

AD-A253 204



**RL-TR-92-9, Vol I (of three)
Final Technical Report
January 1992**



ASSURED SERVICE CONCEPTS AND MODELS Summary

Secure Computing Technology Corporation

**Sponsored by
Strategic Defense Initiative Office**

**DTIC
ELECTE
JUL 24 1992
S A D**

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

92-19832



The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Strategic Defense Initiative Office or the U.S. Government.

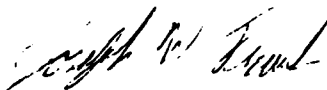
**Rome Laboratory
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700**

92 7 22 029

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-92-9, Vol I (of three) has been reviewed and is approved for publication.

APPROVED:



JOSEPH W. FRANK
Project Engineer

FOR THE COMMANDER:



JOHN A. GRANIERO
Chief Scientist
Command, Control & Communications Directorate

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL (C3AB), Griffiss AFB NY 13441-5700. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

ASSURED SERVICE CONCEPTS AND MODELS
Summary

J.T. Haigh
R.C. O'Brien
W.T. Wood
T.G. Fine
M.J. Endrizzi
S. Yalamanchili

Contractor: Secure Computing Technology Corporation
Contract Number: F30602-90-C-0025
Effective Date of Contract: 23 February 1990
Contract Expiration Date: 22 February 1991
Short Title of Work: Assured Service Concepts and Models
Period of Work Covered: Feb 90 - Feb 91

Principal Investigator: Tom Haigh
Phone: (612) 482-7400

RL Project Engineer: Joseph W. Frank
Phone: (315) 330-2925

Accession For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

Approved for public release; distribution unlimited.

RECEIVED INSPECTED 2

This research was supported by the Strategic Defense Initiative Office of the Department of Defense and was monitored by Joseph W. Frank, RL (C3AB) and Emilio J. Siarkiewicz, RL (C3AB) Griffiss AFB NY 13441-5700 under Contract F30602-90-C-0025.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave Blank)		2. REPORT DATE January 1992		3. REPORT TYPE AND DATES COVERED Final Feb 90 - Feb 91
4. TITLE AND SUBTITLE ASSURED SERVICE CONCEPTS AND MODELS Summary			5. FUNDING NUMBERS C - F30602-90-C-0025 PE - 63223C PR - 3109 TA - 01 WU - 01	
6. AUTHOR(S) J. T. Haigh, R. C. O'Brien, W. T. Wood, T. G. Fine, M. J. Endrizzi, S. Yalamanchili				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Secure Computing Technology Corporation 1210 West County Road, East Suite 100 Arden Hills MN 55112			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Rome Laboratory (C3AB) Griffiss AFB NY 13441-5700			10. SPONSORING/MONITORING AGENCY REPORT NUMBER RL-TR-92-9, Vol I (of three)	
11. SUPPLEMENTARY NOTES Rome Laboratory Project Engineers: Joseph W. Frank/C3AB (315) 330-2925 Emilie J. Siarkiewicz/C3AB (315) 330-3241				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This report describes the work performed under the Assured Service Concepts and Models (ASCM) contract. The report is organized as follows. Volume I is a summary of all of the work done in the ASCM project. Volume II describes the various security policies that were developed on the contract. Volume III describes the availability policies that were developed on the contract and the approaches that were developed for identifying trade-offs between secrecy and availability. Volume III also contains the findings of the formalism study.				
14. SUBJECT TERMS Computer Security, Assured Service, Availability, Distributed Systems			15. NUMBER OF PAGES 28	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT U/L	

PREFACE

This report describes the work performed under the Assured Service Concepts and Models (ASCM) contract. The work was supported by Rome Laboratory under the ASCM contract (contract no: F30602-90-C-0025). The prime contractor is the Secure Computing Technology Corporation (SCTC) and the sub-contractor is the Georgia Tech Research Corporation (GTRC). This report overviews all of the work performed on the contract.

The ASCM project began in April 1990; the technical work was completed in May 1991.

The SCTC team members were Mike Endrizzi, Todd Fine, Tom Haigh, Richard O'Brien, and Bill Wood. The GTRC team member was Sudhakar Yalamanchili.

