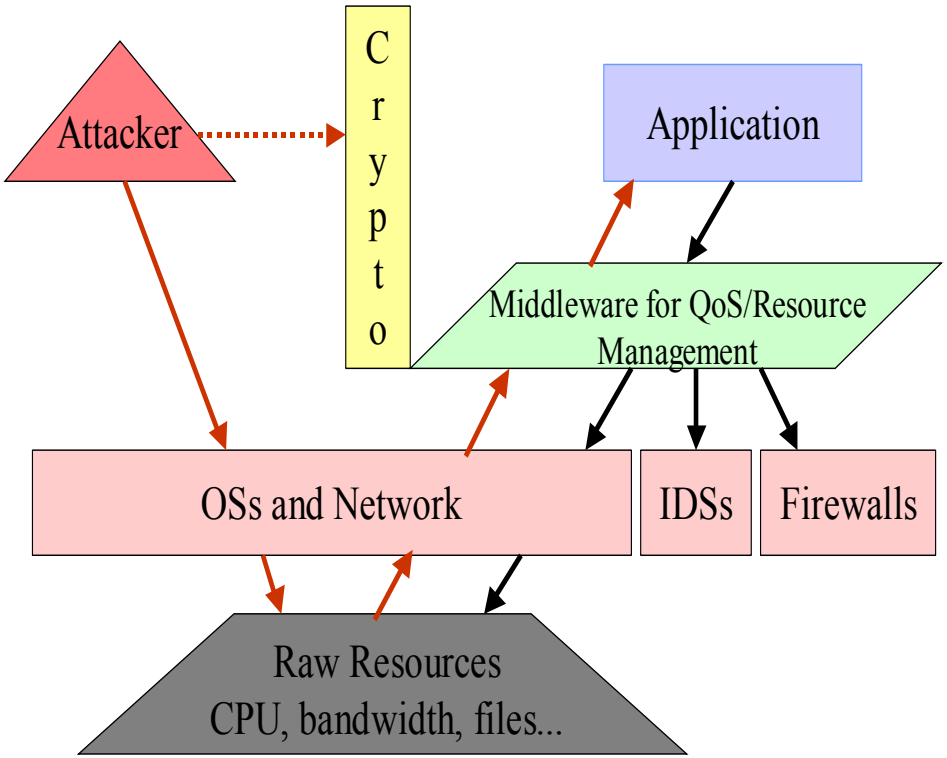
Intrusion Tolerance by Unpredictable Adaptation

Presented by

Partha Pal

William H. Sanders

Motivation



- Operational (distributed) systems are increasingly under attack
  - Gain access and gather intelligence
  - Consume/corrupt resources
- Usually coordinated and Staged
  - Attempt to disrupt ongoing/future operation

ITUA focuses on surviving attacks that are partially successful in inducing corruption at various levels

- Developing middleware-based defense using adaptation and redundancy based intrusion tolerant algorithms
- Evaluating the defense by experimenting on a realistic example (transition target)
- Developing and using validation techniques for Intrusion Tolerance solutions

Key ITUA Ideas and Presentation Outline

- Adaptive response and its management to provide a range of adaptive response aimed to stop further damage and attack progression
  - Quick local reaction to detected symptoms using rapid response loops
  - Decentralized management of redundant resources (hosts and replicated objects) using ITUA Managers
  - Support for application level policy and unpredictability using QuO adaptation control
- Use of redundancy and intrusion-tolerant (IT) algorithms to tolerate intrusion-induced corruption at multiple levels
  - Communication-level intrusion tolerance using IT GCS
  - IT inter-object interaction using IT gateways
  - IT replication management using ITUA managers

Remainder of this talk:

- Brief overview of rapid response loops, ITUA management and unpredictable and application level policy support
- Technical details of the IT Gateway and IT GCS and some initial results

