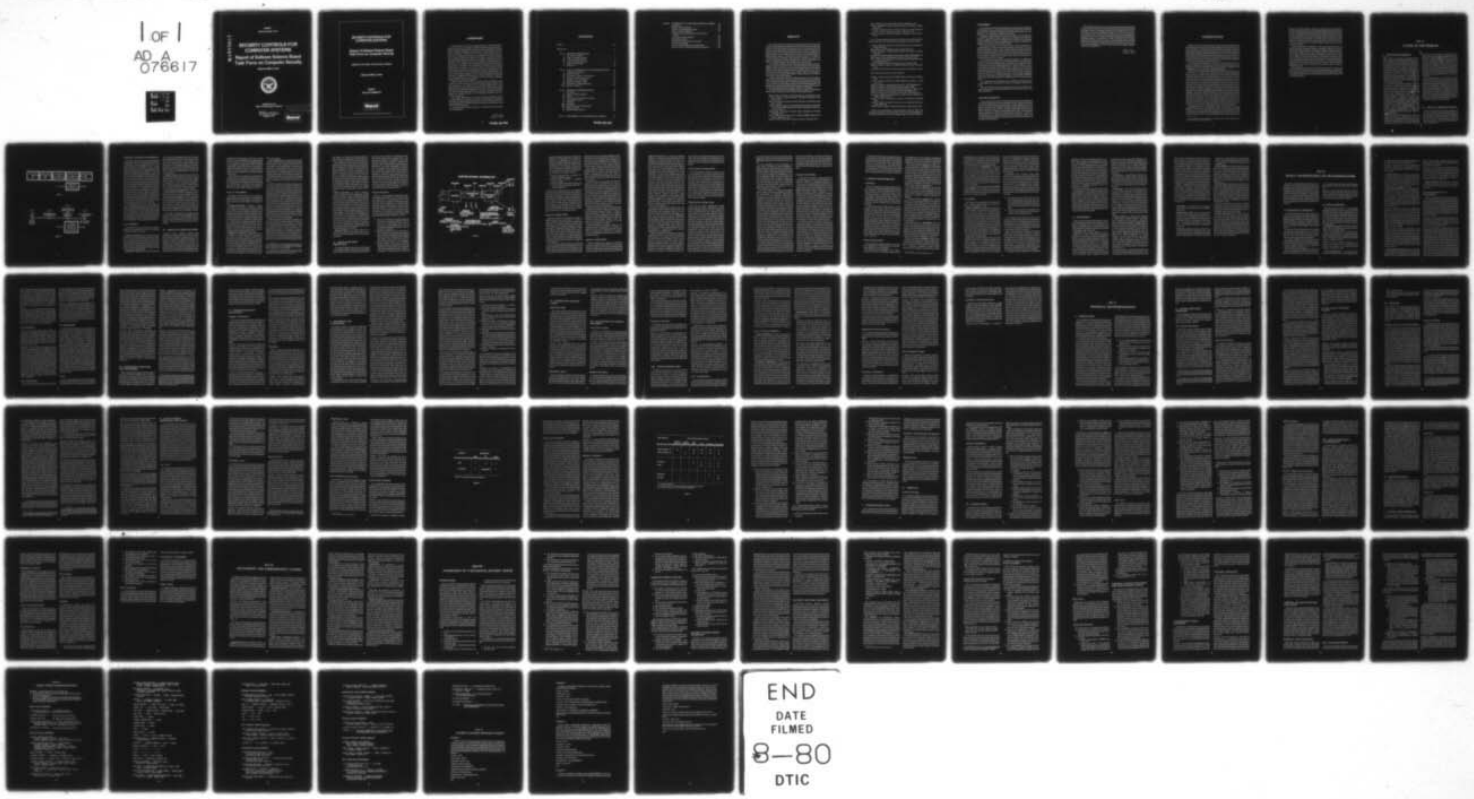
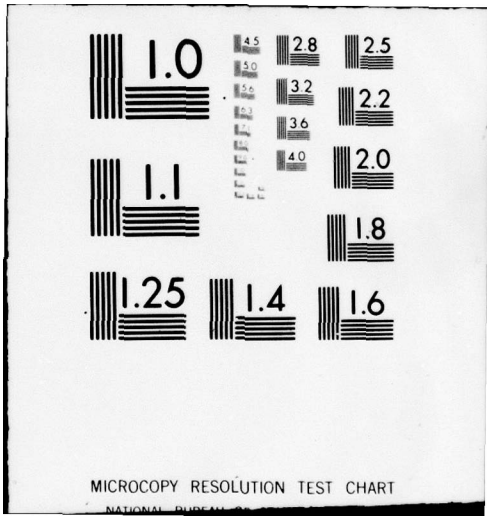
AD-A076 617 RAND CORP SANTA MONICA CA F/G 9/2
SECURITY CONTROLS FOR COMPUTER SYSTEMS. REPORT OF DEFENSE SCIEN--ETC(U)
OCT 79 W H WARE
UNCLASSIFIED RAND/R-609-1 NL

1 OF 1
AD A 076617



END
DATE FILMED
8-80
DTIC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART

AD A076617

R-609-1

Reissued October 1979

SECURITY CONTROLS FOR COMPUTER SYSTEMS

Report of Defense Science Board Task Force on Computer Security

Edited by Willis H. Ware



Published for the
Office of the Secretary of Defense

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

REPRODUCED BY
NATIONAL TECHNICAL
INFORMATION SERVICE
U.S. DEPARTMENT OF COMMERCE
SPRINGFIELD, VA. 22161

Rand
SANTA MONICA, CA. 90406

SECURITY CONTROLS FOR COMPUTER SYSTEMS

Report of Defense Science Board Task Force on Computer Security

Published for the Office of the Secretary of Defense

Edited by Willis H. Ware

R-609-1

Reissued October 1979

Rand
SANTA MONICA, CA. 90406

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

FOREWORD

In October 1967 a Task Force was organized by the Advanced Research Projects Agency (now the Defense Advanced Research Projects Agency) to study and recommend appropriate computer security safeguards that would protect classified information in multi-access, resource-sharing computer systems. The report of the Task Force, which functioned under the auspices of the Defense Science Board, was published by The Rand Corporation in February 1970 for the Office of the Director of Defense Research and Engineering, Department of Defense. A slightly modified version of the report—the only omissions were two memoranda of transmittal from the Task Force to the Chairman of the Defense Science Board and onward to the Secretary of Defense—was subsequently published as Rand Report R-609, *Security Controls for Computer Systems.* At that time it was felt that because representatives from government agencies participated in the work of the Task Force, the information in the report would appear to be of an official nature, suggestive of the policies and guidelines that would eventually be established. Consequently, it was felt prudent to classify the report Confidential overall. On October 10, 1975, the Defense Advanced Research Projects Agency declassified it.

Nearly a decade later the report is still a valuable comprehensive discussion of security controls for resource-sharing computer systems. Ideas first expressed in this report and even occasional figures from it have gradually seeped into the technical literature, but it still contains material that has not been published elsewhere. For example, it includes an appendix that outlines and formally specifies a set of access controls that can accommodate the intricate structure of the classification system used by the defense establishment.

