high-level view of attacks. The system detects the intention and behavior of attackers. An early detection report can be generated to minimize the damages. We have tested the CAIDS system at USC with an Internet trace of 23.35 millions of traffic packets, intermixed with 200 attacks from the Lincoln Lab IDS dataset.

In Fig.4, we plot the ROC curves corresponding to 4 attack classes. The detection rate grows quickly to its peak value within a small increase of false alarm rate. To achieve a total detection rate above 75% of DoS attacks, we have to tolerate 5% or more false alarms. The R2L (*root-to-local*) attacks have the second best performance. The port-scanning Probe attacks perform about the same as R2L attacks. The U2R (*user-to-root*) attacks have the lowest detection rate of 25% at 10 % false alarms, due to the stealthy nature of those attacks. When the false alarm rate exceeds 5%, all attacks reaches their saturated performance.

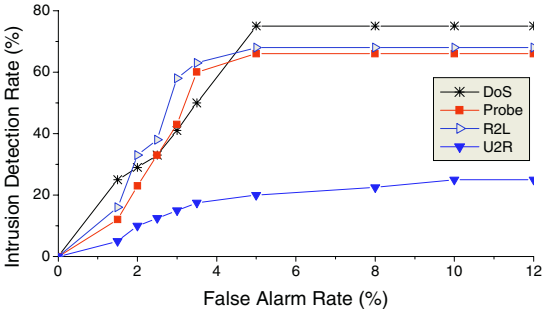


Fig. 4. Intrusion detection rate versus false alarm rate in using the CAIDS (Cooperative Anomaly and Intrusion Detection System) developed at USC [6]

5 DHT-Based Overlay for Worm Containment

We build a scalable DHT overlay to cover a large number of autonomous domains in edge networks. Our *WormShield* system [1] consists of a set of geographically distributed monitors located in multiple administrative domains (Fig.5). They are self-organize into a structured P2P overlay ring network based on the Chord algorithm [13]. Each monitor is deployed on the DMZ (*Demilitarized Zone*) of the edge network and analyzes all packets passing through it.

In *WormShield,* each monitor i remembers the set of source addresses $S(i,j)$ and the set of destination addresses $D(i,j)$ for each substring j. When the global prevalence of substring j is greater than the prevalence threshold T_p, each monitor will send their locally maintained source and destination addresses to the root monitor root j. The

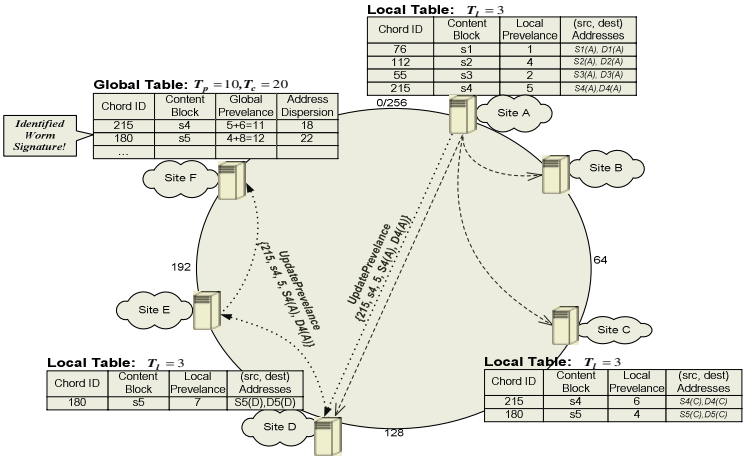


Fig. 5. An example of WormShield system with six sites for worm control

root monitor then compute the global address dispersion for the substring *j*. If *C*(*j*) is greater then an address dispersion threshold T_c, the substring *j* will be identified as a potential worm signature [11]. The root monitor will construct a multicast tree on the overlay network and disseminate the signature to all monitors participated.

For each monitor *i*, we use Rabin footprint algorithm to compute the substrings for each packet payload. Then it computes the local prevalence *L*(*i*,*j*) for each substring *j*. After a predefined interval *t* or *L*(*i*,*j*) is greater than a local prevalence threshold T_l, monitor *i* will update the global prevalence *P(j)* for substring *j* that tracks all prevalence seen in the network with *WormShield* monitors deployed. A selected monitor is assigned to maintain the global prevalence for a substring *j* using consistent hashing as in Chord [13].

6 Tracking and Pushback DDoS Attacks

We tackle two issues towards effective DDoS defense: (1) accurately identifying the ingress routers (i.e., the edge routers of the domain to be protected) that unknowingly participate in the forwarding of malicious DDoS attack flows; and (2) identifying the malicious flows and incisively cutting such flows at these *Attack-Transit Routers* (ATRs) [2].