The ITUA Architecture

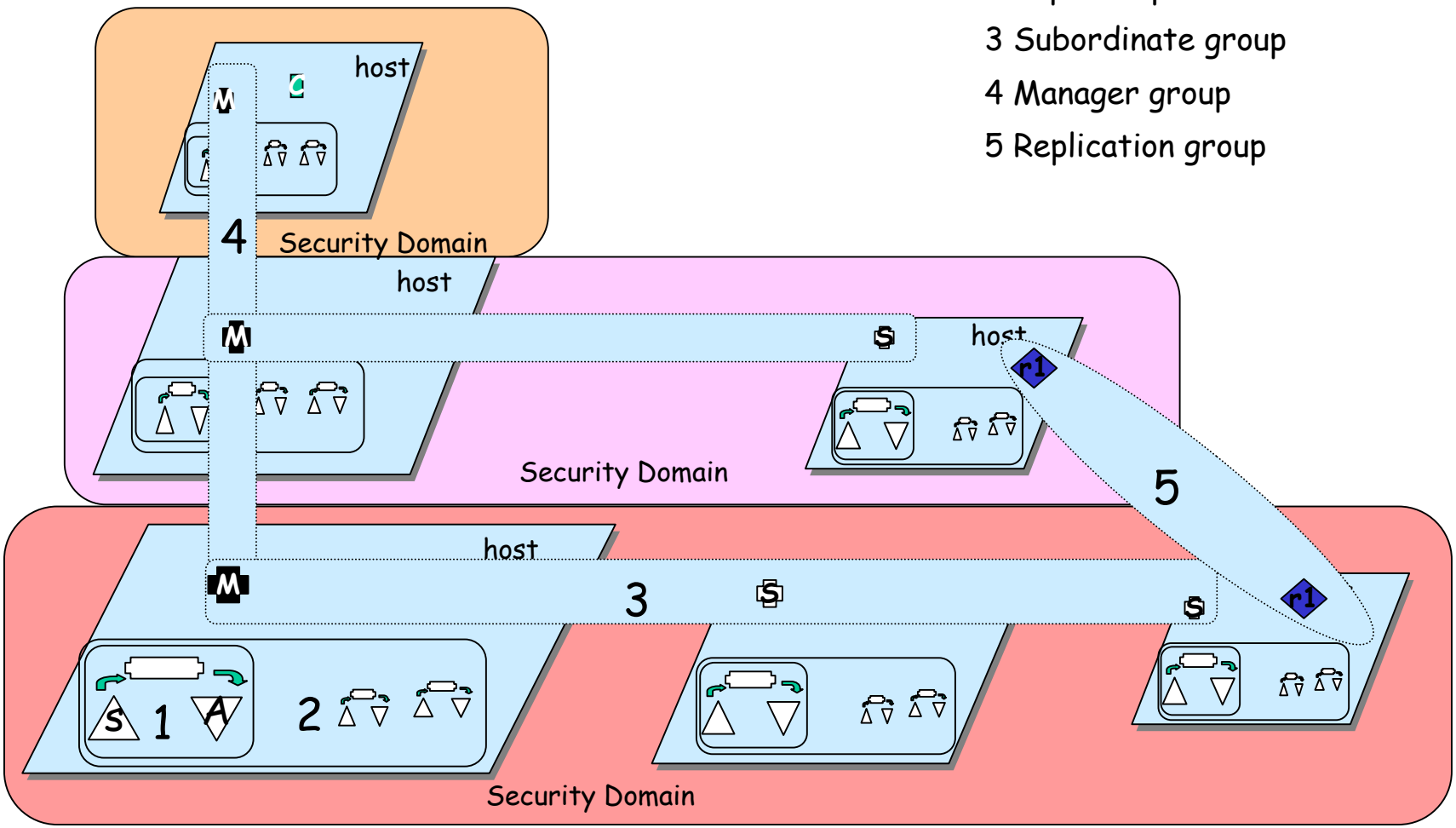
1 Sensor-actuator loop

2 Rapid response

3 Subordinate group

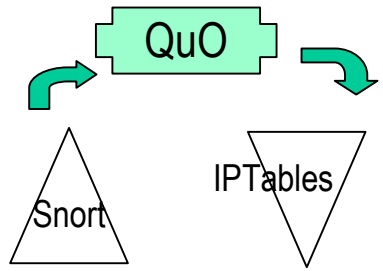
4 Manager group

5 Replication group



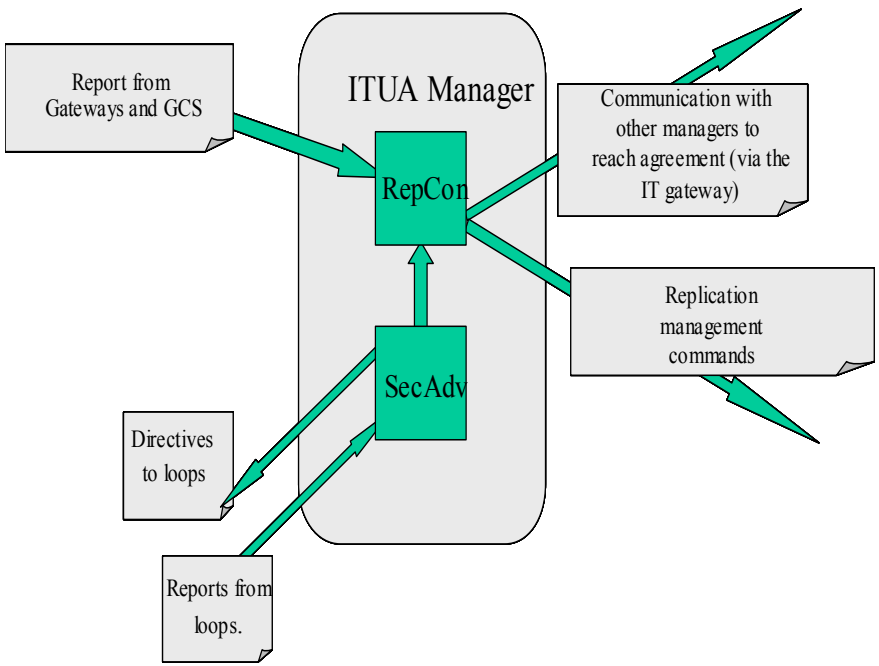
Providing Quick Local Response to Detected Symptoms