CONTENTS

Section	Page
1 Introduction	3
1.1 Objective	3
1.2 Accomplishments	3
1.3 Documents	4
2 Formalisms Study	5
3 Secrecy Policy Development	6
4 Availability Model Development	8
5 Trade-Off Study	10
6 Research Topics	11
7 Conclusion	12

SECTION 1

INTRODUCTION

This document summarizes the work performed on the ASCM project. Details are provided in the attached volumes.

1.1 OBJECTIVE

The objective of the ASCM project was to develop approaches for analyzing secure, distributed C^2 systems. The objective was met by completing the following tasks:

- a. Perform a study to determine the secrecy and availability needs for distributed C^2 systems and the relationship between secrecy and availability in distributed C^2 systems.
- b. Perform a study to determine the appropriate modeling formalism to use for the analysis of distributed C^2 systems with respect to secrecy and availability.
- c. Develop adaptive secrecy policies that are appropriate for analyzing distributed C^2 systems.
- d. Develop availability models that are appropriate for analyzing distributed C^2 systems.
- e. Examine approaches for identifying trade-offs that must be made between secrecy and availability in distributed C^2 systems.

1.2 ACCOMPLISHMENTS

The ASCM project has achieved a number of significant accomplishments. They include:

- a. Identifying the ways that various availability mechanisms both complement and conflict with secrecy policies.
- b. Identifying the advantages and disadvantages of using various modeling formalisms.
- c. Clarifying the relationships among various security policies.
- d. Identifying deficiencies in previously developed information flow policies for nondeterministic systems.
- e. Developing adaptive security policies.
- f. Demonstrating that composability is a requirement for security policies rather than only a desirable property.
- g. Identifying deficiencies in proposed approaches for using composability to significantly simplify a security analysis.
- h. Developing an approach for formally analyzing fault tolerance mechanisms.

- i. Developing liveness policies that prohibit deadlock, starvation, and mutual starvation.
- j. Developing a worked example of the analysis of a real-time system.
- k. Developing approaches for identifying trade-offs between secrecy and availability in distributed C^2 systems.

We discuss each of these results in more detail in the following sections.

1.3 DOCUMENTS

The ASCM final report is organized as follows. Volume 1 (this document) is a summary of the report. It summarizes all of the work done on the ASCM project. Volume 2 (CDRL A005) describes the various security policies that were developed on the contract. Volume 3 (CDRL A004) describes the availability policies that were developed on the contract and the approaches that were developed for identifying trade-offs between secrecy and availability. Volume 3 also contains the findings of the formalism study.

SECTION 2

FORMALISMS STUDY

Volumes 2 and 3 contain the results of the formalisms study task.

Volume 2 describes how a deficiency in modeling formalisms has greatly complicated the formalization of secrecy in an information flow policy. The deficiency is that modeling formalisms typically ignore causality. For example, while it is easy to specify that a particular output can never come before a particular input, it is difficult to specify that the input causes the output. The relevance to information flow policies is that this deficiency makes it very difficult to distinguish high-level events that are caused by low-level processes from other high-level events. Informally, an information flow policy requires that:

Actions taken by high-level processes are not visible to low-level processes.

This policy is formalized by requiring that the results visible at the low-level are the same regardless of whether the high-level actions occur. The process of removing the parts of the execution history caused by high-level actions is referred to as *purging* the high-level actions. A high-level action consists of the set of events that it causes. Thus, we must determine which events are caused by a high-level action before we can purge the action. Because it is difficult to tell which events are caused by a high-level action, it is difficult to correctly purge high-level actions. Volume 2 argues that the many failures in developing an information flow policy for nondeterministic systems are a result of the lack of a notion of causality. Further research into this area would greatly benefit the theory of information flow policies for nondeterministic systems. This issue is related to the discussion of composability in the next section.

Volume 3 considers the following formalisms in more detail: state machines, traces (and more generally, CSP), Petri nets, temporal logic, interval temporal logic (ITL), and real time logic (RTL). The advantages and disadvantages of each formalism are discussed and a list is provided of the problem domains to which each is applicable. CSP was found to be the most useful formalism for distributed C^2 systems except for the case in which the system must be analyzed with respect to a real-time policy. Since there does not appear to be any way to "correctly" incorporate real-time into CSP, some other formalism must be used for real-time policies. Some peculiarities in the semantics of RTL make it difficult to use, and non-trivial ITL specifications are very difficult to understand. Thus, neither ITL nor RTL appear useful for addressing real-time policies. Instead, we found timed state predicates, a variant of RTL, to be the most useful.