The original classification of the report limited its distribution largely to defense agencies and defense contractors; civil agencies of government and industry at large generally did not have access to it. Because of the continuing importance of computer security, the report is being reissued at this time for wider distribution.

The support of The Rand Corporation in reprinting this report is gratefully acknowledged.

Willis H. Ware
October 10, 1979

Preceding page blank

CONTENTS

Preface vii

Introduction xi

Part A. NATURE OF THE PROBLEM 1
I. The Security Problem 1
II. Types of Computer Systems 1
III. Threats to System Security 3
IV. Areas of Security Protection 5
V. System Characteristics 10
VI. Definitions 12

Part B. POLICY CONSIDERATIONS AND RECOMMENDATIONS 14
I. Fundamental Principles 14
II. System Personnel 14
III. Information Structure and Transforms 17
IV. System Transaction Accounting 18
V. Reliability and Auto-Testing 19
VI. Information Security Labels 21
VII. Management of Storage Resources 21
VIII. System Certification 22

Part C. TECHNICAL RECOMMENDATIONS 26
I. Introduction 26
II. Central Processor Hardware 27
III. Software 29
IV. Access Control Throughout the System 31
V. Communication Lines 38
VI. Terminals 38
VII. Certification 39
VIII. Open Environment Considerations 42
IX. Research Needed 43
X. Overall System Problems 43

Part D. MANAGEMENT AND ADMINISTRATIVE CONTROL 46

Preceding page blank

Appendix: AUTOMATION OF A MULTILEVEL SECURITY SYSTEM 48
Introduction 48
Computer System Catalogs 50
Security Control System Generation 50
Security Structure Definition 51
Personnel Security Definition and User Clearance Update 54
Authorization Group Definition 55
Universal Privileges 55
Terminal Security Definition and Update 56
File Access Processing 56
Annex A: Formal System Access Specification 58
Annex B: Security Component Definition Examples 62

PREFACE

The question of security control in resource-sharing systems was brought into focus for the Department of Defense by a series of events in the spring and summer of 1967. Such systems were being procured in increasing numbers for government installations; the problems of security for them were becoming of pressing concern both to defense contractors and to military operations; the Research Security Administrators had forwarded a position paper through the Defense Supply Agency to the Director for Security Policy in the Office of Assistant Secretary of Defense (Administration) soliciting action. Since the matter involved technical issues, the paper was referred to the Office of the Director of Defense Research and Engineering for consideration.

In June 1967, the Deputy Director (Administration, Evaluation and Management) requested the Director of the Advanced Research Projects Agency (ARPA) to form a Task Force to study and recommend hardware and software safeguards that would satisfactorily protect classified information in multi-access, resource-sharing computer systems. Within ARPA, the responsibility for this task was forwarded to Mr. Robert W. Taylor, Director of the Office of Information Processing Techniques.

A series of discussions was held during the summer and fall months of 1967 with people from the university and industrial communities, culminating in the formation by October 1967 of a Task Force consisting of a Steering Group and two Panels. The organizational meeting was held the following month, and thereafter the Panels and the Steering Group met on a regular basis to formulate the recommendations that constitute the body of this Report.

The Task Force has operated formally under the authority of the Defense Science Board. Following are the members of the Steering Group:

Willis H. Ware, Chairman, *The Rand Corporation, Santa Monica, Calif.*

J. Patrick Haverty, Deputy Chairman, *The Rand Corporation, Santa Monica, Calif.*

Robert A. Mosier, Vice Chairman, *System Development Corporation, Santa Monica, Calif.*

Arthur A. Bushkin, Secretary, *Lockheed Missiles and Space Co., Palo Alto, Calif. (formerly, Massachusetts Institute of Technology and Bolt, Beranek and Newman)*

Elliot Cassidy, *Directorate for Security Policy, Department of Defense, Washington, D.C.*

John F. Egan, *Office of the Secretary of Defense/DDR&E, Department of Defense, Washington, D.C.*

Edward L. Glaser, *Case Western Reserve University, Cleveland, Ohio*

John W. Kuipers, *Central Intelligence Agency, Washington, D.C.*
Jerome D. Moskowitz, *National Security Agency, Fort George G. Meade, Maryland*
Lawrence G. Roberts (formerly, Robert W. Taylor), *Advanced Research Projects Agency, Department of Defense, Washington, D.C.*
Robert von Buelow, *System Development Corporation, Santa Monica, Calif.*

The two panels organized under the Steering Group are the Policy Panel and the Technical Panel. The following are members of the Policy Panel:

Jerome D. Moskowitz, Chairman, *National Security Agency, Fort George G. Meade, Maryland*
Donal Burns, *Central Intelligence Agency, Washington, D.C.*
Thomas Chittenden, *National Security Agency, Fort George G. Meade, Maryland*
Richard G. Cleaveland, *Defense Communication Agency, Washington, D.C.*
Roy McCabe, *System Development Corporation, Sacramento, Calif.*
Barry Wessler, *Advanced Research Projects Agency, Department of Defense, Washington, D.C.*
Ronald Wigington, *Chemical Abstracts Service, Columbus, Ohio*
Edward L. Glaser *(ex officio), Case Western Reserve University, Cleveland, Ohio*
Willis H. Ware *(ex officio), The Rand Corporation, Santa Monica, Calif.*

The Technical Panel consists of the following:

Edward L. Glaser, Chairman, *Case Western Reserve University, Cleveland, Ohio*
Arthur A. Bushkin, Secretary, *Lockheed Missiles and Space, Co., Palo Alto, Calif.*
James P. Anderson, *James P. Anderson and Co., Fort Washington, Pa.*
Edward H. Bensley, *The MITRE Corporation, Bedford, Mass.*
Charles R. Blair, *International Business Machines Corp., Yorktown, N.Y.*
Daniel L. Edwards, *National Security Agency, Washington, D.C.*
Harold M. Jayne, *Executive Office of The President, Washington, D.C.*
Lawrence G. Roberts, *Advanced Research Projects Agency, Department of Defense, Washington, D.C.*
Jerome H. Saltzer, *Massachusetts Institute of Technology, Cambridge, Mass.*
Jerome D. Moskowitz *(ex officio), National Security Agency, Fort George G. Meade, Maryland*
Willis H. Ware *(ex officio), The Rand Corporation, Santa Monica, Calif.*

Initially, the representative of the Directorate for Security Policy was Lieutenant Commander Armen Chertavian (USN); and the representative to the Policy Panel from the Central Intelligence Agency, was Mr. Fred Ohm.

AUTHORSHIP

The members of the Task Force participated as individuals knowledgeable of the technical, policy, and administrative issues involved. Thus, the views stated herein do not reflect the policy of the Federal Government, any of its agencies, or any university or industrial corporation.

Ultimately, a Report has to be written by one person. The original draft was written by Willis H. Ware using sources as noted below. It was then critiqued, modified, emended, and shaped by the members of the Steering Group and the Panels. A second complete draft was written by Thomas Chittenden, and the final version by Willis H. Ware.

Each Panel produced a series of papers which formed the basis for the recommendations on software, hardware, procedures, and policy. The Introduction and portions of Part A were initially authored by Wade B. Holland, utilizing material provided by Willis H. Ware and other sources. Section V of Part A, on System Characteristics, is largely from Willis H. Ware, incorporating material from a paper by the Technical Panel and some information from personal letters of Prof. E. L. Glaser.