- First line of defense in the early stages of a coordinated attack
  - Quick attempt to stop further damage, to be followed by more global and coordinated response
  - Makes transfer of privilege more difficult
- Sensor-Actuator loops deployed on each host
  - "Sensor" reports incident, "actuator" attempts to address it
  - Estimate effectiveness and generate report
- Coordinated by QuO Adaptive middleware
- Two prototype " loops" implemented:
  - Snort- IPTables and Tripwire- File Restorer
- Reusable (e.g., in another context), Modular (e.g., replace a better "snort")
- Tunable parameters:
  - Agility: how frequently it evaluates attacks to compute response
  - Sensitivity: sensor's level of operation
  - Measurement Window: sliding window over attacks reported
  - Nature of response:
    - Drop, Block, Drop-Timed, Block-timed, Preset Configuration
    - Delete File , Restore File



A sensor-actuator loop

Coordinated and Global Response: Decentralized Management of Redundant Resources



Contain and eliminate intrusion induced failures and corruptions

- Decentralized is better for survival (and more complex)
- Awareness provided by loops, security advisers, gateway and GCS
- Adaptive response in the form of killing or starting a replica, abandoning a domain

Technical challenge

- Distributed decision making across security domains
  - What if a manager is corrupt?

Intrusion tolerance in the managers

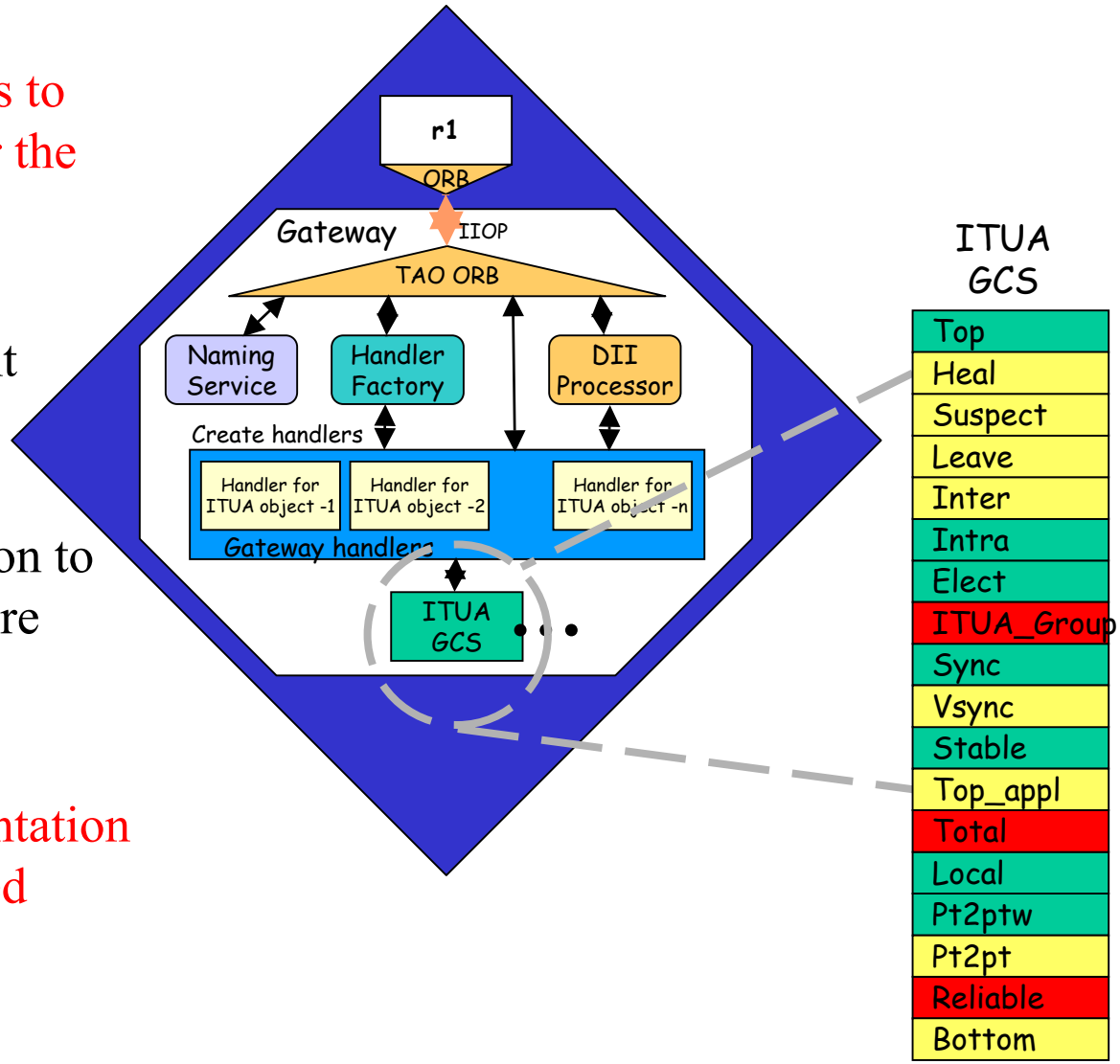
- Signed messages
- Limited types of tolerated corruption
- Corrupt managers convicted, but not recovered