Real-Time Traffic Matrix Tracking: We propose a low-complexity traffic monitoring technique that is based on measuring both the *packet-level* and *flow-level* traffic matrices among routers in real-time. Our proposed technique is based on accumulating very lightweight statistics for packets or flows at each router within the domain. When huge volumes of packets or flows arrive at a particular last-hop router as depicted in Fig.6, this victim router identifies the ATRs with very high accuracy using the lightweight statistics exchanged among the routers. It only requires *O*(*log log N*) storage capacity for *N* packets or flows on each router [5, 9], compared with the *O*(*N*) complexity in using a Bloom filter [2].

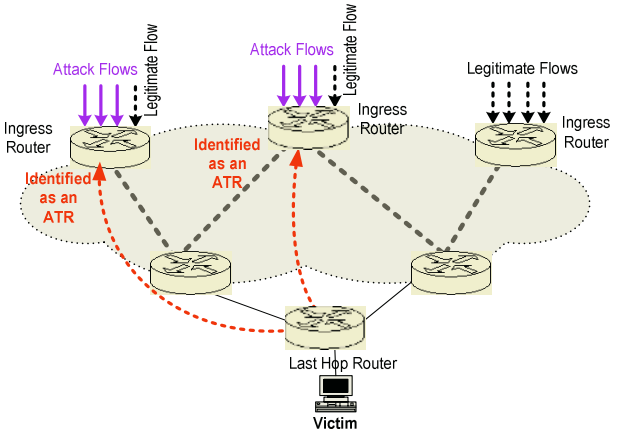


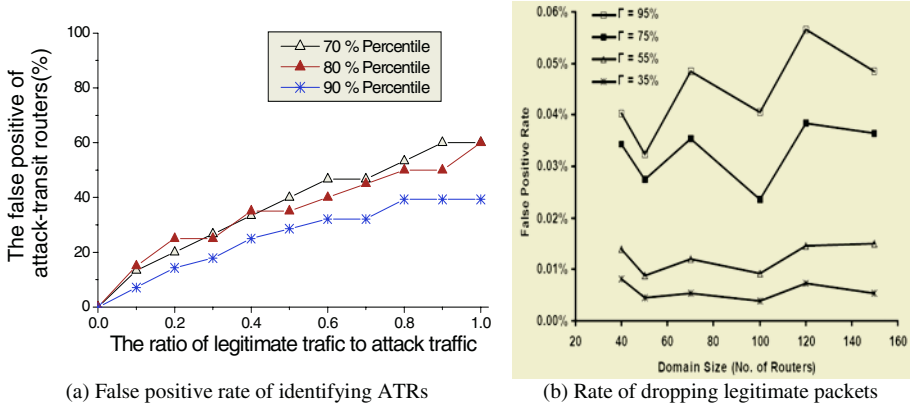
Fig. 6. The pushback scheme for identifying the attack-transit routers and blocking the malicious flows with spoofed source IP addresses

Packet- and Flow-Level Counting: Define S_i as the set of packets that enter the domain from an ingress router r_i, and D_j as the set of packets that leave the domain from an egress router r_j. To compute packet-level traffic matrix, we use the first 28-byte invariant bytes of a packet (20-byte IP header with 4 bytes masked out plus the first 12 bytes of payload). This will result in a very small collision rate. We compute the packet-level traffic matrix $\boldsymbol{A} = \{a_{ij}\}$, where $a_{ij} = |S_i \cap D_j| = |S_i| + |D_j| - |S_i \cup D_j|$ [9]. Here, we can easily compute $|S_i|$ and $|D_j|$ at each router.

For the term $|S_i \cup D_j|$, we use two probabilistic counting techniques, namely the *stochastic averaging algorithm* and *distributed max-merge algorithm* [5]. which require only O (*log log N*) storage space for N packets in the set. For flow-level traffic matrix, we use the 5-tuple {*source IP, source port, destination IP, destination port, protocol*} as the identifier for each packet. The flow-level traffic matrix $\boldsymbol{B} = \{b_{ij}\}$, where $b_{ij} = |S_i^F \cap D_j^F|$ is computed in a similar fashion. The counting complexity is $O(N)$, where N is the number of packets in a set.

Even with the ATR identification issue efficiently solved by our novel traffic tracking technique, the second issue is also a daunting challenge because IP addresses are commonly spoofed, that making correct identification of malicious flows very difficult. We propose a new MAFIC algorithm to support the adaptive packet dropping policy at the identified ATRs [3]. Through probing, MAFIC would drop malicious attack packets with very high accuracy while minimizes the loss on legitimate traffic flows [3].