Part B, the Policy Considerations and Recommendations, is substantially from the final paper produced by the Policy Panel. Many of the explanatory comments come from the original paper, although some were added in the final writing. The Technical Recommendations, Part C, mainly reflect the content of two papers produced by the Technical Panel, modified to a minor extent by information from personal letters of Prof. Glaser. Finally, Part D, on Management and Administrative Control, was written by Willis H. Ware, and utilizes ideas from "Security of Classified Information in the Defense Intelligence Agency's Analyst Support and Research System" (February 1969, C-3663/MS-5), and from "Security Procedures for the RYE System" (W. B. Ellis, December 1968).

The Appendix was first drafted by Arthur A. Bushkin and Willis H. Ware; it was subsequently extended and rewritten by Mr. Bushkin and Robert M. Balzer.

The final editing and details of format and style are due to Wade B. Holland. The Report was printed and published by The Rand Corporation, under ARPA sponsorship.

ACKNOWLEDGMENTS

The success of a venture such as this depends upon the personal dedication and volunteer participation of the individuals involved. In addition to the listed members of the Steering Group and the Panels, it is also a pleasure to acknowledge the contributions of Dr. Robert M. Balzer and Mr. Wade B. Holland, The Rand Corporation, Santa Monica, California; Miss Hilda Faust, National Security Agency, Fort George G. Meade, Maryland; and Mr. Clark Weissman, System Development Corporation, Santa Monica, California. A special acknowledgment is due Thomas Chittenden, National Security Agency, Fort George G. Meade, Maryland, who rewrote the entire document to produce the all-important second draft.

The subject of security control in multi-access computer systems is of sufficiently wide interest that many members of the Steering Group and the Panels contacted a number of individuals, organizations, and agencies in the course of this effort. It would be impossible to mention every person with whom we have talked and who in some way has influenced our final recommendations. Among others, however, we interacted with Colonel Roy Morgan of the Defense Intelligence Agency representing the ANSR computing system, and Mr. George Hicken, National Security Agency, representing the RYE and COINS systems. The Steering Group and its Panels also acknowledge the contributions of the many individuals who read our draft material and supplied valuable comments and suggestions.

Willis H. Ware
January 1, 1970

INTRODUCTION

With the advent of *resource-sharing* computer systems that distribute the capabilities and components of the machine configuration among several users or several tasks, a new dimension has been added to the problem of safeguarding computer-resident classified information. The basic problems associated with machine processing of classified information are not new. They have been encountered in the batch-processing mode of operation and, more recently, in the use of remote job-entry systems; the methods used to safeguard information in these systems have, for the most part, been extensions of the traditional manual means of handling classified documents.

The increasingly widespread use of resource-sharing systems has introduced new complexities to the problem. Moreover, the use of such systems has focused attention on the broader issue of using computers, regardless of the configuration, to store and process classified information.

Resource-sharing systems are those that distribute the resources of a computer system (e.g., memory space, arithmetic units, peripheral equipment, channels) among a number of simultaneous users. The term includes systems commonly called *time-sharing, multiprogrammed, remote batch, on-line, multi-access,* and, where two or more processors share all of the primary memory, *multiprocessing.* The principle distinction among the systems is whether a user must be present (at a terminal, for example) to interact with his job (time-sharing, on-line, multi-access), or whether the jobs execute autonomously (multiprogrammed, remote batch). Resource-sharing allows many people to use the same complex of computer equipment concurrently. The users are generally, although not necessarily, geographically separated from the central processing equipment and interact with the machine via remote terminals or consoles. Each user's program is executed in some order and for some period of time, not necessarily to completion. The central processing equipment devotes its resources to servicing users in turn, resuming with each where it left off in the previous processing cycle. Due to the speeds of modern computers, the individual user is rarely aware that he is receiving only a fraction of the system's attention or that his job is being fragmented into pieces for processing.

Multiprogramming is a technique by which resource-sharing is accomplished. Several jobs are simultaneously resident in the system, each being handled by the various system components so as to maximize efficient utilization of the entire configuration. The operating system[1] switches control from one job to another in such a way that advantage is taken of the machine's most

[1]The system software, which schedules work through the computer system, assigns resources to each job, accounts for resources used, etc.

powerful—and most expensive—resources. In practice, one of the basic features of multiprogramming is to prevent jobs demanding large amounts of time in input or output functions (I/O-bound jobs) from tying up the central processor; this is accomplished usually by allowing each job to execute until an input or output operation is required, at which point another job begins to execute concurrently with the I/O request. On the other hand, a time-sharing system regularly interrupts each job in turn, allowing each to execute for some interval of time determined by the computer system itself rather than by the structure of the job.

Systems incorporating capabilities of the types enumerated represent some of the latest advances in computer technology. Basically, they are intended to provide the most efficient utilization of expensive computing facilities for the widest range of users. A single system is able to handle several users or several sets of data simultaneously, contributing to more economical operation. In addition to the direct advantages of vastly improved resource utilization and greatly increased economy of operation, they can drastically reduce service turn-around time, enable users with little or no formal knowledge of programming to interact directly with the machine, and extend computing capabilities to many smaller installations that would be unable to support a dedicated machine.

This study, while receiving its impetus from the concern that has been generated by the increasing number of time-sharing systems, is addressed to all computer systems that may process classified material. Methods developed to insure the security of resource-sharing systems are applicable to other kinds of computing systems.

Part A

NATURE OF THE PROBLEM

I. THE SECURITY PROBLEM

The wide use of computers in military and defense installations has long necessitated the application of security rules and regulations. A basic principle underlying the security of computer systems has traditionally been that of isolation—simply removing the entire system to a physical environment in which penetrability is acceptably minimized. The increasing use of systems in which some equipment components, such as user access terminals, are widely spread geographically has introduced new complexities and issues. These problems are not amenable to solution through the elementary safeguard of physical isolation.

In one sense, the expanded problems of security provoked by resource-sharing systems might be viewed as the price one pays for the advantages these systems have to offer. However, viewing the question from the aspect of such a simplistic tradeoff obscures more fundamental issues. First, the security problem is not unique to any one type of computer system or configuration; it applies across the spectrum of computational technology. While the present paper frames the discussions in terms of time-sharing or multiprogramming, we are really dealing not with system configurations, but with security; today's computational technology has served as catalyst for focusing attention on the problem of protecting classified information resident in computer systems.

Secondly, resource-sharing systems, where the problems of security are admittedly most acute at present, must be designed to protect each user from interference by another user or by the system itself, and must provide some sort of "privacy" protection to users who wish to preserve the integrity of their data and their programs. Thus, designers and manufacturers of resource-sharing systems are concerned with the fundamental problem of protecting information. In protecting classified information, there are differences of degree, and there are new surface problems, but the basic issues are generally equivalent. The solutions the manufacturer designs into the hardware and software must be augmented and refined to provide the additional level of protection demanded of machines functioning in a security environment.

The recommendations of the Defense Science Board's Task Force on Computer Security represent a compilation of techniques and procedures which should be considered both separately and in combination when designing or adopting data processing systems to provide security or user privacy. The solutions to specific problems are intended to be flexible and adaptive to the needs of any installation, rather than being oriented to any one applications environment. It is intended that the general guidelines in this Report be of use to DOD components, other government installations, and contractors.