Unpredictable Adaptation: A Potential Approach to Complicate Attackers Mission

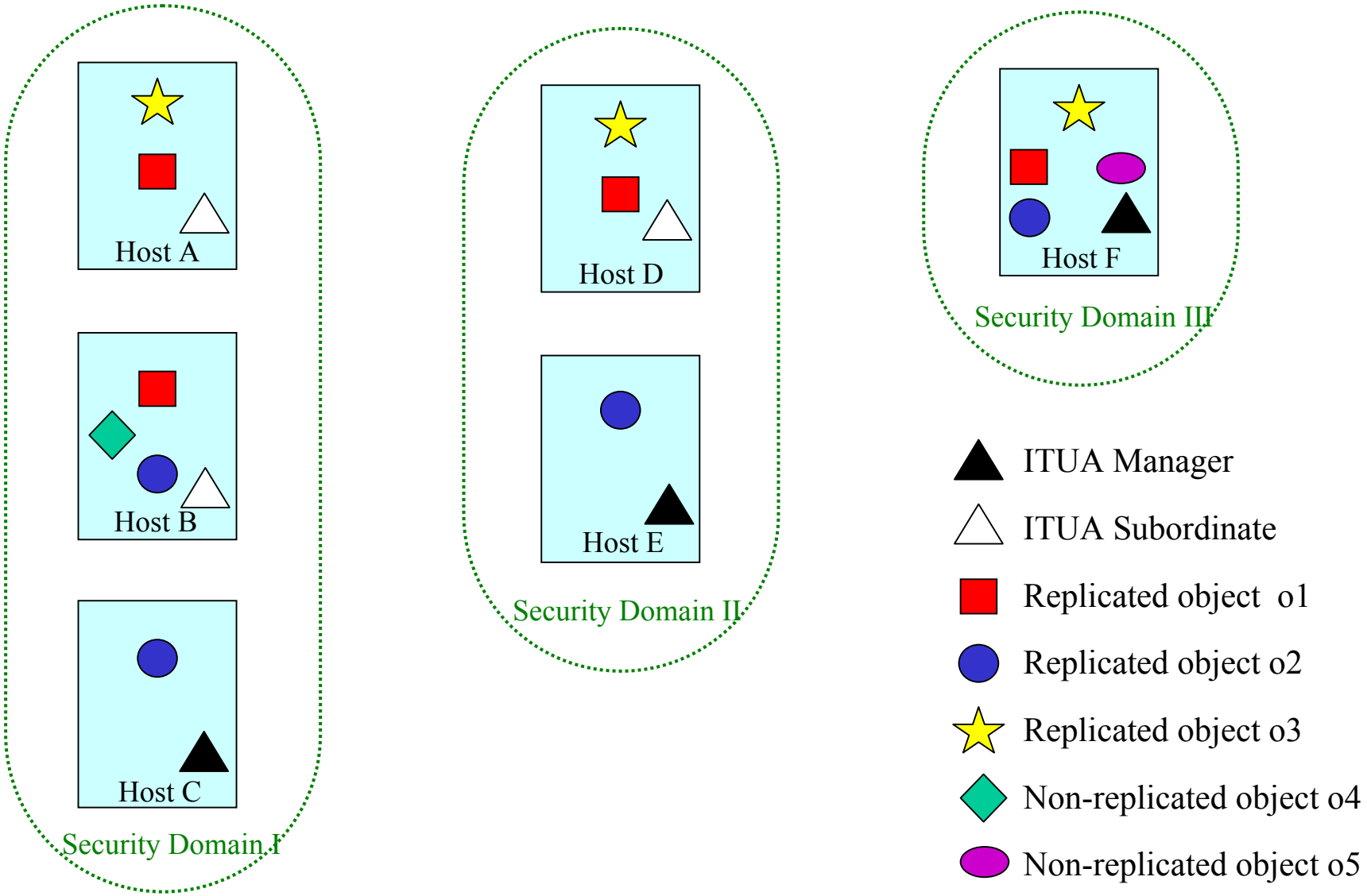
- Unpredictable adaptation mechanism in ITUA (to date):
  - Application level policy support: QuO adaptation
  - Replication Management level: placement of new replica
  - Rapid Reaction level: varying agility, sensitivity, measurement window, nature of response
- Use of *unpredictability* in defense seems to have gathered some momentum (especially in network level)
  - Dynamic Address Translation (DYNAT) (Pre-ITUA) : Frequent dynamic reconfiguration improves system assurance
  - Countering Traffic Analysis in IPSec: The last hop reveals the endpoint: generate additional hops after real recipient
  - Distributed Authentication for Mobile Internet: N packets will cause N different encrypted packets and N different forwarding
  - Probabilistic Routing/Adaptive Multipath Routing: Address the vulnerabilities of static routing/Tolerate accidental and malicious failures in control (routing) and data (transport)
- Experimental corroboration (on a related project)
  - Red team "IOR smashing" was foiled by unpredictable adaptation strategy
- Ongoing work in ITUA: assessing the value of the unpredictability by modeling

