Fuzzy Trust Integration for Security Enforcement in Grid Computing

Shanshan Song, Kai Hwang and Mikin Macwan

University of Southern California
Los Angeles, CA 90089 USA

Presented by Kai Hwang at NPC-2004
Wuhan, China, October 18, 2004

Website: http://GridSec.usc.edu/

Presentation Outline:

- Research Motivations
- Fuzzy Logic based Trust Integration
- The SARAH for Resource Optimization
  - (Secure Grid Outsourcing)
- Grid Performance with Trust Integration
- Lessons Learned and Future Research

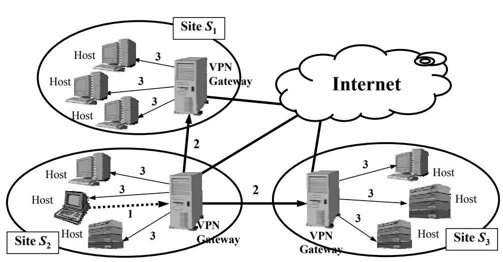
Motivations

- Grid applications demand not only resources, but also trusted resources to avoid security crashes at remote sites in an open and risky Grid environment
- Benefiting many security-sensitive Grid applications:
  - Scientific explorations, health-care
  - Public safety, cyber crime control, homeland security
  - Digital government, distance education, social community, national information services
  - Grids for business, enterprises, and e-commerce

Trusted Grid Computing Requirements

- Trusted resource allocation, sharing, and scheduling
- Secure communications among Grid sites, clusters, and protected download operations among peer machines
- Intrusion resistance, attack repelling, etc
- Fortification hardware/software (firewalls, packet filters, VPN gateways, traffic monitors, etc.)
- Self-defense toolkits/middleware (Distributed IDSs, risk assessment, response automation)

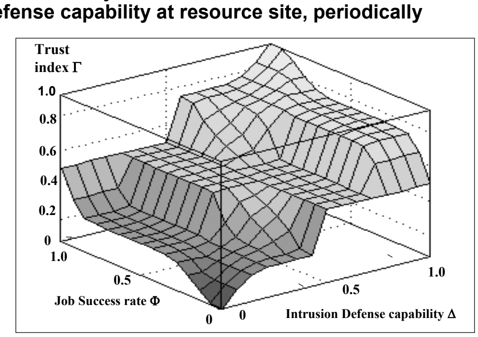
GridSec: A Grid Security Project at USC



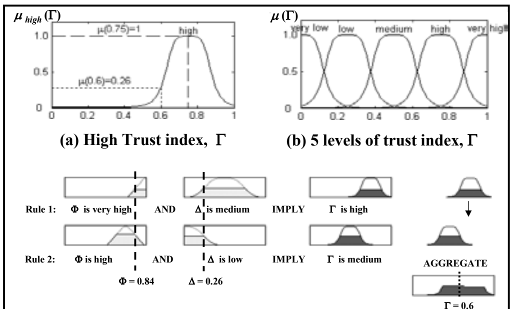
Self-defense Steps at resource site :
Step 1: ····► **Intrusion detected by host-based firewall /IDS**
Step 2: ──► **All VPN gateways are alerted with the intrusion**
Step 3: ──► **Gateways broadcast response commands to all hosts**

Trust Index of Grid Resource Site:

derived from job success rate and intrusion defense capability at resource site, periodically



Fuzzy Logic for Trust Integration

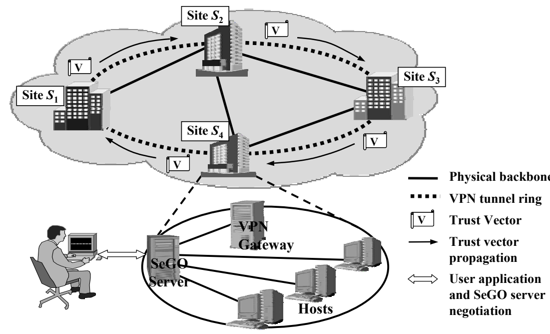


(a) High Trust index, Γ

(b) 5 levels of trust index, Γ

(c) Fuzzy inference procedure

Trust Integration over VPN Ring



A Trust Integration Example

Eq. 1 $t_{ij}^{new} = \alpha t_{ij}^{old} + (1-\alpha)s_{ij}$

Eq. 2 $\Delta(S_j) = \Delta(S_j) + \varepsilon(\Delta)$

Eq. 3 $V_j^{new} = \frac{m-1}{m} V_j^{old} + \frac{1}{m} V_i$

$$\begin{pmatrix} 0.3 & 0.2 & 0.3 & 0.4 \\ 0.5 & 0.4 & 0.5 & 0.5 \\ 0.7 & 0.7 & 0.6 & 0.7 \\ 0.8 & 0.8 & 0.9 & 0.8 \end{pmatrix}$$