SECTION 3

SECURITY POLICY DEVELOPMENT

Volume 2 contains an informal description of a simple C^2 system. This system is used to motivate the discussion of security policies. Volume 2 also discusses the various security policies that are currently in use. Where possible, relationships between policies are described. In particular, a description is given of how newer policies address the deficiencies in previous policies.

Since distributed C^2 systems are nondeterministic to a certain degree, security policies for nondeterministic systems must be developed before security policies can be developed for distributed C^2 systems. Volume 2 describes deficiencies in current formalizations of information flow policies for nondeterministic systems and proposes a series of new definitions that address these deficiencies. Two areas that must be researched further are:

- a. The use of stochastic information flow policies to address noisy covert channels.

Although we propose a stochastic information flow policy, further research is required to determine the feasibility of using it to analyze a real system.

- b. Viewing computer systems as self-evolving systems rather than assuming that requests are generated external to the system.

The behavior of a subject in a computer system is defined by the subject's code object and the system's scheduling policy rather than by actions external to the system. Since information flow policies usually ignore the connection between a subject's code object and the system's scheduling policy and the instructions the subject executes, it is possible that a system might satisfy an information flow policy while still allowing a covert channel through executable objects or the scheduling policy. Further research is required to develop a formal security policy that addresses this deficiency.

After developing a firm foundation for security policies, Volume 2 discusses adaptive security policies. An adaptive security policy is one that addresses special operations that violate the letter of an MLS policy. Examples include:

- a. The reclassification of processes and data.
- b. Reconfiguration of a system.
- c. Broadcast messages across levels.
- d. Change of operational mode.

The violations can be separated into two classes, those that can actually compromise the sensitive information and those that cannot. The second class consists of trusted subjects whose design prevents them from disclosing information at an inappropriate level even though they have privilege to downgrade information. The former class contains the rest of the trusted subjects.

Volume 2 describes a policy that can be used to incorporate the analysis of the second class of trusted subjects with the analysis of the untrusted subjects and to clearly identify that the remaining trusted subjects are exceptions that must be analyzed using some other means. Since this approach is much more unified than the traditional approach, it provides a more complete and believable analysis. Volume 2 also discusses how this approach can be simplified if the system being analyzed enforces a role enforcement policy such as Type Enforcement or Clark-Wilson policy. Role enforcement policies provide the capability of constraining trusted subjects based on their role in the system. For example, suppose a system contains a subject *D* that is trusted to downgrade information after it has sanitized it. Since *D* is trusted to violate the system's MLS policy, the MLS policy places no constraints on *D*. Thus, if the system only enforces an MLS policy, the system has no control over *D*'s actions. On the other hand, if the system enforces a role enforcement policy, then the system can enforce the policy that *D* must sanitize information before downgrading it.

Volume 2 also considers the relevance of composability to secrecy policies. Previous work has suggested that while composability is desirable it is only necessary when a system is constructed by combining components that are individually shown to be secure. Our findings were that any policy that is not composable is seriously flawed. The notion of composability was developed to address flaws that were observed in existing information flow policies. Volume 2 shows that these flaws were the result of the underlying formalisms not providing a notion of causality. The noncomposability of the policies is a consequence of ignoring causality. The importance of these findings is that they suggest that future research should be directed at addressing causality rather than at developing composable policies.

In addition to raising doubt as to the theoretical importance of composability, Volume 2 also questions its practical importance. Earlier work has suggested that a complex system can be demonstrated to be secure by demonstrating that it is a composite of pieces that satisfy a composable security policy. There are two problems with using this approach.

- a. Policies are composable with respect to a specific method for composing systems. If the implementation of a system uses a different method for composing systems, then the composability argument is no longer valid.
- b. In the process of decomposing the system, a level is reached at which the components are no longer secure in isolation; instead, the components constrain each other so that the overall system is secure.

SECTION 4

AVAILABILITY MODEL DEVELOPMENT

Volume 3 describes our approach to analyzing a distributed C^2 system with respect to availability. The approach is to:

- a. Develop a specification of the system ignoring the possibility of faults.
- b. Determine the set of faults to be tolerated and the effects of each fault.
- c. Extend the specification of the system to address the possibility of faults.
- d. Demonstrate that the system satisfies our fault tolerance policy.
- e. Demonstrate that the CSP specification that ignores the possibility of faults satisfies any of our availability policies that are desired.

