Section.??.?. Technical Rationale

Secure/Survivable Distributed Execution Environment Concept

Our approach to implementing dynamic instantiation of virtual enclaves based on global security policy and QoS (QoS) metrics (e.g. performance, dependability, dynamic route costs, etc.), in a network and operating system independent fashion, based on overall security policy, depends on a stepwise integration and refinement of several key technologies applied against stressing programs e.g. AEGIS and SC-21 (shipboard applications), LAMPS (littoral warfare application) and CORPS SAM (global mobile application). The table below summarizes the key technologies that will contribute to our Secure/Survivable Distributed Execution Environment Concept. The paragraphs that follow will better describe the problems we are trying to solve and our approach to deriving an integrated solution.

Contributing Group	Key Technology
Sanders/MSU	Applied SHARE and HPSC technologies, dynamic routing, dynamic discovery based on capability, network/OS independent secure message passing and message passing parallel interface and high speed cryptography
LMT/V-ONE/UMD	Dynamic security policy and dynamic application handling support, high speed authentication, addition of secure association in network object broker (please add to this list - Andrew)
Stanford	Expansion of security mediator concept to share secure messages in a network of this proportion
Northeastern	High speed, large, time-dependent behavioral simulation, QoS metrics and application-negotiated trade-off control based on QoS metrics, adaptive programming, real-time resource management based on improved network time synchronization and fault-tolerance
IDS	Incorporation of next generation improvements to application control of group level message passing into ISIS.

The current state-of-the-art in secure distributed systems is that a physical system area networks (SANs) is associated with a singular level of security and authentication/encryption/decryption is performed on packets that move from SAN to SAN. Dynamically instantiated virtual enclaves mandate association of nodes within different SANs to form a enclave that can be associated with a security level and that enclave members can change with time. Furthermore, one needs, locally at each node to have the ability to negotiate/process a wide variety of policies, to have the ability to interoperate, compose and dynamically discover security policy.

The security policies of particular network environments/dynamically instantiated enclaves do not always integrate and interface with easy to administer security hardware/software applications. Most security policies inherently provide difficult or cumbersome interfaces for users trying to access Trusted/Secure network environments.

Furthermore, desktop security applications are hard to implement and administer without supervision from a network engineer or systems administrator. In particular users requiring access to trusted/secure enclaves vary from project to project and therefore dynamical change security requirements for access to networks and particular services and servers. {action - back fit Gios research contributions. Given a secure system, he permits virtual sharing of messages in a way consistentent with our dynamic enclave concept. }

We will integrate a seamless way of registering users over an "Intranet" or "Inter-networking" environment, allowing for the integration of {Fortezza/SmartCard} tokens to mutually authenticate client/server applications and allow for the encryption and non-repudiation of the communications between trusted and untrusted environments. Integrated in this revolutionary approach is the ability to predetermine the entitlements of trusted users to particular services/servers, IP addresses, objects, ports or in the case of web services, particular URL pages. (Fine Grain Access Control) Dynamic Reconfiguration provides a transparent means of verifying a users privileges in a trusted/secure networking environment and provides the System administrator web-based tools to edit or delete entitlements on a daily basis without any interface with the client application at the desk-top. These administrative tasks can be performed securely with encryption and token based mutual authentication from any location in the world.

Another limitation to network security is that security policies implemented by most operating systems and network protocols are inflexible and do not address application level security requirements. The operating system provides no assistance in implementing such policies. {insert Robert's writeup here on the way MPI and PacketWay (as a function of SHARE) will help us. Include the HORUS/ISIS part of the equation}

We have identified the following objectives that are largely not satisfied by present day security technologies: Support for a broad range of security policies, Support for securing application objects (fine grained labeling and access control), Support for dynamic security associations. To use an object oriented framework and library of dynamically loadable security components. Specific security components are implemented as concrete subclasses which can be dynamically loaded and attached to the abstract interfaces. This allows us to satisfy each of the objectives in the following way: A broad range of security policies (Mandatory Access Control, Role Based Access Control, Discretionary Access Control) can be supported by developing abstract interfaces that capture the commonality across various types of security policies. By developing the security components as object classes, it is possible to embed these objects in application programs as well as operating systems or middleware. This allows applications to provide fine grained access control and labeling on application objects. Security policies can be dynamically established between a client and a server by dynamically loading the appropriate security components. This canbe done as part of negotiating an overall QoS between the client and the server.

Other technologies that can aid in solving these include code signing and Authentication Brokering. Code signing is a technology invented and patented by LMFS-G in August of

1994. Thistechnology allows code modules to be digitally signed through the use of any public key encryption algorithm. The digital signature can be used to insure the integrity of the code module as well as to identify the actual source of the module when a module is delivered over an open network. Authentication Brokering can be used to establish a single sign-on capability in a heterogeneous network. User's log into an Authentication Broker once, and then the Authentication Broker transparently manages the process of logging the user into other network services requiring various forms of authentication.