(a) Initial trust matrix

$$\begin{pmatrix} 0.3 & \mathbf{0.2} & 0.3 & 0.4 \\ 0.5 & \mathbf{0.4} & 0.5 & 0.5 \\ 0.7 & \mathbf{0.8} & 0.6 & 0.7 \\ 0.8 & \mathbf{0.9} & 0.9 & 0.8 \end{pmatrix}$$

(b) Update V_2 at column 2 using Eqs. 1 and 2

$$\begin{pmatrix} \mathbf{0.4} & \mathbf{0.3} & \mathbf{0.4} & \mathbf{0.5} \\ \mathbf{0.6} & \mathbf{0.5} & \mathbf{0.6} & \mathbf{0.6} \\ 0.7 & 0.8 & 0.6 & 0.7 \\ 0.8 & 0.9 & 0.9 & 0.8 \end{pmatrix}$$

(c) Enhance $\Delta(S_1)$ and $\Delta(S_2)$ at rows 1 and 2 using Eq.3

$$\begin{pmatrix} 0.4 & 0.3 & 0.4 & \mathbf{0.5} \\ 0.6 & 0.5 & 0.6 & \mathbf{0.5} \\ 0.7 & 0.8 & 0.6 & \mathbf{0.6} \\ 0.8 & 0.9 & 0.9 & \mathbf{0.8} \end{pmatrix}$$

(d) Update V_4 at column 4 using Eqs.1 and 2

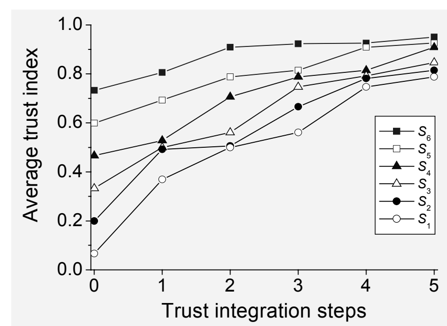
$$\begin{pmatrix} \mathbf{0.5} & \mathbf{0.4} & \mathbf{0.5} & \mathbf{0.6} \\ \mathbf{0.6} & \mathbf{0.6} & \mathbf{0.6} & \mathbf{0.6} \\ \mathbf{0.7} & \mathbf{0.8} & \mathbf{0.7} & \mathbf{0.7} \\ 0.8 & 0.9 & 0.9 & 0.8 \end{pmatrix}$$

(e) Enhance $\Delta(S_1)$, $\Delta(S_2)$ and $\Delta(S_3)$ at first 3 rows using Eq.3

$$\begin{pmatrix} \mathbf{0.6} & \mathbf{0.6} & \mathbf{0.6} & \mathbf{0.6} \\ \mathbf{0.7} & \mathbf{0.7} & \mathbf{0.7} & \mathbf{0.7} \\ \mathbf{0.8} & \mathbf{0.8} & \mathbf{0.8} & \mathbf{0.8} \\ \mathbf{0.9} & \mathbf{1.0} & \mathbf{1.0} & \mathbf{0.9} \end{pmatrix}$$

(f) Enhanced matrix after update all trust vectors

Effects of Trust Integration



Trusted Resource Allocation

- Based on the fuzzy trust model, a *Secure Grid Outsourcing* (*SeGO*) scheduler was developed for Grid resource allocation:
  - $S_i = (P_i, V_i, C_i)$, representing the *computing power*, *trust vector*, and *unit service cost*.
  - *Job* = (*W*, *D*, *T*, *B*), representing the *workload*, *execution deadline*, *minimum trust requirement*, and *budget limit.*
- The optimization process is modeled as a nonlinear programming problem, with the objective to maximize the *Grid Efficiency*:

$$E = \sum_{i=1}^{m} W_i t_{ij} \Big/ \sum_{i=1}^{m} W_i C_i = \sum_{i=1}^{m} x_i P_i L t_{ij} \Big/ \sum_{i=1}^{m} x_i P_i L C_i$$

SARAH : Security-Assured Resource Allocation arcHitecture

Algorithm 3: *SARAH* (R_j, *Job* = (W, D, T, B))

Input: Submit ***Job*** = (W, D, T, B) to resource site R_j at time τ, R_j requests resources from all m sites.