II. TYPES OF COMPUTER SYSTEMS

There are several ways in which a computer system can be physically and operationally organized to serve its users. The security controls will depend on the configuration and the sensitivity of data processed in the system. The following discussion presents two ways of viewing the physical and operational configurations.

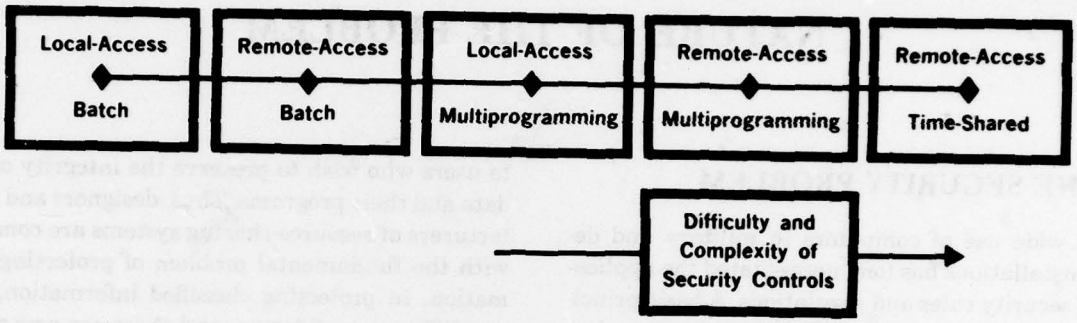


Figure 1

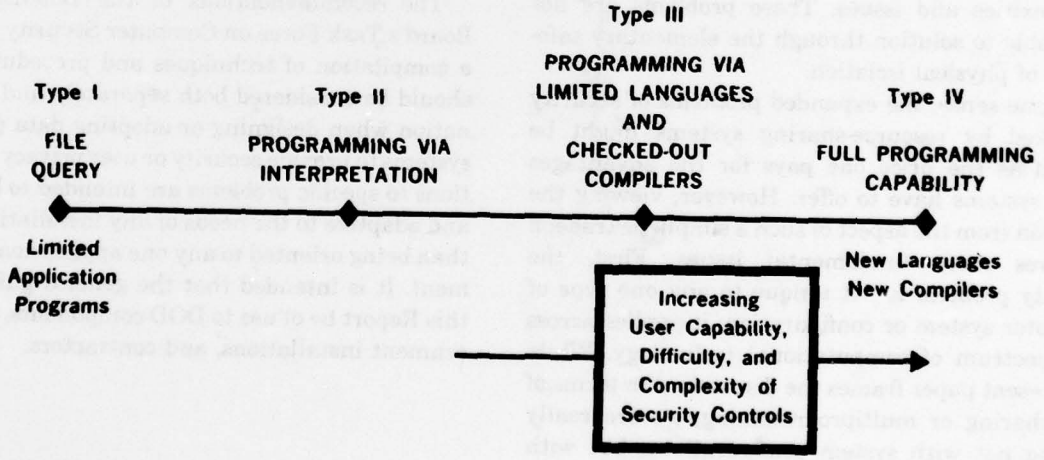


Figure 2

Equipment Arrangement and Disposition

The organization of the central processing facilities for batch or for time-shared processing, and the arrangement of access capabilities for local or for remote interaction are depicted in Fig. 1. Simple batch processing is the historical and still prevalent mode of operation, wherein a number of jobs or transactions are grouped and processed as a unit. The batches are usually manually organized, and for the most part each individual job is processed to completion in the order in which it was received by the machine. An important characteristic of such single-queue, batched, run-to-completion systems which do not have an integrated file management system for non-demountable, on-line memory media is that the system need have no "management awareness" from job to job. Sensitive materials can be erased or removed from the computer quickly and relatively cheaply, and mass memory media containing sensitive information can be physically separated from the system and secured for protection. This characteristic explains why solution to the problem we are treating has not been as urgent in the past.

In multiprogramming, on the other hand, the jobs are organized and processed by the system according to algorithms designed to maximize the efficiency of the total system in handling the complete set of transactions. In *local-access* systems, all elements are physically located within the computer central facility; in *remote-access* systems, some units are geographically distant from the central processor and connected to it by communication lines.

User Capabilities

Another way of viewing the types of systems, shown in Fig. 2, is based on the levels of computing capability available to the user.

File-query systems (Type I) enable the user to execute only limited application programs embedded in the system and not available to him for alteration or change. He selects for execution one or more available application programs. He may be able to couple several of these programs together for automatic execution in sequence and to insert parameters into the selected programs.

Interpretive systems (Type II) provide the user with a programming capability, but only in terms of input language symbols that result in direct execution within the computer of the operations they denote. Such symbols are not used to construct an internal machine language program that can subsequently be executed upon command from the user. Thus, the user cannot obtain control of the machine directly, because he is buffered from it by the interpretive software.

Compiler systems (Type III) provide the user with a programming capability, but only in terms of languages that execute through a compiler embedded in the system. The instructions to the compiler are translated by it into an assembly language or basic machine language program. Program execution is controlled by the user; however, he has available to him only the limited compiler language.

Full programming systems (Type IV) give the user extensive and unrestrained programming capability. Not only can he execute programs written in standard compiler languages, but he also can create new programming languages, write compilers for them, and embed them within the system. This gives the user intimate interaction with and control over the machine's complete resources—excepting of course, any resources prohibited to him by information-protecting safeguards (e.g., memory protection, base register controls, and I/O hardware controls).

In principle, all combinations of equipment configurations (Fig. 1) and operational capabilities (Fig. 2) can exist. In practice, not all the possible combinations have been implemented, and not all the possibilities would provide useful operational characteristics.

III. THREATS TO SYSTEM SECURITY

By their nature, computer systems bring together a series of vulnerabilities. There are human vulnerabilities throughout; individual acts can accidentally or deliberately jeopardize the system's information protection capabilities. Hardware vulnerabilities are shared among the computer, the communication facilities, and the remote units and consoles. There are software vulnerabilities at all levels of the machine operating system and supporting software; and there are vulnerabilities in the

organization of the protection system (e.g., in access control, in user identification and authentication, etc.). How serious any one of these might be depends on the sensitivity (classification) of the information being handled, the class of users, the computational capabilities available to the user, the operating environment, the skill with which the system has been designed, and the capabilities of potential attackers of the system.

These points of vulnerability are applicable both in industrial environments handling proprietary information and in government installations processing classified data. This Report is concerned directly with only the latter; it is sufficient here to acknowledge that the entire range of issues considered also has a "civil" side to which this work is relevant.

Types of Vulnerabilities

The design of a secure system must provide protection against the various types of vulnerabilities. These fall into three major categories: accidental disclosures, deliberate penetrations, and physical attack.

Accidental Disclosure. A failure of components, equipment, software, or subsystems, resulting in an exposure of information or violation of any element of the system. Accidental disclosures are frequently the result of failures of hardware or software. Such failures can involve the coupling of information from one user (or computer program) with that of another user, the "clobbering" of information (i.e., rendering files or programs unusable), the defeat or circumvention of security measures, or unintended change in security status of users, files, or terminals. Accidental disclosures may also occur by improper actions of machine operating or maintenance personnel without deliberate intent.