Intrusion-Tolerant Inter-Object Interaction

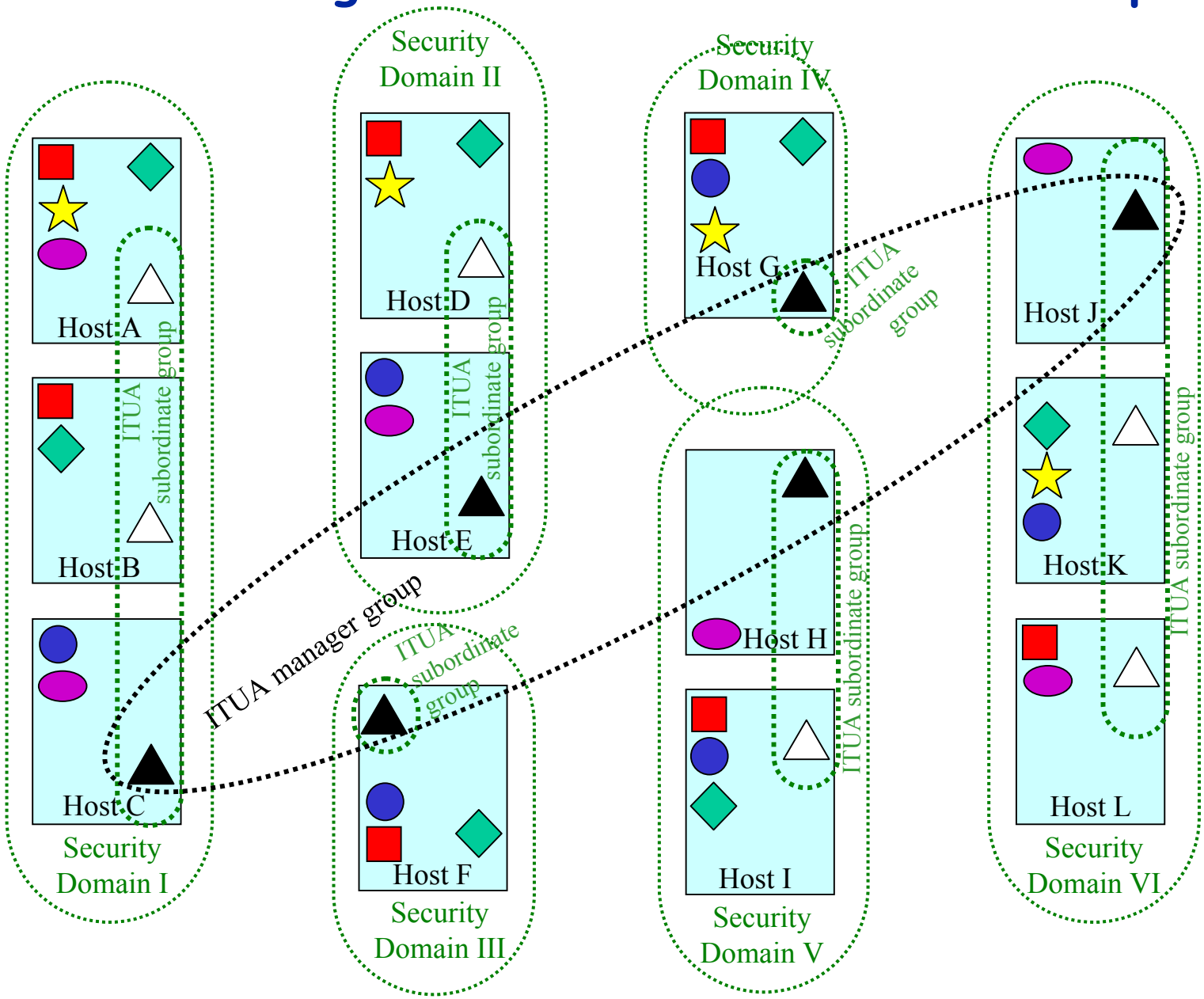
- Replicas use the IT gateways to interact with each other over the ITUA GCS
- The gateways:
  - Provide intrusion-tolerant remote invocations to replicated objects
  - Provide IT communication to management infrastructure
- Several handlers have been developed
- Initial IT Gateway Implementation Completed and Demonstrated