Our fault tolerance policy requires that the input-output behavior of a service be unaltered by the occurrence of faults. Thus, we can reduce the analysis of the system to simply analyzing the non-faulty behavior of the system by showing the system is fault tolerant. Although our fault tolerance is similar in spirit to previously proposed fault tolerance policies, it addresses deficiencies present in earlier work.

We also considered two ways to weaken our fault tolerance policy to obtain a graceful degradation policy. The first way is to allow the set of faults with respect to which the system is fault tolerant to decrease with time. The correspondence between this situation and graceful degradation of service is that the system's ability to tolerate faults degrades with time. The second way to weaken our policy is to allow faults to alter the input-output behavior, but to require that the faulty behavior be "similar" to the nonfaulty behavior. Since the system behavior is altered by faults, it is degraded. The degradation is graceful in the sense that the faulty behavior must be "similar" to the nonfaulty behavior. Further research is needed to complete the formalization of these forms of graceful degradation and determine their applicability to realistic systems.

Volume 3 discusses CSP formalizations of policies prohibiting deadlock, starvation, and mutual starvation. In addition, Volume 3 describes how concepts such as temporal dependence, eventuality, livelock, and fairness can be formalized in CSP. Although the policies were developed in the context of deterministic systems, we propose generalizations to nondeterministic systems. Informally, the policies are:

- Deadlock Policy

Given any set of processes that are not permitted to deadlock, whenever there are events e and f such that one of the processes can participate in e before f , all of the processes in the set must participate in e before f .

Otherwise, deadlock would occur since one process would attempt to perform e before f while another process would attempt to perform f before e .

- Starvation Policy

Given any two processes P and Q such that P is not permitted to starve Q , P must interact with Q whenever Q requests interaction.

Otherwise, Q would be indefinitely blocked waiting for service from P .

- Mutual Starvation Policy

Given any two processes P and Q that are not permitted to starve each other, whenever P and Q interact, they do so in a consistent manner.

Otherwise, P and Q would each be indefinitely blocked waiting for the other to use the proper protocol for the interaction.

If absolute availability is required, then the sets of processes permitted to deny service are defined to be empty. For example, by specifying that no sets of processes are permitted to be deadlocked, the deadlock policy prohibits any deadlock from occurring in the system. On the other hand, a policy that only prohibits system services from becoming deadlocked, while allowing user processes to become deadlocked, would be obtained by defining the sets of processes that are permitted to be deadlocked so that none of them include any system services.

Further work is necessary to demonstrate the validity of these policies and to develop policies for other classes of availability concerns.

Since real-time policies are more application dependent than liveness policies, Volume 3 provides a worked example of the analysis of a simple real-time system rather than a general discussion of real-time policies. The system is an elevator control system that was developed by SRI using the ITL specification language. We found timed state predicates to be a much more useful specification for the elevator example. In addition to discovering some significant errors in the ITL specification, we were also able to provide an argument that the elevator satisfied its service policy. In contrast, the SRI work only included a specification of the elevator and performed no analysis of the model. Although our work with real-time policies was specific to the elevator example, it appears reasonable to adapt the approach to address real-time policies for other systems. Further research is needed to determine whether this actually is feasible.

SECTION 5

TRADE-OFF STUDY

Volume 3 also discusses approaches that can be used to identify trade-offs between secrecy and availability.

We found that the most reasonable way to address trade-offs between secrecy and availability is to weaken the respective policies to obtain a policy that clearly:

- a. Identifies the conflicts between secrecy and integrity,
- b. Identifies the degree of secrecy and integrity that holds in each case of conflict,
- c. Identifies the absolute policies that hold when there are no conflicts.