Deliberate Penetration. A deliberate and covert attempt to (1) obtain information contained in the system, (2) cause the system to operate to the advantage of the threatening party, or (3) manipulate the system so as to render it unreliable or unusable to the legitimate operator. Deliberate efforts to penetrate secure systems can either be active or passive. *Passive* methods include wire tapping and monitoring of electromagnetic emanations. *Active* infiltration is an attempt to enter the system so as to obtain data from the files or to interfere with data files or the system.[1]

Active Infiltration. One method of accomplishing active infiltration is for a legitimate user to penetrate portions of the system for which he has no authorization. The design problem is one of preventing access to files by someone who is aware of the access control mechanisms and who has the knowledge and desire to manipulate them to his own advantage. For example, if the access control codes are all four-digit numbers, a user can pick any four-digit number, and then, having gained access to some file, begin interacting with it in order to learn its contents.

Another class of active infiltration techniques involves the exploitation of trap-door[2] entry points in the system that by-pass the control facilities and permit direct access to files. Trap-door entry points often are created deliberately during the design and development stage to simplify the insertion of authorized program changes by legitimate system programmers, with the intent of closing the trap-door prior to operational use. Unauthorized entry points can be created by a system programmer who wishes to provide a means for bypassing internal security controls and thus subverting the system. There is also the risk of implicit trap-doors that may exist because of incomplete system design—i.e., loopholes in the protection mechanisms. For example, it might be possible to find an unusual combination of system control variables that will create an entry path around some or all of the safeguards.

Another potential mode of active infiltration is the use of a special terminal illegally tied into the communication system. Such a terminal can be used to intercept information flowing between a legitimate terminal and the central processor, or to manipulate the system. For example, a legitimate user's sign-off signal can be intercepted and cancelled; then, the illegal terminal can take over interaction with the processor. Or, an illegal terminal can maintain activity during periods when the legitimate user is inactive but still maintaining an open

---

[1]The discussion of subversion is largely based on the article by H. E. Petersen and R. Turn, "System Implications of Information Privacy," *AFIPS Conference Proceedings*, Vol. 30, Thompson Books, Washington, D.C., 1967, pp. 291-300.

[2]Any opportunity to penetrate, subvert, mislead, or by-pass security controls through an idiosyncracy of the software, software-hardware, hardware, procedural controls, etc.

line. Finally, the illegal terminal might drain off output directed to a legitimate terminal and pass on an error message in its place so as to delay detection.

Active infiltration also can be by an agent operating within the secure organization. This technique may be restricted to taking advantage of system protection inadequacies in order to commit acts that appear accidental but which are disruptive to the system or to its users, or which could result in acquisition of classified information. At the other extreme, the agent may actively seek to obtain removable files or to create trap doors that can be exploited at a later date. Finally, an agent might be placed in the organization simply to learn about the system and the operation of the installation, and to obtain what pieces of information come his way without any particularly covert attempts on his part at subversion.

Passive Subversion. In passive subversion, means are applied to monitor information resident within the system or being transmitted through the communication lines without any corollary attempt to interfere with or manipulate the system. The most obvious method of passive infiltration is the wire tap. If communications between remote terminals and the central processor are over unprotected circuits, the problem of applying a wire tap to the computer line is similar to that of bugging a telephone call. It is also possible to monitor the electromagnetic emanations that are radiated by the high-speed electronic circuits that characterize so much of the equipment used in computational systems. Energy given off in this form can be remotely recorded without having to gain physical access to the system or to any of its components or communication lines. The possibility of successful exploitation of this technique must always be considered.

Physical Attack. Overt assault against or attack upon the physical environment (e.g., mob action) is a type of vulnerability outside the scope of this Report.

IV. AREAS OF SECURITY PROTECTION

The system designer must be aware of the points of vulnerability, which may be thought of as leakage points, and he must provide adequate mechanisms to counteract both accidental and deliberate events. The specific leakage points touched upon in the foregoing discussion can be classified in five groups: physical surroundings, hardware, software, communication links, and organizational (personnel and procedures). The overall safeguarding of information in a computer system, regardless of configuration, is achieved by a combination of protection features aimed at the different areas of leakage points. Procedures, regulations, and doctrine for some of these areas are already established within DOD, and are not therefore within the purview of the Task Force. However, there is some overlap between the various areas, and when the application of security controls to computer systems raises a new aspect of an old problem, the issue is discussed. An overview of the threat points is depicted in Fig. 3.

Physical Protection

Security controls applied to safeguard the physical equipment apply not only to the computer equipment itself and to its terminals, but also to such removable items as printouts, magnetic tapes, magnetic disc packs, punchcards, etc. Adequate DOD regulations exist for dissemination, control, storage, and accountability of classified removable items. Therefore, security measures for these elements of the system are not examined in this Report unless there are some unique considerations. The following general guidelines apply to physical protection.

(a) The area containing the central computing complex and associated equipment (the machine room or operational area) must be secured to the level commensurate with the most highly classified and sensitive material handled by the system.
(b) Physical protection must be continuous in time, because of the threat posed by the possibility of physical tampering with equipment and because of the likelihood that classified information will be stored within the computer system even when it is not operating.
(c) Remote terminal devices must be afforded physical protection commensurate with the classification and sensitivity of information that can be handled through them. While responsibility for instituting and maintaining

