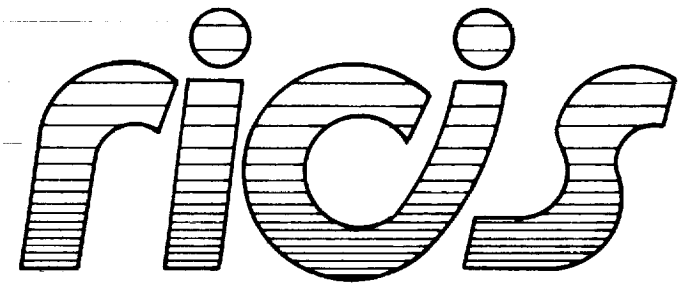
SECURITY FOR SAFETY CRITICAL SPACE BORNE SYSTEMS

S. LeGrand
SofTech, Inc.

December 31, 1987

Cooperative Agreement NCC 9-16
Research Activity No. *SE.12*



Research Institute for Computing and Information Systems
University of Houston - Clear Lake

T·E·C·H·N·I·C·A·L R·E·P·O·R·T

The RICIS Concept

The University of Houston-Clear Lake established the Research Institute for Computing and Information systems in 1986 to encourage NASA Johnson Space Center and local industry to actively support research in the computing and information sciences. As part of this endeavor, UH-Clear Lake proposed a partnership with JSC to jointly define and manage an integrated program of research in advanced data processing technology needed for JSC's main missions, including administrative, engineering and science responsibilities. JSC agreed and entered into a three-year cooperative agreement with UH-Clear Lake beginning in May, 1986, to jointly plan and execute such research through RICIS. Additionally, under Cooperative Agreement NCC 9-16, computing and educational facilities are shared by the two institutions to conduct the research.

The mission of RICIS is to conduct, coordinate and disseminate research on computing and information systems among researchers, sponsors and users from UH-Clear Lake, NASA/JSC, and other research organizations. Within UH-Clear Lake, the mission is being implemented through interdisciplinary involvement of faculty and students from each of the four schools: Business, Education, Human Sciences and Humanities, and Natural and Applied Sciences.

Other research organizations are involved via the "gateway" concept. UH-Clear Lake establishes relationships with other universities and research organizations, having common research interests, to provide additional sources of expertise to conduct needed research.

A major role of RICIS is to find the best match of sponsors, researchers and research objectives to advance knowledge in the computing and information sciences. Working jointly with NASA/JSC, RICIS advises on research needs, recommends principals for conducting the research, provides technical and administrative support to coordinate the research, and integrates technical results into the cooperative goals of UH-Clear Lake and NASA/JSC.

SECURITY FOR SAFETY CRITICAL SPACE BORNE SYSTEMS

Preface

Sue LeGrand is Program Manager for SofTech Houston Operations. Sue did the primary research for this project under a subcontract between SofTech, Inc. and the University of Houston-Clear Lake.

This project was conducted under the auspices of the Research Institute for Computing and Information Systems. Dr. Charles McKay, Director of the Software Engineering Research Center at UH-Clear Lake, provided the overall technical direction for this research. Funding was provided by the Spacecraft Software Division, Mission Support Directorate, NASA Johnson Space Center through Cooperative Agreement NCC 9-16 between NASA/JSC and UH-Clear Lake.

The guidance and direction of this project by both Dr. McKay and the NASA Technical Monitor, Steve Gorman, is gratefully acknowledged.

The views and conclusions contained in this report are those of the author and should not be interpreted as representative of the official policies, either express or implied, of NASA or the United States Government.

A Report on Security for

Safety Critical Space Borne Systems

Final Report

31 December 1987

Contract SE. 12

Prepared for

University of Houston at Clear Lake
2700 Bay Area Blvd.
Houston, Texas 77058

Prepared by

SofTech, Inc.
1300 Hercules Drive, Suite 105
Houston, TX 77058-2747
(713) 480-1994 TWX: 710-3242-6401

TABLE OF CONTENTS

Security for Safety Critical Space Borne Systems

Sue LeGrand

SofTech Houston Operations

Abstract

The Space Station contains safety critical computer software components in systems that can affect life and vital property. These components require a multilevel secure system that provides dynamic access control of the data and processes involved. A study is under way to define requirements for a security model providing access control through level B3 of the Orange Book. The model will be prototyped at NASA/Johnson Space Center. This paper will summarize those requirements as summarized.

Introduction

The Tri-Service Software Systems Safety Working Group composed the following definition: [3]

Software systems safety is the optimization of systems safety in the design, development, use and maintenance of software systems and their integration with safety critical hardware systems in an operational environment.

At about the same time, an independent task force led by the Software Engineering Research Center (SERC) at the University of Houston at Clear Lake (UH-CL) proposed the following definition for use in the research for the Space Station Program:

Safety critical computer software components are defined as computer software components (processes, functions, values or program states) whose inadvertent occurrence, failure to occur when required, occurrence out of sequence, occurrence in combination with other functions or erroneous value can result in a hazard or loss of system predictability or control.

Software can be used in the following ways to safely control a system:

Autonomous control
Control of potentially hazardous hardware or system components
Management of information requiring immediate operator action
Management of information with which an operator or another system makes critical decisions.

The Space Station is one of the most complex systems yet undertaken. This increases the risk of a combination of faults that individually may not be a problem, but combined can cause a serious threat. Safety planning must be incorporated into the system development effort from the beginning. This allows technical safety mechanisms to complement and augment rather than supplant the system design.

An integral part of this safety is security. Control must be left in the domain of the security staff and the trusted computer system. It should never be available to the hacker who inadvertently causes damage, the terrorist who maliciously seeks to cause destruction, or a disgruntled former staff member who leaves with potentially dangerous knowledge. Furthermore, software domains should be protected against run away program components in another domain and should be recoverable at higher layers within their own domain.

A special multilevel security model must be developed and prototyped that will fill the safety requirements of the entire Space Station system with its ground, on-orbit and co-orbit subsystems.

Standards Used

The following standards are recommended for developing a security prototype for Space Station multilevel security.

OSI Communication Model [8]

The Open Systems Interconnection provides standards for the exchange of information among systems that are "open" to another for this purpose by virtue of their mutual use of the applicable standards. A system is a set of one or more computers, associated software, peripherals, terminals, human operators, physical processes, information transfer means, etc., that forms an autonomous whole capable of performing information processing and/or information transfer.