For example, the policies developed in Volume 2 can be used to define exactly which system operations violate the secrecy policy. A complete policy can be obtained by extending the policy to define the permissible actions the system can take for each of the exceptions. Other ways of weakening policies to remove conflicts include stochastic policies and the concept of *effectively ignoring*. For example:

- By using a stochastic policy, it is possible to have the secrecy policy allow noisy covert channels while prohibiting noiseless covert channels. A common approach to addressing conflicts between secrecy and availability is to introduce an availability mechanism at the expense of a noisy covert channel. By using a stochastic policy it is possible to demonstrate that the channel is noisy and to determine the amount of noise present.
- If the purging of high-level actions is defined to only ignore certain parts of high-level actions rather than all high-level actions, then we say that the high-level actions are effectively ignored rather than completely ignored. If a system can be shown to satisfy an information flow policy with this weaker notion of purging, then the system is demonstrated to only have information flow through the parts of the high-level actions that were not purged. By more accurately identifying the source of the illicit information flow, further analysis of the system is simplified.

SECTION 6

RESEARCH TOPICS

In this section we list issues that the project did not completely address:

- a. Although we identified the lack of any notion of causality as a deficiency in many modeling formalisms, we failed to identify a formalism that addressed the deficiency. After finding such a formalism, it would be interesting to use it to state an information flow policy and see whether all of the problems present in current information flow policies are addressed.
- b. Our policy that views a system as self-evolving rather than responding to external requests must be formalized.
- c. Our policies for graceful degradation of service must be formalized.
- d. The original plan for the ASCM project called for research into the composition of heterogeneous security policies. We intended to address this by working examples of general instances of the composition of heterogeneous systems. This would have resulted in a library of generic examples that could be used when analyzing a specific system. We did not have time to perform this work on the contract.
- e. We developed as many service policies as time permitted, but there are still classes of availability policies that are not addressed.
- f. The original plan for the ASCM project called for the policies and models developed on the project to be applied to the THETA-DOS operating system. This would provide validation of the approach and serve as a guide for future efforts to develop and analyze secure, distributed C² systems. Unfortunately, there was not enough time on the contract to perform this analysis. It is important that the policies and models be validated by using them to analyze a moderately complex system. In particular, the following policies and models should be applied to such a system:
 - (1) Our adaptive information flow policy,
 - (2) Our stochastic information flow policy,
 - (3) Our policy viewing a system as self-evolving rather than responding to external requests,
 - (4) Our fault tolerance and graceful degradation policies,
 - (5) Our deadlock, starvation, and mutual starvation policies,
 - (6) Our approach for analyzing systems with respect to real-time policies.

The approaches we have identified for performing trade-offs between secrecy and availability should be used to address any conflicts that arise during the analysis.

SECTION 7

CONCLUSION

The ASCM project has met its goal of developing an approach for analyzing distributed C² systems with respect to both secrecy and availability. In addition to pulling earlier work together into a unified approach, the project addressed deficiencies that were present in the earlier work by defining new policies and models. The most serious deficiencies in the work performed on the contract are:

- More realistic examples are needed to validate the approach developed.
- More classes of policies and models need to be defined.

DISTRIBUTION LIST

addresses	number of copies
RL/C3AR ATTN: Joseph W. Frank Griffiss AFB NY 13441-5700	12
Secure Computing Technology Corp 1210 West County Road E, Suite 100 Arden Hills MN 55112	5
RL/DOVL Technical Library Griffiss AFB NY 13441-5700	1
Administrator Defense Technical Info Center DTIC-FDAC Cameron Station Building 5 Alexandria VA 22304-6145	2
Strategic Defense Initiative Office Office of the Secretary of Defense Wash DC 20301-7100	2
RL/C3AR Griffiss AFB NY 13441-5700	1
HQ USAF/SCIT Washington DC 20330-5100	1
HQ SAC/SCPT OFFUTT AFB NE 68046	2

AFIT/LDEF
Building 642, Area B
Wright-Patterson AFB OH 45433-6523

1

AUL/LSE
Bldg 1405
Maxwell AFB AL 36112-5564

1

HQ ATC/TTOT
ATTN: Lt Col Killian
Randolph AFB TX 76150-5001

1

US Army Strategic Def
CSSD-IM-PA
PO Box 1500
Huntsville AL 35807-3301

1

Commanding Officer
Naval Avionics Center
Library D/765
Indianapolis IN 46219-2189

1

Commanding Officer
Naval Ocean Systems Center
Technical Library
Code 9642B
San Diego CA 92152-5000