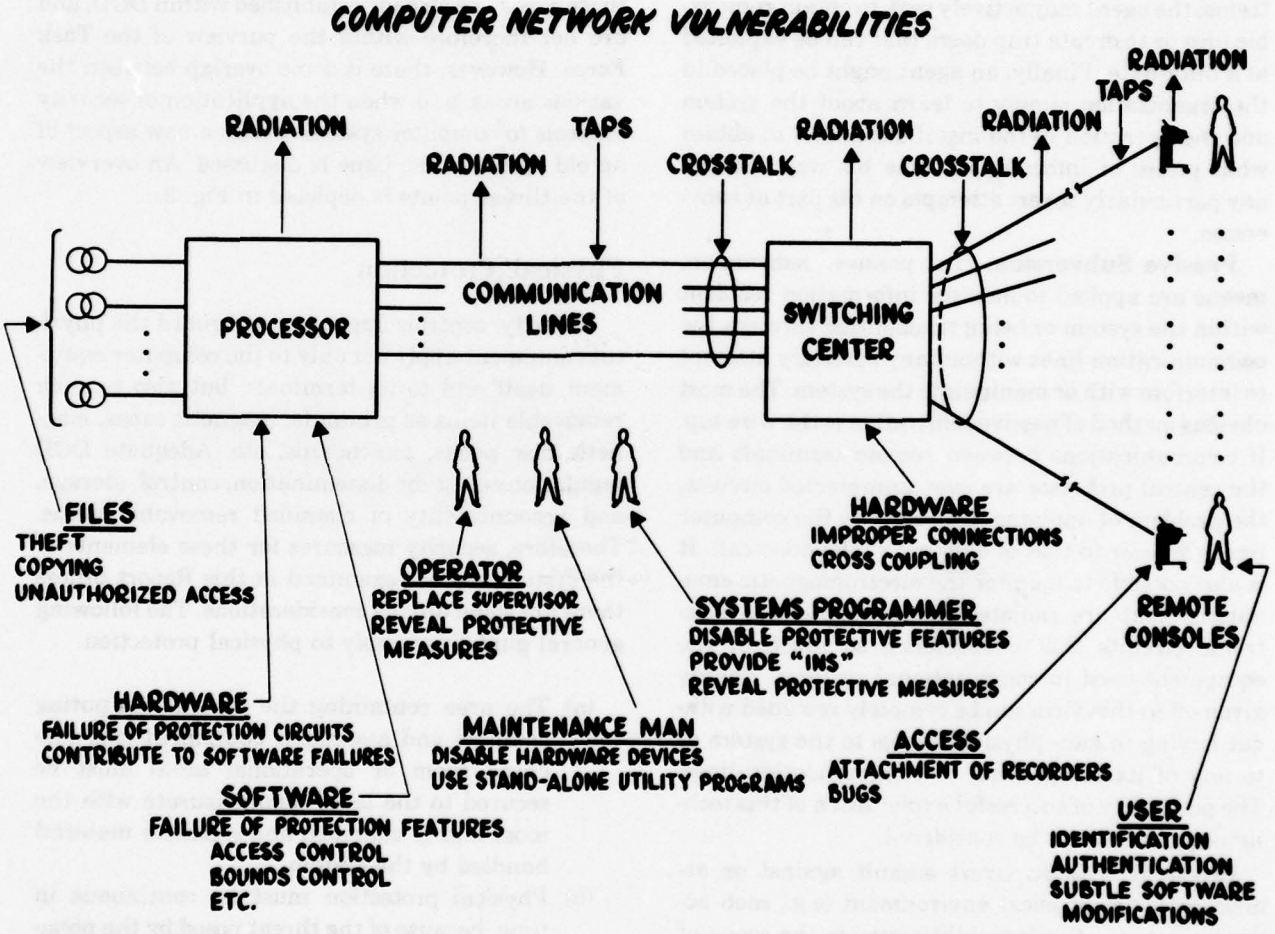

COMPUTER NETWORK VULNERABILITIES
RADIATION
TAPS
RADIATION
RADIATION
TAPS
RADIATION
CROSSTALK
RADIATION
CROSSTALK
PROCESSOR
COMMUNICATION LINES
SWITCHING CENTER
FILES
THEFT
COPYING
UNAUTHORIZED ACCESS
OPERATOR
REPLACE SUPERVISOR
REVEAL PROTECTIVE MEASURES
HARDWARE
IMPROPER CONNECTIONS
CROSS COUPLING
SYSTEMS PROGRAMMER
DISABLE PROTECTIVE FEATURES
PROVIDE "INS"
REVEAL PROTECTIVE MEASURES
REMOTE CONSOLES
HARDWARE
FAILURE OF PROTECTION CIRCUITS
CONTRIBUTE TO SOFTWARE FAILURES
MAINTENANCE MAN
DISABLE HARDWARE DEVICES
USE STAND-ALONE UTILITY PROGRAMS
ACCESS
ATTACHMENT OF RECORDERS
BUGS
SOFTWARE
FAILURE OF PROTECTION FEATURES
ACCESS CONTROL
BOUNDS CONTROL
ETC.
USER
IDENTIFICATION
AUTHENTICATION
SUBTLE SOFTWARE MODIFICATIONS

Figure 3

physical protection measures is normally assigned to the organization that controls the terminal, it is advisable for a central authority to establish uniform physical security standards (specific protection measures and regulations) for all terminals in a given system to insure that a specified security level can be achieved for an entire system. Terminal protection is important in order to:

- Prevent tampering with a terminal (installing intelligence sensors);
- Prevent visual inspection of classified work in progress;
- Prevent unauthorized persons from trying to call and execute classified programs or obtain classified data.

If parts of the computer system (e.g., magnetic disc files, copies of printouts) contain unusually sensitive data, or must be physically isolated during maintenance procedures, it may be necessary to physically separate them and independently control access to them. In such cases, it may be practical to provide direct or remote visual surveillance of the ultra-sensitive areas. If visual surveillance is used, it must be designed and installed in such a manner that it cannot be used as a trap-door to the highly sensitive material it is intended to protect.

Hardware Leakage Points

Hardware portions of the system are subject to malfunctions that can result directly in a leak or cause a failure of security protection mechanisms elsewhere in the system, including inducing a software malfunction. In addition, properly operating equipment is susceptible to being tapped or otherwise exploited. The types of failures that most directly affect security include malfunctioning of the circuits for such protections as bounds registers, memory read-write protect, privileged mode operation, or priority interrupt. Any hardware failure potentially can affect security controls; e.g., a single-bit error in memory.

Both active and passive penetration techniques can be used against hardware leakage points. In the passive mode, the intervener may attempt to monitor the system by tapping into communication lines, or by monitoring compromising emanations. Wholly isolated systems can be physically shielded to eliminate emanations beyond the limits of the secure installation, but with geographically dispersed systems comprehensive shielding is more difficult and expensive. Currently, the only practical solutions are those used to protect communications systems.

The problem of emanation security is covered by existing regulations; there are no new aspects to this problem raised by modern computing systems. It should be emphasized, however, that control of spurious emanations must be applied not only to the main computing center, but to the remote equipment as well.

Although difficult to accomplish, the possibility exists that covert monitoring devices can be installed within the central processor. The problem is that the computer hardware involved is of such complexity that it is easy for a knowledgeable person to incorporate the necessary equipment in such a way as to make detection very difficult. His capability to do so assumes access to the equipment during manufacture or major maintenance. Equipment is also vulnerable to deliberate or accidental rewiring by maintenance personnel so that installed hardware appears to function normally, but in fact by-passes or changes the protection mechanisms.

Remote consoles also present potential radiation vulnerabilities. Moreover, there is a possibility that recording devices might be attached to a console to pirate information. Other remote or peripheral equipment can present dangers. Printer ribbons or platens may bear impressions that can be analyzed; removable storage media (magnetic tapes, disc packs, even punchcards) can be stolen, or at least removed long enough to be copied.

Erasure standards for magnetic media are not within the scope of this Task Force to review or establish. However, system designers should be aware that the phenomena of retentivity in magnetic materials is inadequately understood, and is a threat to system security.

Software Leakage Points

Software leakage points include all vulnerabilities directly related to the software in the computer system. Of special concern is the operating system and the supplementary programs that support the operating system because they contain the software

safeguards. Weaknesses can result from improper design, or from failure to check adequately for combinations of circumstances that can lead to unpredictable consequences. More serious, however, is the fact that operating systems are very large, complex structures, and thus it is impossible to exhaustively test for every conceivable set of conditions that might arise. Unanticipated behavior can be triggered by a particular user program or by a rare combination of user actions. Malfunctions might only disrupt a particular user's files or programs; as such, there might be no risk to security, but there is a serious implication for system reliability and utility. On the other hand, operating system malfunctions might couple information from one program (or user) to another; clobber information in the system (including information within the operating system software itself); or change classification of users, files, or programs. Thus, malfunctions in the system software represent potentially serious security risks. Conceivably, a clever attacker might establish a capability to induce software malfunctions deliberately; hiding beneath the apparently genuine trouble, an on-site agent may be able to tap files or to interfere with system operation over long periods without detection.