OSI is concerned with the exchange of information between open systems and not the internal functioning of each individual open system. The objective of OSI is to define a set of standards to enable open systems to cooperate. It uses a layered architecture for operations such as intersystem connections, transmission of data and error functions. The OSI model is mandated for the design of a network operating system for the Space Station Data Management System. [4]

CAIS-A System Interface Model [21]

MIL-STD 1838-A, "CAIS-A", provides specifications for a set of Ada packages with their intended semantics, which together form the set of common interfaces for Ada Programming Support Environments (APSEs). This set of interfaces is known as the Common APSE Interface Set (CAIS). This interface is designed to promote the source-level portability of Ada programs, particularly Ada software development tools. The CAIS Model is found in the specification for the Space Station Data Management System (DMS). [4]

The goal of the CAIS is to promote interoperability and transportability of Ada software across DoD APSEs. Interoperability is defined as the ability

of APSEs to exchange database objects and their relationships in forms usable by tools and user programs without conversion. The DMS Specifications call for interoperability of information and database components. Transportability of an APSE tool is defined as the ability of the tool to be installed on a different Kernel APSE (KAPSE); the tool must perform with the same functionality in both APSEs.

The CAIS Model has entities called objects shown as hierarchical nodes on a directed graph. The contents, relationships and attributes of nodes are defined as well as attributes of the relationships.

The Orange Book Level B3 Security Requirements [5]

The DoD Trusted Computer System Evaluation Criteria, commonly called the "Orange Book", provides a metric with which to evaluate the degree of trust that can be placed in computer systems for the secure processing of classified and other sensitive information. It designates two types of privileged access: discretionary and mandatory. Discretionary access control limits authorized access of objects to named individuals or groups. Mandatory security protection involves a comparison of the individual's clearance or authorization for the information and the classification or sensitivity designation of the information being sought.

A Level B3 trusted computer base (TCB) involves both discretionary and mandatory access control. It must satisfy the reference monitor requirements that it mediate all access of subjects to objects, be tamper proof and be small enough to be subjected to analysis and tests. A security administrator is supported, audit mechanisms are expanded to signal security-relevant events, and system recovery procedures are required. The access controls are said to be dynamic, since they are administered at run time rather than at compile time.

The Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria (TNI) [6]

The Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria (TNI) provides a basis for the evaluation of effectiveness of security controls built into networks and network component products. The specific security feature, the assurance requirements, and the rating structure of the Trusted Computer System Evaluation Criteria are extended to networks in part 1. In part 2, there is a description of a number of additional security services that arise in conjunction with networks.

The Clear Lake Model for Dynamic Multilevel Security [9]

This model was first presented at the AIAA/ACM/NASA/IEEE Computers in Aerospace V Conference, October, 1985. It was developed in a program now called the Software Engineering Research Center at University of Houston at Clear Lake and sponsored by NASA Headquarters. The Director is Dr. Charles McKay, who spawned most of the concepts. The model complies with the philosophies of the Orange Book, TNI, OSI and CAIS-A standards. It contains attributes about the subjects and objects used in the above standards. The model emphasizes developing and sustaining integrity of mission and safety critical components in the target environment rather than the protection of classified data.

Estimated Requirements for Proof-of-Concept Prototype

These requirements are based on the Clear Lake Model and reflect current Space Station design and current versions of the above standards.

Major SSP Data Management System (DMS) Requirements [4]

The following is a sample of capabilities mentioned in the SSP Definition and Design Requirements Section 2.3.4 on the Data Management System that present a challenge to security in this large complex system. This is especially true in flight safety critical subsystems. There is a question as to how to verify the authority in each case.

Override capability of automatic systems is required for authorized onboard and ground crews.
A User Interface Executive must keep track of which user is linked to which application tool.
A User Interface Language Interpreter must validate that the user has access to processes and files.
A User Interface Data Base must contain user profiles.
Performance and trend data is to be displayed to those authorized.
Payload interfaces are to be monitored by those authorized.
Authorized emergency priority messages must be forwarded.
Error performance must be user selectable.
Constraint checking is required.
Command differentiation is to be suitable for managing operations and protect the system from inadvertent and unauthorized payload operations.
Potentially hazardous commands shall be identified.
Command, communications and data interfaces shall be permitted for user operations facility to payload.
Onboard computational resources shall be provided to host authorized customer-provided software.
Authorized customers shall be allowed to employ data privacy mechanisms.
Privacy must be assured for scientific or user-proprietary data.
Customers must be able to exchange data subject to privacy constraints.
Authorized access is required to onboard planning and operational support data with change notation/notification/approval capabilities built in.
Authorized crew or ground personnel shall be able to modify DMS applications software.
Controlled exchange of data between onboard payloads must be offered.
Protected onboard mass storage capability for customer use must be provided.
Report of transmission status for authorized customer data transfers must be supported.

The customer must be kept informed on the status of the part of the end-to-end onboard network which he is using.
The transparent transmission, reception, processing, controlling, storage and distribution of operational data and commands must be supported but also maintained by those authorized.
Caution and warning systems must be managed station-wide and interoperable between international modules and maintained by those authorized.
The DMS Systems Manager must interact with the Space Station Information System (SSIS) Network Integration Manager for security management.
Local and remote access to onboard DMS databases must be consistent with security and privacy practices.
Authorized onboard processes must be able to obtain or clear a global name with the SSIS naming authority.
Multiple, concurrent, authorized access to onboard data shall be provided without interference.
Restricted views of the database must be offered to the users.
Structures are to be provided that allow the user to build his own checks and control processes for security.
An emergency mode fix to operational software or data tables may be installed onboard with less comprehensive control.

Major Orange Book Requirements

The Trusted Computer Evaluation Criteria ("Orange Book") lists the following requirements for level B3 security domain for a trusted computer base (TCB) that compliment the DMS software specification:

Discretionary access control shall be to define and control access between named users and named objects.
Mandatory access control shall consist of hierarchical classification levels and non-hierarchical categories.
Sensitivity labels shall be associated with each system resource.
A subject can read an object at an equal or less security level.
A subject can write to an object at an equal or more security level.
Storage objects shall be purged before reuse.
The security level of each communication channel shall be maintained and audited.
All human readable output must be marked at the top and bottom of the display.
Terminal users must be immediately notified of a change in security level of their associated objects.
Subjects must be identified before performing any actions.
Authentication data must be protected.
Individual users must be uniquely identified.
Interface between the TCB and user must be by a trusted path.
There must be an audit trail of access to all objects.
Accumulation of event that may indicate an imminent violation of security policy must be monitored.

Immediate notification must be made when security thresholds are exceeded.
The domain of the TCB shall be established for its own execution and must be protected from interference.
The design of the TCB must incorporate significant use of layering, abstraction and data hiding.
The security system must be periodically validated.
A Security Administrator must be identified and his actions be auditable.
Formal and informal models of security policy must be maintained.
A configuration management plan must be maintained.
A user's guide must be established and maintained.
A design document must be maintained that describes the interfaces of the TCB.