Application Objects and ITUA System Objects in Security Domains



ITUA Manager and ITUA Subordinate Groups



Overview of ITUA GCS

Process groups in ITUA (manager group, subordinate groups, replication groups) share common concerns

maintaining consistent membership info across all correct processes, virtual synchrony, reliable and atomic delivery of multicast messages

ITUA GCS is a convenient way to address above concerns at lower level of system, by providing a process group abstraction

Key Intrusion-tolerant Protocols in GCS

group membership protocol

reliable delivery protocol

total-ordering protocol

Implementation

protocol stack composed of layers providing high-level intrusion-tolerant properties

Modified and enhanced crash-tolerant GCS (C-Ensemble) to get intrusion-tolerant ITUA GCS

Removing Implicit Trust among group members

Authentication by public-key cryptography

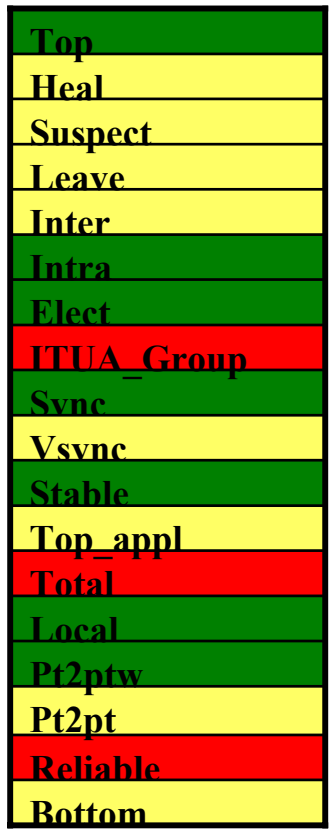
New layers for group membership, reliable and ordered delivery

Validation

By informal proofs (done)

By intrusion injection and performance measurement (done)

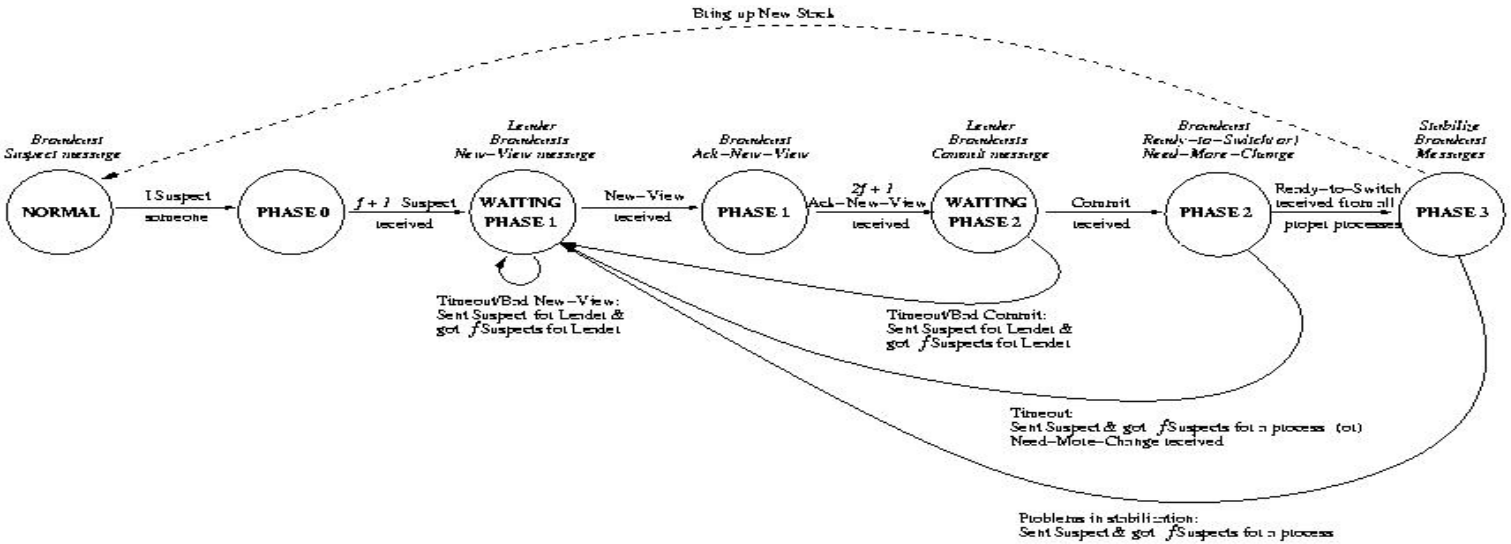
Formal specification and verification using SPIN model checker (Ongoing)