The security safeguards provided by the operating system software include access controls, user identification, memory bounds control, etc. As a result of a hardware malfunction, especially a transient one, such controls can become inoperative. Thus, internal checks are necessary to insure that the protection is operative. Even when this is done, the simultaneous failure of both the protection feature and its check mechanism must always be regarded as a possibility. With proper design and awareness of the risk, it appears possible to reduce the probability of undetected failure of software safeguards to an acceptable level.

Probably the most serious risk in system software is incomplete design, in the sense that inadvertent loopholes exist in the protective barriers and have not been foreseen by the designers. Thus, unusual actions on the part of users, or unusual ways in which their programs behave, can induce a loophole. There may result a security breach, a suspension or modification of software safeguards (perhaps undetected), or wholesale clobbering of internal programs, data, and files. It is conceivable that an attacker could mount a deliberate search for such loopholes with the expectation of exploiting them to acquire information either from the system or about the system—e.g., the details of its information safeguards.

Communication Leakage Points

The communications linking the central processor, the switching center and the remote terminals present a potential vulnerability. Wiretapping may be employed to steal information from land lines, and radio intercept equipment can do the same to microwave links. Techniques for intercepting compromising emanations may be employed against the communications equipment even more readily than against the central processor or terminal equipment. For example, crosstalk between communications lines or within the switching central itself can present a vulnerability. Lastly, the switch gear itself is subject to error and can link the central processor to the wrong user terminal.

Organizational Leakage Points

There are two prime organizational leakage points, personnel security clearances and institutional operating procedures. The first concerns the structure, administration, and mechanism of the national apparatus for granting personnel security clearances. It is accepted that adequate standards and techniques exist and are used by the cognizant authority to insure the reliability of those cleared. This does not, however, relieve the system designer of a severe obligation to incorporate techniques that minimize the damage that can be done by a subversive individual working from within the secure organization. *A secure system must be based on the concept of isolating any given individual from all elements of the system to which he has no need for access.* In the past, this was accomplished by denying physical access to anyone without a security clearance of the appropriate level. In resource-sharing systems of the future, a population of users ranging from uncleared to those with the highest clearance levels will interact with the system simultaneously. This places a heavy burden on the overall security control apparatus to insure that the control mechanisms incorporated into the computer system are

properly informed of the clearances and restrictions applicable to each user. The machine system must be designed to apply these user access restrictions reliably.

In some installations, it may be feasible to reserve certain terminals for highly classified or highly sensitive or restricted work, while other terminals are used exclusively for less sensitive operation. Conversely, in some installations any terminal can be used to any degree of classification or sensitivity, depending on the clearance and needs of the user at the given moment. In either of these cases, the authentication and verification mechanisms built into the machine system can be relied upon only to the degree that the data on personnel and on operational characteristics provided it by the security apparatus are accurate.

The second element of organizational leakage points concerns institutional operating procedures. The consequences of inadequate organizational procedures, or of their haphazard application and unsupervised use, can be just as severe as any other malfunction. Procedures include the insertion of clearance and status information into the security checking mechanisms of the machine system, the methods of authenticating users and of receipting for classified information, the scheduling of computing operations and maintenance periods, the provisions for storing and keeping track of removable storage media, the handling of printed machine output and reports, the monitoring and control of machine-generated records for the security apparatus, and all other functions whose purpose is to insure reliable but unobtrusive operation from a security control viewpoint. Procedural shortcomings represent an area of potential weakness that can be exploited or manipulated, and which can provide an agent with innumerable opportunities for system subversion. Thus, the installation operating procedures have the dual function of providing overall management efficiency and of providing the administrative bridge between the security control apparatus and the computing system and its users.

The Task Force has no specific comments to make with respect to personnel security issues, other than to note that control of the movement of people must include control over access to remote terminals that handle classified information, even if only intermittently. The machine room staff must have the capability and responsibility to control the movement of personnel into and within the central computing area in order to insure that only authorized individuals operate equipment located there, have access to removable storage media, and have access to any machine parts not ordinarily open to casual inspection.

Leakage Point Ecology

In dealing with threats to system security, the various leakage points cannot be considered only individually. Almost any imaginable deliberate attempt to exploit weaknesses will necessarily involve a combination of factors. Deliberate acts mounted against the system to take advantage of or to create leakage points would usually require both a system design shortcoming, either unforeseen or undetected, and the placement of someone in a position to initiate action. Thus, espionage activity is based on exploiting a combination of deficiencies and circumstances. A software leak may be caused by a hardware malfunction. The capability to tap or tamper with hardware may be enhanced because of deficiencies in software checking routines. A minor, ostensibly acceptable, weakness in one area, in combination with similar shortcomings in seemingly unrelated activities, may add up to a serious potential for system subversion. The system designer must be aware of the totality of potential leakage points in any system in order to create or prescribe techniques and procedures to block entry and exploitation.

The security problem of specific computer systems must be solved on a case-by-case basis employing the best judgment of a team consisting of system programmers, technical, hardware, and communications specialists, and security experts. This Report cannot address the multitude of details that will arise in the operation of a particular resource-shared computer system in an individual installation. Instead, it is intended that the Report provide guidelines to those responsible for designing and certifying that a given system has satisfactory security controls and procedures. On the other hand, the security controls described in Parts B through D can markedly reduce the probability that an undetected attempt to penetrate a resource-sharing computer system will succeed.

This Report addresses the most difficult security control situation, a time-sharing system serving geographically distributed users. Where circumstances warrant, a lesser set of controls may be satisfactory, and it is not intended that in such cases there be prohibitions on implementing a system with a lesser set of safeguards. The recommendations have been framed to provide maximum latitude and freedom of action in adapting the ideas to specific installations.

V. SYSTEM CHARACTERISTICS

Constraints

The U.S. Government classifies defense information within a well defined and long established structure. Although it might be desirable from the computer point of view to modify these rules, to do so would be equivalent to tailoring the structure to fit the computer operation and would constitute an inappropriate recommendation. Obviously then, a constraint is that a secure computer system must be consonant with the existing security classification structure.

A second constraint, at least initially, is the assumption that the general tenets of the existing, familiar, manual security control procedures will prevail. For example, the Task Force recommendations require not only that a secure computer system *identify* a user, but also that the user establish (prove) his authenticity; furthermore, he will be asked to receipt by a simple response for any and all classified information that is made available to him through any type of terminal. This is a desirable feature, not only from a consideration of system accountability, but also from the point of view of protection for the user. It is conceivable that an error by the computer system might result in an allegation that it had given a user certain information, when, in fact, it had not.

General Characteristics

In formulating its recommendations, the Task Force recognized the following general characteristics as desirable in a secure system.

The system should be **flexible**; that is, there should be convenient mechanisms and procedures for maintaining it under conditions of shifting job assignments, issuance and withdrawal of clearances, changes in need-to-know parameters, transfer of personnel from one duty assignment to another, etc.