Major Requirements: Clear Lake Model for Dynamic Multilevel Security

The Clear Lake Model is based on LeGrand's access control model for a distributed, CAIS-conforming system [9]. It has features that answer the requirements of the DMS Specifications and the Orange Book requirements. It also complies with the Clear Lake Model for a Network Operating System. This system uses Network Application Services, Network Information Services and Network Communication Services which share a common interface and a common approach to Network Configuration Management Services.. Each set of services has agents or managers that act on behalf of subjects or control aspects of objects.

Subject and object nodes, as used in the CAIS Model, would have attributes that comply with the ISO Reference Model, Orange Book and TNI. Subjects having physical capability and security privileges access objects through object managers. The subject attributes would be sent with the request to access an object. The object attributes would be located with a description of the object, and the object manager in the object's subsystem would provide access after making appropriate comparisons.

Attributes of the subject would be provided by the subject's application software and by the access manager in the subject's subsystem. The subject's application software would provide the following:

User/User class/Access Identification (ID) recognizable throughout the distributed system.

Logical Reference meaningful to the user and not necessarily an ID unique throughout the system.

Intent to be compared to the intent under the object's attributes.

Object Type where each object will be typed just as Ada elements are typed (record, directory, editor, tape drive, etc.)

Requested Priority

Password as required.

The access manager in the subject's subsystem verifies the input and provides the following:

Subject's clearance level.

Subject's subsystem location and device.

Unique ID of Parent Thread-of-Control

Unique Transaction ID

Unique object ID.

Current roles: Capability sets as provided by CAIS (e.g.; user, job, project librarian).

At the object's subsystem, the following attributes may be associated with the object node and listed once each:

Unique Object ID

Type determining possible values and legal operations.

Ownership

Lock Management Information

Priority Management Information

Access History, a summary of on-line records plus archives.

Multiple Copy References for distributed copies.

Encryption Name, such as an algorithm or public key, if allowed.

Sensitivity Labels for DoD mandatory access control.

The unique ID may be composed of network entity ID, network LAN ID, network cluster ID, cluster component ID, resource ID, processor ID, and time/date stamp. This is based on the premise that each single processor can create, at most, one object or one transaction during a time/date stamp period. For each possible subject, this would be listed once:

User/User Class/Access ID

The following may be listed as many times as necessary for each subject:

Logical Reference, an alias permitted to the subject.

Intent, which establishes access synchronization with users of a node.

Access Capability Requirement, necessary and resulting privileges provided in CAIS.

Location Restriction, such as secure area requirements.

Password, as required.

Template Restriction allows access to only designated fields, records, or other subsets of the file.

The CAIS node model shows that up to four instances of the last set of attributes may be needed. They are:

1. The node itself.

2. The attributes of the node.

3. The relationships of the object node to other nodes.

4. The attributes of the relationships.

For instance, the subject may be able to copy the object but not be allowed to read the attributes. The subject may be allowed to know that an object exists but is not able to copy it (e.g., execute).

General Requirements for a Dynamic Multilevel Security Prototype

Support for access control should be physically located in the extended run time library (XRTL) and execute as privileged code under the control of a Mission and Safety Critical (MASC) Kernel. It should be developed in a modular fashion so that no more support than is needed is loaded into a target.

Fault management should be offered in the XRTL. Data integrity must be preserved in spite of system failures.

Access control and constraint checking must be maintained non-stop.

Dynamic redefinition and selection of all data being transferred must also be verified.

Security activities must be accomplished with no disruption of DMS services.

Checkpoints are needed for fault tolerance. The mechanism for these must be secured in all flight safety critical systems.

Deadlock detection and resolution are needed for fault tolerance. This mechanism should also be secured in all flight safety critical systems.

Security of data must be maintained through out its entire life cycle.

Security labels must be included with all transmitted data and transparent to the user.

Devices should be labeled and the label should include the physical location.

Information and/or control messages for mission and safety critical components that are to be transmitted, mailed or otherwise subject to possible exposure must be encrypted.

Space Station subsystems must operate autonomously. No more data than necessary must be exchanged.

Some secure data must be "invisible" to unauthorized subjects. The CAIS-A Model provides for such access control.

All access activity (including failed attempts) must be monitored and an audit trail maintained. A decision is needed for each type of security as to how long this log must be archived and where it should be located. The CAIS Model provides paths for an audit trail.

The User Interface (UI) Executive is charged with keeping track of which user is linked to which object and with routing messages or requests. This concept should be merged with the idea of object managers in the Clear Lake Model.

The UI Language (UIL) Interpreter must validate UIL statements and that the user has access to objects. This should also be considered under the aspect of the object manager.

User interface menus should not be displayed until the user has been verified and should only display options and objects which are currently legal to the user.

A user profile data file, as identified in the DMS Specification, should exist in a secure location and contain his access authority information perhaps in an encrypted form. This information should not be duplicated outside of the profile. Passwords should be meaningless but pronounceable and created by the owner.

User-created access controls must be confined to objects owned by that user. If ownership changes, the access controls must be re-created by the new owner.

Automatic backup and recovery is not supported by the software installation system. Restart conventions, including manual identification of a safe backup point, must be through controlled access.

The DMS shall dynamically manage the redundancy of the DMS network services and resources and automatically reconfigure for fault tolerance. The same access control that existed before the fault must be maintained. Changes in access control data that may have been made during the recovery process must be dynamically monitored and installed in the reconfigured system.

The security level of any part of the system must be dynamically adjustable within the multilevel security definition at run time.

Binding of appropriate resources must be dynamically adjustable at run time.

The entire system or any part must be dynamically extensible at run time.

A transaction (T) or subtransaction (S) is the smallest group of actions that can be considered as a unit. Each will succeed or fail as one unit. The system must be capable of managing distributed, nested transactions which support:

parallel execution of Ts and Ss
recovery at the T or S level.

For each transaction or subtransaction, an application specified set of ordered recovery options will be supported.

Each process in a state of execution must have its thread of control hierarchically accessible by the PCEE. Each process must be able to trace its ancestry to the root node of the program or query. Each must be able to identify and query its children processes.

Services and resources of the system must be structured hierarchically and be sustained as stable interface sets. [14] All recovery procedures must be performed through these layers. Higher layers can recover from specified types of failure in lower layers.

Procedures in individual system parts must be capable of calling or sending messages to each other remotely. Some messages require the semantics of "send, no wait". Some messages require the semantics of "remote procedure call' with a specified approach to orphans and other faults. the third set of message requirements support the "rendevous" semantics.