1

Cmdr
Naval Weapons Center
Technical Library/C3431
China Lake CA 93555-6001

1

Superintendent
Code 524
Naval Postgraduate School
Monterey CA 93943-5000

1

Los Alamos National Laboratory
Report Library
MS 5000
Los Alamos NM 87544

1

AEDC Library
Tech Files/MS-100
Arnold AFB TN 37339

1

SEI JPD
ATTN: Major Charles J. Ryan
Carnegie Mellon University
Pittsburgh PA 15213-3890

1

Director
NSA/CSS R12
ATTN: Mr. Dennis Heinbuch
9800 Savage Road
Fort George G. Meade MD 20755-6000

1

DoD
R31
9800 Savage Road
Ft. Meade MD 20755-6000

1

DIPNSA
R509
9800 Savage Road
Ft Meade MD 20775

1

Director
NSA/CSS
ROR/R 2 = 3LDE
Fort George G. Meade MD 20755-6000

1

DDO Computer Center
C/TIC
9800 Savage Road
Fort George G. Meade MD 20755-6000

1

DCMAO/GWF
ATTN: JOHN CHENG
US COURTHOUSE/SUITE B-34
401 N MARKET
WICHITA KS 67202-2095

1

Defense Technology Sec Admin (DTSA) ATTN: STTD/Patrick Sullivan 400 Army Navy Drive Suite 300 Arlington VA 22202	1
DIA NSA/CSS P23 ATTN: Michael P. Ware 9800 Savage Rd. Ft. George G. Meade MD 20755-6000	1
Secure Computing Technology Corp ATTN: Jerry A. Herby 1210 West County Road E, Suite 100 Arden Hills MN 55112	1
Unisys Corp/Network Info Sys Div ATTN: Lorraine Martin 5151 Camino Ruiz Camarillo CA 93010	1
Trusted Information Systems, Inc. ATTN: Richard E. Schwinger P.O. Box 45 3060 Washington Rd. Glenwood MD 21738	1
AFSC/CV-0 Andrews AFB MD 20334-5000	1
ESD/AVC Hanscom AFB MA 01731-5000	1
Naval Research Laboratory Code 5540J ATTN: H. O. Lubbes Washington DC 20375-5000	1
DARPA/ISTO ATTN: LtCol Brian Poesch 1400 Wilson Blvd Arlington VA 22209-2303	1

SRI International
ATTN: Darlene Sherwood
FOR: Comp Sci Lab/Peter G. Neumann
333 Ravenswood Ave
Menlo Park CA 94025

1

ORA Corporation
ATTN: THETA Group
301A Harris B. Dates Dr.
Ithaca NY 14850-1313

1

Naval Ocean Systems Center
Code 413
ATTN: Les C. Anderson
271 Catalina Blvd.
San Diego CA 92152-5000

1

Computational Logic Inc.
ATTN: Lawrence M. Smith
1717 West Sixth St, Suite 290
Austin TX 78703-4776

1

BBN Systems and Technologies
ATTN: James C. Berets
10 Moulton St
Cambridge MA 02138

1

Concurrent Computer Corp
ATTN: Raymond K. Clark
One Technology Way
Westford MA 01886

1

BBN Systems and Technologies
ATTN: Carl D. Howe
10 Moulton St
Cambridge MA 02138

1

Marvin Schaefer
3657 Sharp Rd
Glenwood MD 21738

1

Honeywell Inc
Sensor & System Dev Center
ATTN: Ronald K. Crowe (M465-2300)
3660 Technology Dr
Minneapolis MN 55413-1096

1

AFCSC/SPER
ATTN: 1Lt Glenn Armstrong
San Antonio TX 78243-5000

1

**MISSION
OF
ROME LABORATORY**

Rome Laboratory plans and executes an interdisciplinary program in research, development, test, and technology transition in support of Air Force Command, Control, Communications and Intelligence (C³I) activities for all Air Force platforms. It also executes selected acquisition programs in several areas of expertise. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of C³I systems. In addition, Rome Laboratory's technology supports other AFSC Product Divisions, the Air Force user community, and other DOD and non-DOD agencies. Rome Laboratory maintains technical competence and research programs in areas including, but not limited to, communications, command and control, battle management, intelligence information processing, computational sciences and software producibility, wide area surveillance/sensors, signal processing, solid state sciences, photonics, electromagnetic technology, superconductivity, and electronic reliability/maintainability and testability.