In a real-time distributed system, QoS can vary significantly due to dynamically changing characteristics of the computing system. Those characteristics can be of two types: (1) structural changes (some nodes fail, some protocols fail, packets are delivered out of sequence, data types are inconsistent, software-hardware incompatibility), (2) analytical changes (due to either the above structural changes, or due to increased load, the dynamic characteristics change: throughput, delays, and other dynamics related). These two groups are related, but both need to be addressed to provide timely and reliable service under the changing conditions.

To develop a closed-loop adaptive system that monitors QoS and controls it according to a prespecified control goal through the utilization of a number of techniques that include adaptation (both structural and analytical) of its processing in response to faults, changing loads, requirements and capabilities, or through dynamic re-negotitation and reconfiguration of its processing shcemes.

Controlling structural changes: (Insert Karl's stuff comes here - we already have inserts)

- Development of the network modeling tool

- Adaptive programming

- Synchronization of clocks

- Self-stabilizing protocols

Controlling analytical changes: We propose to carry out research leading to the satisfaction of the goal of controlling a distributed computing system so that QoS specifications are met. The main premise of our approach is that a distributed computing system is a dynamic system and thus needs to be modelled, designed, analyzed and controlled from this perspective. We will develop monitoring policies, modelling and designing approaches and finally control algorithms. Control algorithms will utilize both traditional control approaches like adaptive control and robust control, as well as new control paradigms that include restructurable control, gain scheduling, hybrid systems, and intelligent control. To implement this kind of control, a revolutionary approach to designing distributed systems is necessary. The system design needs to be based on specifications that include not only functionality of the system but also all the characteristics that affect the QoS measure. The design must incorporate not only the the

functions that are needed from the computational point of view, but also functions that are needed from the control point of view. The authors have conducted experiments in real-time adaptability of a distributed system to changing requirements and they will expand on this knowledge to achieve the objectives of this proposal.

This task is closely related with the previous task. In one of the adaptive control schemes, called Model-Reference Adaptive Control (MRAC), a model needs to be designed to generate the desired behavior of the controlled system. In our approach, the network modelling tool of Task 1 will be used for this purpose. This tool will monitor the behavior of the network (does it have this capability ?????????????) and generate the QoS requirements for the adaptive controllers associated with particular nodes. Additionally, the tool will be augmented with the capability of assessing the stability of the network for structural changes that may occur during the network operation and with the capability of assessing the quality of the control algorithms.

We will apply the above technology against shipboard, littoral warfare and globle mobile applications. For instance, in the shipboard case, the distributed heterogeneous laboratory that models Navy combat system operation at NSWC uses a PC acting as a Compartmented Mode Workstation to communicate between the secure and unsecure modeling environments. The HiPer-D Program uses this testbed as the application to insert technology and is used to support various application and network related trade-offs. On the other hand, the SHARE Program has created secure and real-time modifications to the MPI and PacketWay standards and a high speed network/crypto design to support military/commercial applications. The problem is A) the NSWC testbed needs to support a more general secure message passing scheme that may be used in wider-scale shipboard applications, e.g. SC-21, B) the SHARE Program needs specific application requirements to complete its detailed design and C) the NSWC testbed is planning to evolved to a highly interconnected testbed (e.g. described in Quorum) that will mandate a revolutionary higher speed general purpose message passing, brokering existing applications.

On the short term, we propose a coupling of technologies, currently evolving on the NSWC testbed, with those developed under the SHARE Program, using an off-the-shelf, NSA approved, crypto module as the heart of the SHARE secure router (this is where V-one comes in) (refer to figure). This will permit an early demonstration of a secure, network and operating system independent message passing. This system will provide for authentication, independent of encryption, and will take advantage of the existing centralized instrumentation. On the longer term, the goal is to A) provide dynamic discovery of routes and network participant permissions based on topology, node capability and application negotiation, B) add an object broker to provide a generalized client/server model and interface language to better partition applications based on workload and the overall network security methodology, C) extend the above with the more generalized, highly-interconnected NSWC laboratory evolution and D) add the SHARE high-speed cryptography hardware.

By establishing and adding secure extensions and a secure PacketWay interface to the current ISIS-based message passing system, the SHARE secure network routing system may be used with minimal impact to NSWC applications. The ISIS improvements will be woven into the fabric of the next generation ISIS product, HORUS, whose existing security architecture will be melded and implemented with the SHARE approach. This approach will achieve short-term operability, with ISIS/PacketWay providing OS independence and PacketWay providing network independence (see figure). Next, an ORBIX-ISIS product can be used for an interim object-broker, while more suitable object brokers are being traded, providing a baseline for these trades. Extensions to MPI will provide the dynamic discovery requirements described in the objectives. This early baseline, low risk, approach guarantees on-demand operable demonstrations that can be coupled with the HiPer-D demonstrations, and a feedforward of knowledge/requirements to closely spaced incremental project phases.

{refer to Corps SAM, SC-21 and LAMPS application}