For each mission and/or safety critical application process which is eligible to be executed on some specified set of processing resources, behavioral predicates will be asserted which produce independent and timely health and status checks on the progress of the process. When progress is unacceptable because of problems involving the process, the processor, or the context of the execution environment, specified recovery actions will be initiated.

Comparison of Requirements and Available Standards/Proposals

The following table lists the recommended features of the proposed security prototype and sources of these requirements or available implementation guides.

Prototype Requirements and Available Standards

Requirement	CL Model	CAIS-A	Orange Book	DMS Spec.	OSI
Access history	x		x	x	
Agents	x				x
Audit trail	x	x	x	x	

Requirement	CL Model	CAIS-A	Orange Book	DMS Spec.	OSI
Binding of resources (dynamic)	x				
CAIS, OSI binding	x				
Capability attributes	x	x			
Checkpoints	x				
Classify subject & object	x	x	x	x	
Command/request messages interop.	x			x	
Compare subject, object attributes	x	x	x		
Configuration Mgt.	x		x	x	
Constraint check	x		x		
Data display to authorized	x		x	x	
Date, time stamp	x		x	x	
Deadlock detection and resolution	x				
Device Label	x	x	x	x	
Discretionary access	x	x	x	x	
Emergency mode fix				x	
Encryption name	x				
Extensibility (dynamic)	x				
Fault tolerance	x		x	x	
Firewalling	x		x		
Formal model			x		
Info/DB component interop.	x	x		x	
Informal model			x		
Interface document.	x		x	x	
Label export	x		x	x	x
Label integrity	x	x	x		
Label output	x		x		
Layered recovery	x				
Mandatory Access .	x	x	x	x	
Monitor events	x		x	x	
Multi-copy ref.	x				
Multilevel security (dynamic)	x		x		
Nested transactions (distributed)	x				
Network security	x			x	x
Notify user			x	x	
Non-stop perform.	x			x	
Object managers	x	x			
Object type	x	x	x		
Objects hierarchy	x	x	x		

Requirement	CL Model	CAIS-A	Orange Book	DMS Spec.	OSI
Object reuse			x		
OSI communication	x			x	x
Override automatic systems				x	
Password	x		x	x	
Periodic validation	x		x		
Pools of processes and processors	x				
Prioritized message	x			x	x
Protect mass stores	x		x	x	
Protect TCB	x		x		
Read up	x		x		
Reconfiguration (dynamic)	x				
Redundancy mgt. (dynamic)	x				
Remote procedure call	x	x			x
Remote task rendevous	x			x	
Restart conventions	x			x	
Security administrator	x		x		
Send, no wait	x				x
Simulations	x				
Single site image	x			x	x
Software (multi-version, fault tolerant)	x				
Stable storage	x			x	
Sys. documentation	x		x	x	
Sys. monitor	x		x		
TCB domain		x	x		
Template restriction	x				
Threads of control (hierarchical)	x				
Trusted path			x		
Unique device ID	x			x	
Unique location ID	x				
Unique subject, transaction, object IDs	x		x	x	
User IF database	x		x	x	
User IF executive				x	
User IF language interpreter				x	
User's manual	x		x	x	
User's privacy mechanisms				x	
User profile	x			x	

Requirement	CL Model	CAIS-A	Orange Book	DMS Spec.	OSI
Views restricted	x		x		
Write down	x		x		

Suggested Further Research and Resources

The following are areas that require more study. An attempt is made to identify sources of information or possible prototypes.

The Trusted Network Interpretation (TNI) is now available. This document must be consulted before the prototype is built. Jay Ferguson at the National Computer Security Center, NSA, has offered to advise us of the progress of this effort.

A level B3 trusted computer base (TCB) requires a formal model of the security policy supported by the TCB. J.E. Heaney, PRC Government Information Systems, presented a selection criteria for their secure microcomputer network prototype. [7] None of the formal models studied supported a network security model, and one must be developed for the prototype.

A packet switched network must provide a mechanism to encrypt data but not encrypt packet headers and trailers. One possibility is to encrypt the data before packetizing.

Simulations will be needed to test the security prototype and the final security system.

A binding is needed between the OSI Communication Model and CAIS-A. The CAIS-A design team has stated that communication and network management is beyond the scope of the CAIS-A Model.

The Clear Lake Model of a NOS calls for rendevous of remote tasks, remote procedure calls and "sends with no waits." A standard must be defined for the granularity of security among job, process and task and a means to enforce the security policy.

A way must be found to verify the location of a subject. Clearance might depend on the surroundings. Small portable devices are a challenge for location labels. How can the label be verified every time the device is moved?

A complete security chain consists of accountability, prevention, detection and enforcement. All of these can be mostly accomplished with software, but eventually administration is needed. A prototype plan must include documented administration rules and procedures.

A dynamic multilevel security model is one of twelve components proposed for a system interface set of a Portable Common Execution Environment (PCEE) to provide the proper both mission and safety requirements fulfillment for the Space Station. The other eleven components are required for a prototype that would completely prove security as an important aspect of safety. They are listed below:

Interface features and options for a tailorable run time support environment.
Software structures which facilitate firewalling, layered recovery capabilities, dynamic reconfiguration and extensibility for fault tolerance.
Pools of processes and processors capable of non-stop operation in a fault tolerant programming environment.
A multiversion, fault tolerant programming capability.
Command language interface between the SISs of the integration and target environment's PCEE.
System-wide like-cycle-unique identification and history of all subjects, objects and transactions.
Dynamic, multilevel security in the target and integration environments (B3 class integrity requirements and beyond).
A message interface supporting communication among clusters.
Hierarchical run time structure of the threads of control of each program.
A redundancy management subsystem for safety critical services and resources.
A stable storage subsystem for each cluster.
A management subsystem for distributed, nested transactions.

Summary

Safety for the Space Station necessitates security and many more elements. A means must be found to provide automatic, quick decisions on who or what is allowed access to flight safety critical applications.

REQUIREMENTS AND ESTIMATE FOR A

PROTOTYPE MULTILEVEL SECURE SYSTEM

A minimal configuration for a proof of concept for an end-to-end multilevel secure system for the major entities of the Space Station Program must be capable of simultaneously supporting the following demonstrations of multilevel security enforcement under a variety of work load scenarios:

within the processing of each participating processor,
within the processing of each cluster of parallel processors interconnected by shared memory or shared bus,
within the processing of each local area network (LAN) with clusters interconnected by a serial communication media,
within the processing of a wide area network (WAN) which integrates physically separate LANs in to an end-to-end system.