Details on Group Management

Group Formation & Adding new processes to existing group

- Consensus among managers
  - how many processes in group (depends on intrusion-tolerance requirements)
  - where to start processes (which security domains, which hosts)
- Subordinates at specified hosts create processes, key pairs, get public keys certified by CA
- new processes not started in domains suspected to be infiltrated
- new processes join existing group to dynamically increase IT of group, while responding to attacks



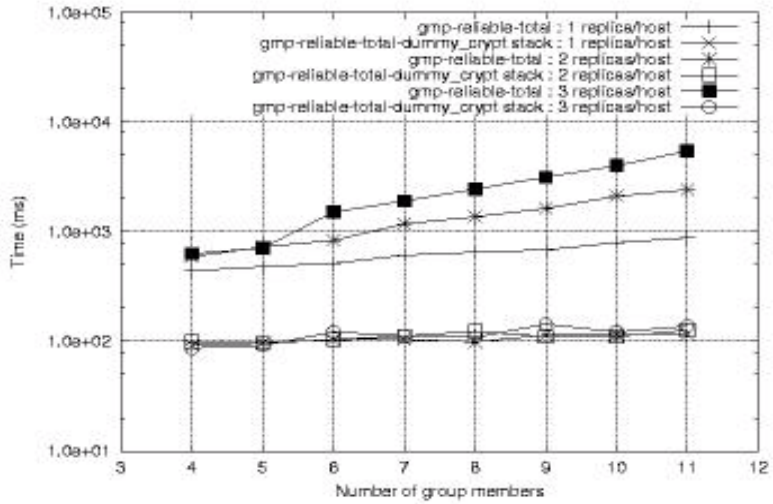
Removing corrupt processes from group (DSN-2002 paper)

- 3-phased agreement protocol based on a finite state automaton (shown above)
- triggered when at least one correct process suspects a fellow group member
- other corrupt processes may be discovered when removing a suspected member (transitional views)

Experimental Validation of Performance

- Extensive analysis of cost of IT group membership and message delivery
- Protocols (DSN-2002 paper)
- Measured costs for
  - various group sizes and key sizes (RSA)
  - single and multiples simultaneous faults
  - different fault permutations
  - different load conditions

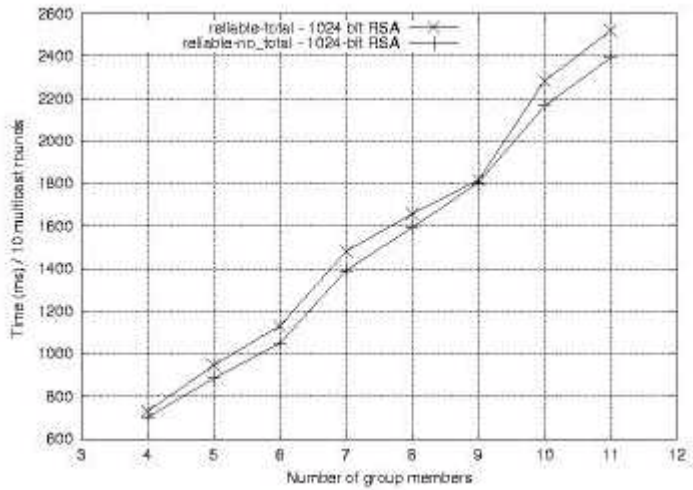
Tolerating single fault under varying load



- Obtained latencies for IT reliable delivery, total ordering
  - by comparing with results for C-Ensemble stack
- Obtained cost due to cryptography
  - by comparing with a protocol stack using dummy cryptographic functions
- Cost for Tolerating Malicious faults >> Cost for tolerating crash faults
  - Public-key cryptography, Communication overhead due to multiple rounds of message exchange for agreement