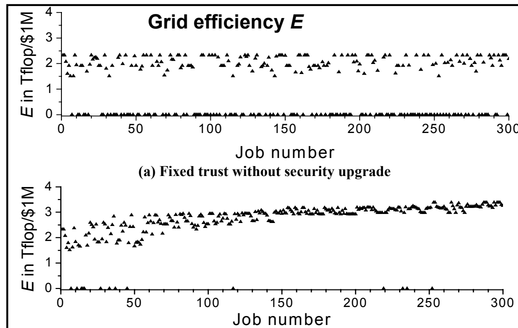
Output: Workload distribution (W_1, W_2, …, W_m) and estimated execution time L for ***Job*** based on allocation $X = (x_1, x_2, …, x_m)$ generated.

(1) R_j sends requests to obtain available resources information from all sites;
(2) **for** $i = 1$ **to** m
(3) **if** ($t_{ij} < T$) $x_i = 0$.
(4) **end for**
(5) Estimate execution time $L = D - \tau$;
(6) Generate the allocation vector $X = (x_1, x_2, …, x_m)$, which maximize $E = \sum_{i=1}^{m} x_i P_i L t_{ij} \Big/ \sum_{i=1}^{m} x_i P_i L C_i$, subject to the following constraints $\sum_{i=1}^{m} x_i P_i L \geq W$, $\sum_{i=1}^{m} x_i P_i L C_i \leq B$, and $0 \leq x_i \leq 1$;
(7) **for** $i = 1$ **to** m $W_i = x_i P_i L$;
(8) **return** (W_1, W_2, …, W_m, L) with allocation $X = (x_1, x_2, …, x_m)$.

Performance Evaluation

- In a computational Grid environment, we execute mainly coarse-grain supercomputing applications.
- The performance of the SeGO scheme was evaluated by a discrete event-driven Grid simulator developed at USC to model the trust integration and the resource optimization processes.
- We use typical workload parameters measured at USC Center for High Performance Computing and Communications. Both user jobs and resource parameters are configured to reflect real-world situations.

Trust Integration Performance Gain

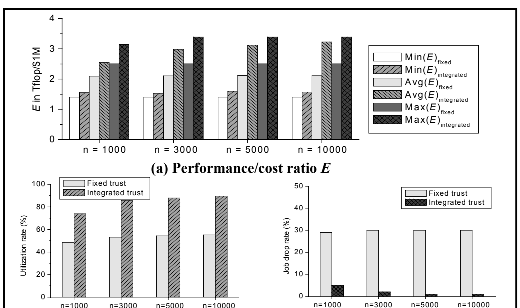


(a) Fixed trust without security upgrade

(b) Integrated trust with security enhancement

Scaling Effects of Job Number (n)

-- n = 1000, …, 10000 Jobs for Parallel Execution on 20 Grid Sites

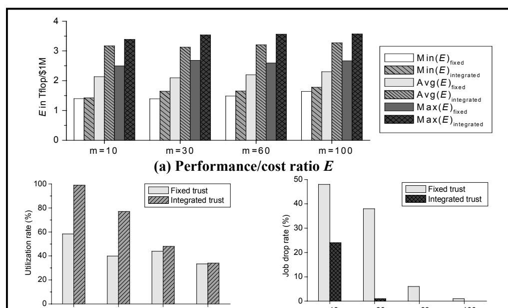


(a) Performance/cost ratio E

(b) Resource utilization rate (c) Job drop rate

Scaling Effects of Grid Size (m)

---- 5000 Jobs Parallel Execution on m = 10, 30, 60, 100 sites



(a) Performance/cost ratio E

(b) Resource utilization rate (c) Job drop rate

Conclusions:

- **Fuzzy trust integration reduces platform vulnerability: Our fuzzy trust model guides the defense deployment across distributed Grid sites. This lays the foundation of distributed security enforcement in Grids.**
- **Trusted Grid resource allocation and configuration: the SARAH has been extended into a SeGO scheduler that can be applied to upgrade the AppLeS and NimRod/G schedulers in security reinforcement.**
- **Our work will benefit many Grid applications in scientific explorations, health-care, public safety, national security, digital government, etc.**

Related Research Challenges

- **Real-life benchmark workloads such as NAS, PSA, and others are being tested on a plug-in Grid testbed scattered at USC, ISI, and several Grid sites at our collaborators in China, France, and Australia.**
- **A production SeGO scheduler will be converted from the SARAH scheme based on a mixed fuzzy trust integration and game-theoretical approach**
- **Security-driven heuristics and fast genetic algorithms developed to achieve trusted on-line job scheduling in computational Grids**