A fully instrumented, highly reconfigurable test bed for the target environment is proposed to facilitate this investigation and development. It will include programs to provide access control management needed to support the integrity of mission and safety critical components for Orange Book level B3 and below. It will rely on the existing network, database and applications management routines in the prototype systems.

The test bed is described by the items below:

1. To achieve this demonstration capability, a minimum configuration is needed to support:

Three LANs interconnected by a WAN
Three clusters per LAN
Three processors and associated resources per cluster

2. Specified procedures and guidelines shall permit additions of LANs, clusters per LAN and/or processors per cluster which shall not require:

The system to be stopped,
The operating system to be changed, or
The integrity of the multilevel security (MLS) system to be compromised.

3. Different clusters may be implemented with different instruction set architectures without loss of:

Binary level interoperability,
Source level transportability, or
MLS integrity among the clusters.

4. Different clusters may have differing perceptions of time without compromising the system MLS support for mission and safety critical components.

5. Both objects and transactions may migrate among specified clusters in the LANs and the WAN without compromising the integrity of the MLS system.

6. Both objects and transactions may be replaced among specified clusters in LANs and the WAN without compromising the integrity of the MLS system.

Advanced development is required in order to implement the test bed. A minimum team of four full time, top-level engineers for a period of six years is needed to provide the input for building the dynamic MLS portion of the test bed and integrating it to the other eleven components. A recommended schedule is to pursue the advanced development of the above items in three phases:

Phase 1	Items 1 and 2	Years 1 and 2
Phase 2	Items 3 and 4	Years 3 and 4
Phase 3	Items 5 and 6	Years 5 and 6

GLOSSARY

AJPO	Ada Joint Programming Office, U.S. Department of Defense
ANSI	American National Standards Institute
APSE	Ada Program Support Environment
CAIS	Common APSE Interface Set
DOD	Department of Defense
ISO	International Standards Organization
ID	Identification
JSC	Johnson Space Center
KAPSE	Kernel Ada Programming Support Environment
KIT	KAPSE Interface Team
MLS	Multilevel Security
NASA	National Aeronautics & Space Administration
NOS	Network Operating System
NOSC	Naval Ocean Systems Center
Orange Book	Trusted Computer System Evaluation Criteria
OSI	Open Systems Interconnection
SERC	Software Engineering Research Center
TCB	Trusted Computer Base
TNI	Trusted Network Interpretation
UH-CL	University of Houston at Clear Lake
WG	Working Group

BIBLIOGRAPHY

1) Anderson, Eric R. "Ada's Suitability for Trusted Computer Systems," IEEE Computer Society Reprint, from Proceedings of the Symposium on Security and Privacy, Oakland, California, April 22-24, 1985.

2) Borning, Alan. "Computer System Reliability and Nuclear War," Communications of the ACM, Vol. 30, Number 2, February 1987, p. 112-131.

3) Brown, Michael L., Chairperson. "Minutes of the Tri-Service Software Systems Safety Working Group, Fourth Annual Meeting." Naval Surface Weapons Center, Dahlgren, Virginia, July, 1986.

4) Covington, Clark, Project Manager. "Space Station Projects Requirements Document," JSC31000, March 6, 1987.

5) Department of Defense Computer Security Center: Department of Defense Trusted Computer System Evaluation Criteria. Fort Mead, Maryland, August 1983.

6) Department of Defense Computer Security Center: Department of Defense Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria. Fort Mead, Maryland, July 31, 1987.

7) Heaney, J. E. "Security Model Selection for Secure Microcomputer Network Prototype." PRC Government Information Systems, Planning Research Corporation, September 1984.

8) ISO: Basic Reference Model for Open Systems Interconnection. International Standards Organization Document 7498, 1983.

9) LeGrand, S. "An Access Control Model For A Distributed, CAIS-Conforming System." Ford Aerospace & Communications Corporation (Participating in the Joint NASA Johnson Space Center/University of Houston-Clear Lake APSE Beta Site Research), October 1985.

10) LeGrand, S., "Access Control For A Safety Critical Distributed System Interface Set." SofTech, Inc., Houston Operations, December 1987.

11) LeGrand, S. "An Open Systems Interconnection Proposed for the Joint NASA/UH-CL APSE Beta Test Site NOS Project" (Master's Thesis). University of Houston-Clear Lake, May 1984.

12) LeGrand, S. and McBride, J. "Why Ada Is Appropriate for The Space Station Program." SofTech, Inc., Houston Operations, March 1987.

13) McKay, C. W. "Distributed Networking of Automated Systems. University of Houston-Clear Lake, April 1984.

14) McKay, C. W., Auty, David and Rogers, Kathy, "Final Report on a Study of System Interface Sets for the Host, Target, and Integration Environments of the Space Station Program", University of Houston at Clear Lake, July 30, 1987.

15) McKay, C.W., "Lifecycle Support for Computer Systems and Software Safety in the Target and Integration Environments of the Space Station Program", University of Houston at Clear Lake, June 15, 1987.

16) Military Standard Common APSE Interface Set (CAIS), CAIS Working Group, October 9, 1986.

17) Perry, W. E. "Security Problems Are People Problems." Government Computer News, March 1987, p. 27-28.

18) Space Station Requirements Document, JSC 30000, NASA Space Station Projects Office, March 6, 1987.

19) Stewart, Bill. "Implementing Security Under Unix." Systems & Software, February 1986, p. 33-34.

20) Sweet, Frank. "How to Build a Security Chain." Datamation, February 1, 1987, p. 69-70.

21) Thall, Richard and Robert Wallace, "CAIS-A Issues Discussion", presented to KIT meeting April 7-9, 1987.

APPENDIX A

ACCESS CONTROL FOR A SAFETY CRITICAL DISTRIBUTED SYSTEM INTERFACE SET

Sue LeGrand

SofTech, Houston

The Space Station Program (SSP) requires a system that has many safety critical resources that must be kept secure. The Space Station Information System (SSIS) is defined as the integrated set of space and ground data and information networks which provide required data and information services to the flight crew, ground operations personnel, and the customer community. It includes as its elements not only flight element systems such as the onboard data management, communications and tracking systems; but also existing and planned institutional systems such as the NASA Communications System (NASCOM), the Tracking and Data Relay Satellite System (TDRSS) and the data and communications networks of the scientific and industrial users and the international partners. The SSIS is conceived to support the full range of users in all operations of their subsystems or experiments that involve data handling, processing and/or storage regardless of where each user is physically located. [18]

Dr. Dana Hall, Acting Director, SSP Information Systems Management Division, says that a user may be a subsystem monitor (human or automated expert system) in a control system, graduate student at a university conducting a space located experiment, or a crewperson correcting a manufacturing process in consultation with someone in the ground home factory. Through all of these activities on this large, complex, non-stop distributed system, the safety critical components must be secure. This paper will discuss the access control of these components and how this fits into a plan by the Software Engineering Research Center (SERC) for developing computer system interface sets (SISs) for these systems.