Lessons Learned

Reliable+Total Ordering vs Reliable only



Crypto costs

- Dominate when computing power is at a premium
- Increase significantly with increase in group size
- Expected to be much less if

  specialized hardware is used

  more powerful machines used

Communication costs

Expected to dominate over crypto costs in WAN environment

Cost of reliable+total-ordering only slightly higher than cost of reliable

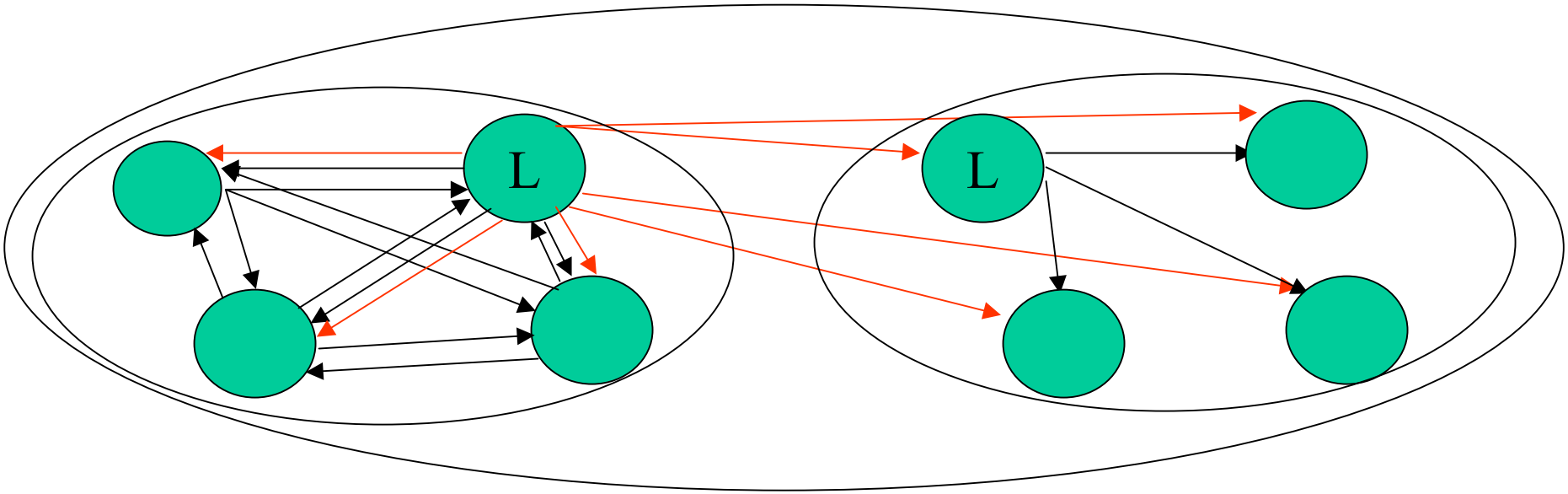
- Total ordering protocol efficient for replication groups where all members multicast similar number of messages
- No sequencing phase, because of pre-assigned ordering

Cost for recovering from simultaneous faults

- Influenced by detection mechanisms (timeouts in a timed asynchronous model)
- Expected be much less with dynamic timeout tuning

IT Handler Algorithm

Step 1 – Multicast signed msg in sending replication group
- 2f+1 signed votes required to proceed to next step



Step 2 --Multicast in connection group, with proof (2f+1 signatures)

Step 3 – Validate Proof
- Multicast in receiving replication group to provide total order
- Upon receipt of this broadcast, proof checked and request/reply sent up to application.

Tolerated IT Gateway Faults

- Any attempt to subvert the broadcast primitive is caught via the group communication system.
- Manipulation of message payload by any member will be caught via voting in step 1, or in proof checking in steps 2 and 3.
- Never sending votes for a particular message (step 1).
- Invalid signature accompanying replication group broadcast in step 1.
- Failure to include sufficient proof in a connection group cast in step 2.
- Failure of leader to send cast in connection group in step 2.
- Failure of leader to send cast in replication group, in step 3.
- Sending a message that does not match the majority of requests generated at other replicas.
- Leader sending a broadcast with a different message or out of sequence messages in step 3.

Status

- Recent Technology Development Accomplishments
  - Completed and demonstrated initial intrusion-tolerant GCS (Santa Fe)
  - Completed initial implementation of of IT gateway
  - Completed initial implementation of IT mangers
  - Completed initial implementation of sensor/actuator loop
  - Completed initial unpredictability support for adaptation (QuO adaptation control extension)
  - Integrated and demonstrated all system parts, and demonstrated using a IEIST Fuselet developed by Boeing (Hilton Head)
- Recent Technology Validation Accomplishments
  - Built and solved high-level and detailed models of redundancy management
- Next Steps
  - Integrate IT GCS and IT Gateway
  - More sophisticated management algorithms
  - More validation …