Our NS-2 simulation indicates that the traffic tracking and flow cutting by dropping attacking packets are up to 90% accurate, as revealed in Fig.7(a). The scheme reduces the loss of legitimate flows to less than 5%. Figure 7(b) shows the false positive rates are quite robust and scalable under increasing domain sizes. The dup-ACK based probing is quite accurate in identifying the attack flows as the identification errors are mostly below 1%. The false alarm rates are less than 0.05% for a TCP flow rates from 35% to 95% of the total traffic volume.



(a) False positive rate of identifying ATRs (b) Rate of dropping legitimate packets

Fig. 7. The false positive rate of identifying ATRs and of dropping legitimate packets at the identified ATRs with different percentages (70%, 80%, and 90%) of traffic flows actually passing through the identified ATRs (*attack-transit routers*)

7 Conclusions

The NSF/ITR-supported GridSec project at its second year has made encouraging progress in trust management, security-driven job scheduling, trusted resource allocation, distributed IDS, collaborative alert correlation, worm containment, and distributed DDoS pushback. We offer a scalable security overlay architecture, experimental validation of distributed IDS design, and new schemes to capture network worms and pushback DDoS attacks. The GridSec system offers early warning of Internet worm spreading and launching effective pushback operations to protect Grid resources.

In the research front, we suggest several meaningful challenges for further work. The major threats come from software vulnerability and naïve users. Today's Windows, Unix and Linux variants are by no means immune from worm attacks, let alone free from DDoS flood attacks. Outbreaks must be dealt with immune response swiftly. The major research challenge lies still in the containment area. In particular, we need automated signature generation and fast suppression of malicious flows.

Internet outbreak detection and monitory are other big challenges. The reaction time, containment strategies, deployment scenarios are all yet to be worked out. We have identified the requirements of *robustness, resilience, cooperativeness, responsiveness, efficiency*, and *scalability.* The DHT-base security overlays offer a viable approach towards a fast cybersecurity solution. Of course, further advances in operating-system security, active networks, and trust management are also important.

References

[1] M. Cai, K. Hwang, Y.-K. Kwok, Y. Chen, and S. Song," Fast Conatinment of Internet Worms for Epidemic Defense using Distributed-Hashing Overlays", *IEEE Security and Privacy,* submitted July 2004 and revised March 6, 2005, to appear Nov/Dec. 2005.

[2] M. Cai, Y. K.-Kwok and K. Hwang, "Inferring Network Anomalies from Mices: A Low-Complexity Traffic Monitoring Approach", in preparation for submission to *ACM SIGCOMM Workshop on Mining Network Data*, 2005

[3] Y. Chen, Y.-K. Kwok, and K. Hwang, "MAFIC: Adaptive Packet Dropping for Cutting Malicious Flows to Pushback DDoS Attacks," *Proc. Int'l Workshop on Security in Distributed Systems* (SDCS-2005), in conjunction with ICDCS 2005, Columbus, Ohio, USA, June 2005.

[4] F. Cuppens and A. Miege, "Alert Correlation in a Cooperative Intrusion Detection Framework," *IEEE Symposium on Security and Privacy*, 2002, pp.187-200.

[5] M. Durand and P. Flajolet, "LogLog Counting of Large Cardinalities," *Proc. European Symp. on Algorithms*, 2003.

[6] K. Hwang, Y. Chen, and H. Liu, "Protecting Network-Centric Computing System from Intrusive and Anomalous Attacks," *Proc. IEEE Workshop on Security in Systems and Networks* (SSN'05), in conjunction with *IPDPS 2005*, April 8, 2005.

[7] S. Kamvar, M. Schlosser, and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks," *Proc. of WWW,* 2003.

[8] H. A. Kim and B. Karp, "Autograph: Toward Automated Distributed Worm Signature Detection," *Proc. USENIX Security Symposium*, 2004.

[9] M. Kodialam, T. V. Lakshman, and W. C. Lau, "High-speed Traffic Measurement and Analysis Methodologies and Protocols," Bell Labs Technical Memo, Aug. 2004.

[10] N. Nagaratnam, P. Janson, J. Dayka, A. Nadalin, F. Siebenlist, V. Welch, S. Tuecke, and I. Foster, "Security Architecture for Open Grid Services," http://www.ggf.org/ogsa-sec-wg

[11] S. Singh, C. Estan, G. Varghese and S. Savage, "Automated Worm Fingerprinting," *Proc. of the USENIX Symp.on Operating System Design and Implementation*, S.F., Dec. 2004.

[12] S. Song, K. Hwang, and Y.-K. Kwok, "Security Binding for Trusted Job Outsourcing in Open Computational Grids," *IEEE Trans. Parallel and Dist. Systems*, revised Dec. 2004.

[13] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, H. Balakrishnan, "Chord: A P2P Lookup Protocol for Internet Applications," *Proc. ACM SIGCOMM*, 2001.