Space Station Safety

The Tri-Service Software Systems Safety Working Group composed the following definition: [3]

> Software systems safety is the optimization of systems safety in the design, development, use and maintenance of software systems and their integration with safety critical hardware systems in an operational environment.

Safety critical computer software components are defined as computer software components (processes, functions, values or program states) whose inadvertent occurrence, failure to occur

when required, occurrence out of sequence, occurrence in combination with other functions or erroneous value can result in a hazard or loss of resource predictability or control.

Software can be used in the following ways to safely control a system:

- Autonomous control
- Control of potentially hazardous hardware or system components
- Management of information requiring immediate operator action
- Management of information with which an operator or another system makes critical decisions

The Space Station is one of the most complex systems yet undertaken. This increases the risk of a combination of faults that individually may not be a problem, but combined can cause a serious threat. Safety planning must be incorporated into the system development effort from the beginning. This allows technical safety mechanisms to complement and augment rather than supplant the system design.

Space Station Access Control

An integral part of this Space Station system safety is access control of critical components. It must be considered from the very beginning of the system lifecycle and planned to be controlled in the domain of a security staff and a trusted computer system. Space Station systems may be characterized in the following ways:

- Parallel processing within a cluster attached to a local area network (LAN).
- Distributed and parallel processing among groups of LANs.
- Distributed and parallel processing among wide area networks (WANs) of integrated LANs and clusters.

All access granted to users must be monitored and controlled for the appropriate operations, resources and time slices. Access to safety and mission critical components of the Space Station should never be available at any time in the lifecycle to the hacker who inadvertently causes damage, the terrorist who maliciously seeks to cause destruction, or a disgruntled former staff member who leaves with potentially dangerous knowledge.

A special multi-level security model must be developed and prototyped that will fulfill the spectrum of safety requirements of the entire Space Station system and still permit the fulfillment of the mission requirements of its ground, on-orbit and co-orbit subsystems.

The Software Engineering Research Center (SERC) Model

In 1986 Ada* was designated as the computer language of choice for the Space Station Program (SSP). This was after recommendations of the ADa Programming Support Environment (APSE) Beta Test Team. This team composed of members from NASA/Johnson Space Center (JSC), University of Houston - Clear Lake (UH-CL) and over 30 participating NASA contractors concurred that Ada was the most appropriate language for the SSP system development and support. This support is expected to be over distributed, heterogeneous host and target systems, will evolve over a 30 year period and have an indefinite life cycle. Ada was designed to provide reduced cost primarily through portability of tools, software and programmers. From this APSE Beta Test Team effort grew the Software Engineering Research Center (SERC). It is located at University of Houston - Clear Lake and sponsored by NASA. Dr. Charles McKay is Technical Director.
In response to the strong commitment of NASA to the lifecycle support of safety of life and property, a SERC team has proposed a Clear Lake Model for Computer Systems and Software Safety in a Portable Common Execution Environment (PCEE). This model is built using the Ada computer language and is meant to provide the following:

- A baseline from which subsequent progress in the appropriate environments may be made.
- An extensible or compactable model which will improve safety in larger, more complex or smaller, more simple applications.
- A "lessons implied/learned" stimulus and opportunity to develop methodologies and tools which better address the lifecycle issues of safety. [14]

SSP Computer Environments

The SERC team reports that the SSP requires three distinct environments and associated activities. They are:

- A host environment where software for the target environment is developed and sustained.
- A target environment where the executable versions of the software developed in the host environment are deployed and operated.
- An integration environment where the configuration of the current target environment baseline is controlled.

Functional requirements such as productivity and phase management are the primary drivers in the host environment. Target environments are strongly influenced by non-functional requirements such as real time, fault tolerance, power and capacity constraints.

*Ada is the registered trademark of the US Government (AJPO).

The integration environment has been added to the traditional approach in order to facilitate scaling up of small models and creating more complex models from proven building blocks. This environment is the domain of those responsible for the test and integration plans used to interactively advance the target environment baseline with approved changes in software emanating from the host environments. It is used to prove that safety and security are considered in each component at all times. This environment is also used for controlling interactions with the target environment to maximize safety during emergencies.

Static and Dynamic Viewpoints

Two macroscopic perspectives are useful for understanding the requirements of the SSP environments. The first is the static viewpoint that encompasses all host environment system and software phases of development and support.

The second perspective is a dynamic viewpoint of program execution and crosses all three environments during all development and support phases. It considers such issues as the ability to sustain safety while fulfilling mission requirements. Related to this issue is the ability to assure access control of all safety critical components.

Space STation System Interface Sets

All three SSP environments have requirements for User Interface Sets (UISs) for human-system interface and System Interface Sets (SISs) for the interfaces of the application software and command language to the underlying system software and hardware resources. The SERC team effort concentrates on the SISs. The goal is to allow such things as tools, rules, application software, test software and command language scripts to be developed, acquired and supported on a foundation of virtual interface specifications. This fulfills the SSP requirement of avoiding dependence on the physical interface specifications of a particular operating system, data management system, communication system or instruction set architecture.

Two aids in fulfilling these requirements are the use of Entity Attribute/Relationship Attribute (EA/RA) models and the concept of SISs. An EA/RA model is one that represents domains of interest in:

The objects within each domain
Relevant attributes of these objects
Relationships among the objects
Relevant attributes of these relationships.

The objects may be either passive (e.g. data objects with no thread of execution of their own) or active (i.e. possess their own thread of execution). Further more the relationships are context sensitive (i.e. "normal" processing vs. "exception" processing) and may be between or among:

Active objects to active objects.
Active objects to passive objects.
Passive objects to passive objects.

An EA/RA model which is available on-line in the execution environment can be an enormous aid in controlling access between and among such objects. The draft International Standards Organization (ISO) standard, Information Resource Dictionary System (IRDS) is being studied by the SERC team for this purpose.[23] This standard defines means to:

Document the environment
Maintain an Inventory of Components
Provide a model of the environment
Support the operational aspects of the environment
Illustrate the interrelationship of components
Document the physical/logical location of components.

An overall stability for the SISs is obtained by fulfilling three requirements. First, from the perspective of an all-encompassing SIS is a virtual interface set resulting from the union of all abstract specifications of the objects designed to be visible at this interface.

Second, from the perspective of any given object within a particular SIS, there should be a formal model in the EA/RA form for the proper visibility among all objects and a grouping of objects into discreet sets with associated services and classes of services.

Third, from the perspective of external objects outside each SIS, a formal model in EA/RA form is needed to describe external interfaces to services and access control of services and resources.

A Common APSE Interface Set (CAIS) has been defined to describe the interface in a host environment of all Ada development tools and all operating systems available to each host. The SERC team recommends the adoption of the CAIS as an extensible baseline for the SIS of the host and integration environments. It reflects only the static viewpoint of tool builders and administrators, but it is an excellent beginning and fits in as a subset of the overall SIS.

A Portable Common Execution Environment

Throughout the iterative, dynamic evolutionary life cycle of each system of the SSP, the software in various forms, versions or representations will reside and migrate among the three computer environments. A portable common execution environment (PCEE) is needed for a solution to the challenge of productivity and support of systems requirements and safety. The proper PCEE model will facilitate the management of the life cycle complexity of software systems in a similar manner across all three environments.

Each unit would have a version of the PCEE that is a stable interface set described by Ada packages selected from a common run time library. The Ada language has this package structure to provide modularity for just this kind of application. A PCEE is defined to consist of:

> A set of policies for the management of services and resources to be provided to the application programs,
> The set of management modules to enforce the policies, and
> A set of rules for modification and extension.

The framework of the entire PCEE must be flexible enough to be represented in a number of ways in order to accommodate tailoring and extension in a large number of implementations. The PCEE must also support the differing operational requirements of the three environments. It should hide underlying system software implementations ranging from a tailorable bare operation in a target to a tailorable operating system in a host to a general purpose run time system that is a combination of the first two environments.

It is the goal of the SERC team that the PCEE support the following assertion in regards to safety:

> If safety is adequately addressed in the host environments throughout development and acquisition
> then the support of this explicitly addressed set of safety specifications will be dependent upon a run time environment built to:
>
> Monitor the system and detect faults that enter the system state vectors as soon as possible.
> Firewall their propagation
> Analyze their effects
> Recover safely. [14]

Each run time environment must be secured in order to assure that the above will always be available. This run time assertion is dependent upon secure and safe context sensitive access control.

The SERC team has produced the "Clear Lake Model for Life Cycle Support of Computer Systems and Software Safety in the Target and Integration Environments of the Space Station Program". [14]

The effort began by studying the safety issues and the models, methods and tools currently purported to address these issues. From this study the team identified requirements for a "safety kernel" of key components of an execution environment underlying a PCEE.

Twelve highly interdependent models of key components in a System Interface Set of a PCEE are being investigated and/or developed. They are:

- Interface features and options for a tailorable run time support environment.
- Software structures which facilitate firewalling, layered recovery capabilities, dynamic reconfiguration and extensibility for fault tolerance.
- Pools of processes and processors capable of non-stop operation in a fault tolerant environment.
- A multiversion, fault tolerant programming capability.
- Command language interface between the SISs of the integration and target environments' PCEE.
- System-wide life-cycle-unique identification and history of all objects and transactions.
- Dynamic, multilevel security in the target and integration environments (B3 class integrity requirements and beyond)
- A message interface supporting communication among clusters.
- Hierarchical run time structure of the threads of control of each program.
- A redundancy management subsystem for safety critical services and resources.
- A stable storage subsystem for each cluster.
- A management subsystem for distributed, nested transactions.

All of the above component models are intended to reside in safety kernels which execute in privileged mode beneath the SIS of processors participating in the PCEE. Note also that the dynamic, multilevel security model is only one of the twelve components. In other words, security is a means to the end of simultaneously supporting both mission and safety requirements, rather than an end in itself.

Space Station Security

The security component of the PCEE is based on a model presented at the AIAA/ACM/NASA/IEEE Computers in Aerospace V Conference in October, 1985. [10] It presented a "superset" of building blocks for enforcing multilevel security in a distributed target environment. It is based on subjects accessing objects. Both the subjects and objects have attributes, and the method involves a comparison of these attributes in a dynamically changing, heterogeneous distributed system. Subjects may be users via their active processes; and the objects may be data, devices and other processes. The system must not only meet immediate security requirements but must react to current capability

limitations such as small memory space or different processor speeds. The software for this model is to be written in Ada, and the model complies with the Orange Book, Open System Interconnection (OSI) and CAIS standards.

Ada is especially suited to the needs of this security model because it is strongly typed. [12] Each object named in a program has the traditional attribute of value (constant, variable) and also the attribute of type. The type attribute defines the kinds of values and operations allowed to be associated with the object. The Ada language requires that all objects be typed and enforces this requirement during compilation. If an untyped object is discovered or if the wrong value or operation is associated with the object, an error is reported. All must be in order before the program is executed. Any secure resource may be named as an object and only allowed operations associated with each one.

The EA/RA structure of the PCEE can be used to passively represent a design structure depicting the relevant objects, their relationships and the key attributes that must be compared for access control. Each object embodies the software engineering principles of abstraction, modularity, information hiding and localization needed for security enforcement. Further more the on-line use of EA/RA structures can be used actively by the PCEE processes to enforce safety and integrity constraints for access control in the execution environment.

The SSP requires multilevel security of the Orange Book (Trusted Computer System Evaluation Criteria) Level 3 and below. This trusted computer base involves both discretionary and mandatory access control. It must satisfy the reference monitor requirements that it mediate all access of subjects to objects, be tamperproof, and be small enough to be subjected to analysis and tests. A security administrator is supported, audit mechanisms are expanded to signal security - relevant events and system recovery procedures are required.

The Puce Book (Trusted Network Evaluation Criteria) is expected to offer guidelines applicable to the PCEE model when it is available. [6]

The Open Systems Interconnection (OSI) Model provides standards for the exchange of information among systems that are "open" to another for this purpose by virtue of their mutual use of the applicable standards. A system is defined in this international standard as it is in the PCEE Model. [9]

The purpose of the OSI is to define a set of standards to enable open systems to communicate and cooperate. It uses a layered architecture for operations such as intersystem connections, transmission data and error functions. This model is mandated for the design of a network operating system for the SSP.

Proposed Prototype

Before proposing a security prototype effort, a study was made of the above standards, the SERC PCEE model and the SSP specifications for a Data Management System (DMS). Careful note was made of DMS requirements that had security implications, especially in safety critical components. Some of these requirements are:

- Override capability of automatic systems is required for authorized onboard and ground crews.
- A User Interface Executive must keep track of which user is linked to which application tool.
- Authorized emergency priority messages must be forwarded.
- Command differentiation is to be suitable for managing operations and to protect the system from inadvertent and unauthorized payload operations.
- Local and remote access to onboard DMS databases must be consistent with security and privacy practices.

To facilitate research for a multilevel secure system for the Space Station, a minimal configuration for proof of concept for a reconfigurable testbed is proposed to NASA. It has the following characteristics and capabilities:

There will be a minimum of 3 local area networks (LANs), each consisting of at least 3 clusters of processors. The LANs will be geographically separated and communicate over a wide area network (WAN) that is OSI compatible.

In each cluster in each LAN there will be at least three processors, all of which can cooperate in parallel processing within the cluster.

There will be the capability of distributed, parallel processing among the groups of LANs.

There will be the capability of distributed, parallel processing among the LANs in the wide area network (WAN).

A single, unified network operating system will be the sum of the individual run time support environments in the clusters. It will provide the following for the prototype:

- A set of policies for the management of services and resources to be provided to the application programs.
- The set of management modules to enforce the policies.
- A set of rules for modification and extension.

This network will be able to support multiprogramming within and among multiprocessors. That is, one processor will be able to simultaneously handle more than one program and one program will be able to be executed with more than one processor. Also, there will be the capability to have multitasking within a program and across multiple programs and processors.

Each cluster will be built upon the twelve components of the SIS of the Clear Lake Model for a PCEE and will consist of a set of components from these four classes:

Processors with shared access to memory subsystems
Memory subsystem with stable and volatile storage
Communication links to other clusters
Sharable services and resources.

There will be a common sense of timing throughout the WAN. That is, each system may have its own timing mechanism and granularity, but it must be able to interpret time stamps from the other systems.

A PCEE written in Ada will be the only environment type used in the development of each component of each cluster. No untyped language may be accessible to provide a trap door to the operating system, hardware or other safety critical resources. Hardware representation specifications will be restricted to those individually approved and monitored by the security administrator.

In order to control access to safety critical components, each node will have only a PCEE-compliant operating system and access only to PCEE-compliant tools. All user interfaces will be through the PCEE only.

Code will be written in Ada to demonstrate the scenarios in the paper entitled "Access Control Model for a Distributed, CAIS-Conforming System". It will include access control attributes for each entity and appropriate relationship in the system. It will include programs to provide access control management for Orange Book level B3 and beyond. It will rely on the existing network, database and applications management routines in the prototype systems.

Summary

The Clear Lake Model for a System Interface Set of a Portable Common Execution Environment is believed to fulfill the requirements for System Interface Sets that protect the safety of critical components of the Space Station orbiting, coorbiting and ground systems and the interaction of these systems while simultaneously supporting mission requirements. A dynamic, multilevel security model has been designed as a part of the PCEE to assure access control of these safety critical components as well as privacy for scientific or user-proprietary data. A prototype of the PCEE model has been proposed to NASA in order to prove the concepts presented here.

Acknowledgment

Dr. Charles McKay is the team leader of the SERC team which developed the integrated model for the assurance of computer systems and software safety of which this security model is a part. He has been a primary consultant to NASA/JSC since the beginning of the SSP. Groups interested in more information about the SERC programs are invited to contact him at the University of Houston - Clear Lake.

BIBLIOGRAPHY

1) Anderson, Eric R. "Ada's Suitability for Trusted Computer Systems," IEEE Computer Society Reprint, from Proceedings of the Symposium on Security and Privacy, Oakland, California, April 22-24, 1985.

2) Borning, Alan. "Computer System Reliability and Nuclear War," Communications of the ACM, Vol. 30, Number 2, February 1987, p. 112-131.

3) Brown, Michael L., Chairperson. "Minutes of the Tri-Service Software Systems Safety Working Group, Fourth Annual Meeting." Naval Surface Weapons Center, Dahlgren, Virginia, July, 1986.

4) Covington, Clark, Project Manager. "Space Station Projects Requirements Document," JSC31000, March 6, 1987.

5) Department of Defense Computer Security Center: Department of Defense Trusted Computer System Evaluation Criteria. Fort Mead, Maryland, August 1983.

6) Department of Defense Computer Security Center: Department of Defense Trusted Network Communication Evaluation Criteria. Fort Mead, Maryland (draft due Fall 1987.)

7) Hall, Dana, NASA Space InformationSystems Overview, AIAA-87-2189, AIAA/NASA International symposium on Space Information Systems in the Space Station Era, Washington, D.C., June 1987.

8) Heaney, J. E. "Security Model Selection for Secure Microcomputer Network Prototype." PRC Government Information Systems, Planning Research Corporation, September 1984.

9) ISO: Basic Reference Model for Open Systems Interconnection. International Standards Organization Document 7498, 1983.

10) LeGrand, S. "An Access Control Model For A Distributed, CAIS-Conforming System." Ford Aerospace & Communications Corporation (Participating in the Joint NASA Johnson Space Center/University of Houston-Clear Lake APSE Beta Site Research), October 1985.

11) LeGrand, S. "An Open Systems Interconnection Proposed for the Joint NASA/UH-CL APSE Beta Test Site NOS Project" (Master's Thesis). University of Houston-Clear Lake, May 1984.

12) LeGrand, S. and McBride, J. "Why Ada* Is Appropriate for The Space Station Program." SofTech, Inc., Houston Operations, March 1987.

13) McKay, C. W. "Distributed Networking of Automated Systems." University of Houston-Clear Lake, April 1984.

14) McKay, C.W., David Auty and Kathy Rogers, "Final Report on A Study of System Interface Sets for the Host, Target, and Integration Environments of the Space Station Program", University of Houston at Clear Lake, July 30, 1987.

15) McKay, C.W. "Lifecycle Support for Computer Systems and Software Safety in the Target and Integration Environments of the Space Station Program", University of Houston at Clear Lake, June 15, 1987.

16) Perry, W. E. "Security Problems Are People Problems." Government Computer News, March 1987, p. 27-28.

17) Proposed Military Standard Common APSE Interface Set (CAIS), CAIS Working Group, January 1985.

18) Space Station Information System Definition Document, Space Station Program Office, Washington, D.C., July 1986.

19) Space Station Requirements Document, NASA Space Station Projects Office, March 6, 1987.

20) Stewart, Bill. "Implementing Security Under Unix." Systems & Software, February 1986, p. 33-34.

21) Sweet, Frank. "How to Build a Security Chain." Datamation, February 1, 1987, p. 69-70.

22) Thall, Richard and Robert Wallace, "CAIS-A Issues Discussion", presented to KIT meeting April 7-9, 1987.

23) Winkler, Dr. A.J. "The IRDS", Presented to the Joint Ada Conference, Washington, D.C., March 18, 1987.