The system should be **responsive** to changing operational conditions, particularly in time of emergency. While not an aspect of security control *per se*, it is important that the system be responsive in that it does not deny service completely to any class of users as the total system load increases. It may prove desirable to design special emergency features into the system that can suspend or modify security controls, impose special restrictions, grant broad access privileges to designated individuals, and facilitate rapid change of security parameters.[3]

The system should be **auditable**. It must provide records to the security control supervisor, so that system performance, security safeguards, and user activities can be monitored. This implies that both manual and automatic monitoring facilities are desirable.

The system should be **reliable** from a security point of view. It ought to be fail-safe in the sense that if the system cannot fulfill its security controls, cannot make the proper decisions to grant access, or cannot pass its internal self-checks, it will withhold information from those users about which it is uncertain, but ideally will continue to provide service to verified users. A fallback and independent set of security safeguards must be available to function and to provide the best level of security possible under the degraded conditions if the system is to continue operation.

The system should be **manageable** from the point of view of security control. The records, audit controls, visual displays, manual inputs, etc., used to monitor the system should be supplemented by the capability to make appropriate modifications in the operational status of the system in the event of catastrophic system failure, degradation of performance, change in workload, or conditions of crisis, etc.

The system should be **adaptable** so that security controls can be adjusted to reflect changes in the classification and sensitivity of the files, operations, and the needs of the local installation. There should be a convenient mechanism whereby special security controls needed by a particular user can be

[3]See the definition of Security Parameters, p. 13.

embedded easily in its system. Thus, the security control problem ideally must be solved with generality and economy. It would be too costly to treat each installation as an individual instance and to conceive an appropriate set of unique safeguards.

The system must be **dependable**; it must not deny service to users. In times of crisis or urgent need, the system must be self-protecting in that it rejects efforts to capture it and thus make it unavailable to legitimate users. This point bears on the number and kinds of internal records that the system must keep, and implies that some form of rationing algorithm must be incorporated so that a penetration would capture no more than a specified share of system capability.

The system must automatically assure **configuration integrity**. It must self-test, violate its own safeguards deliberately, attempt illegal operations, monitor communication continuity, monitor user actions, etc., on a short time basis.

Uncertainties

The Task Force has identified several aspects of secure computer systems which are currently impractical or impossible to assess.

Failure Prediction. In the present state of computer technology, it is impossible to completely anticipate, much less specify, all hardware failure modes, all software design errors or omissions, and, most seriously, all failure modes in which hardware malfunctions lead to software malfunctions. Existing commercial machines have only a minimum of redundancy and error-checking circuits, and thus for most military applications there may be unsatisfactory hardware facilities to assist in the control of hardware/software malfunctions. Furthermore, in the present state of knowledge, it is very difficult to predict the probability of failure of complex hardware and software configurations; thus, redundancy is an important design concept.

Risk Level. Because failure modes and their probability of occurrence cannot be completely cataloged or stated, it is very difficult to arrive at an overall probability of accidental divulgence of classified information in a security-controlling system. Therefore, it is difficult to make a quantitative measurement of the security risk-level of such a system, and it is also difficult to design to some *a priori* absolute and demonstrable security risk-level. Since the security risk probabilities of present manual systems are not well known, it is difficult to determine whether a given design for a secure computer system will do as well as or better than a corresponding manual arrangement. This issue is likely to raise considerable discussion at such time as official policy decisions about security control in computer systems must be made.

As described above, computer systems differ widely in the capabilities they make available to the user. In the most sophisticated (and highest security-risk) case, a user can construct both new programs and new programming languages from his console, and embed such new languages into the computer system for use. In such a computer system, offering the broadest capability to the user, the security problems and risks are considerably greater when users from the following two classes must be served simultaneously:

- Uncleared users over whom there is a minimum administrative control and who work with unclassified data through physically unprotected terminals connected to the computing central by unprotected communications lines.
- Cleared users operating with classified information through appropriately protected terminals and communication links.

It is the opinion of the Task Force that it is unwise at the present time to attempt to accommodate both classes of users simultaneously. However, it is recognized that many installations have an operational need to serve both uncleared and cleared users, and recommendations addressed to this point are presented in Parts B through D.

Cost. Unfortunately, it is not easy at this time to estimate the cost of security controls in a computer system. Only a few computer systems are currently in operation that attempt to provide service to a broad base of users working with classified information. While such systems are serving the practical needs of their users, they are the products of research efforts, and good data reflecting the incremental cost of adding security controls to the system and operating with them are not yet available.

In computer systems designed for time-sharing

applications, some of the capabilities that are present in order to make a time-sharing system work at all are also applicable to the provision of security controls. In other computing systems, any facilities for security control would have to be specially installed. Thus, the Task Force cannot give an accurate estimate of the cost of security. It will depend on the age of the software and hardware, but certainly security control will be cheapest if it is considered in the system architecture prior to hardware and software design. In the opinion of some, the investment in the security controls will give a good return in tighter and more accurate accountability and dissemination of classified information, and in improved system reliability.

The cost of security may depend on the workload of the installation. If all classified operations can be accommodated on a single computer, and all unclassified operations on a second computer, the least expensive way to maintain the integrity of the classified information may be to retain both machines. Such a configuration will present operational inefficiency for those users who need to work with both classified and unclassified data bases, but the concept of a dual installation—with one machine working in the clear and a second machine fully protected —cannot be summarily rejected.

VI. DEFINITIONS

There are many terms commonly used in connection with security control for which usage is not completely standardized. Terms used throughout this Report are defined below as a group; certain other terms (especially computer-related ones) are defined at appropriate places in the text.

Clearance. The privilege granted to an individual on the basis of prescribed investigative procedures to have formal access to classified information when such access is necessary to his work. The three formal national clearances are *Top Secret, Secret,* and *Confidential*. However, it is also expedient from the computer point of view to recognize *Uncleared* as a fourth level of clearance. A clearance is a necessary but not sufficient condition to have access to classified information. By extension, the concept of clearance can be applied also to equipment. For example, when a computer terminal is spoken of as having a given level of clearance, it is implied that certain investigative procedures and tests have established that the corresponding level of classified information can be safely transmitted through that terminal. When referring to an aggregation of equipment, together with its management controls and procedures, *facility clearance* is sometimes used.

Need-to-know. An administrative action certifying that a given individual requires access to specified classified information in order to perform his assigned duties. The combination of a clearance and a need-to-know constitutes the necessary and sufficient conditions for granting access to classified information.

Classification. The act of identifying the sensitivity of defense information by ascertaining the potential level of damage to the interests of the United States were the information to be divulged to an unfriendly foreign agent. The classification of information is formally defined in Executive Order 10501. There are only three formal levels of national classification: *Top Secret, Secret,* and *Confidential,* but it is expedient from the computer point of view also to consider *Unclassified* as a fourth level of classification. The identifiers associated with an item of classified information, indicating the level of classification or any special status, are generically called *labels.*

Special Category (or: Special-Access Category or Compartment). Classified defense information that is segregated and entrusted to a particular agency or organizational group for safeguarding. For example, that portion of defense classified information that concerns nuclear matters is entrusted to the Atomic Energy Commission, which is responsible for establishing and promulgating rules and regulations for safeguarding it and for controlling its dissemination. Classified information in a special category is normally identified by some special marking, label, or letter; e.g., AEC information, whether classified Confidential, Secret, or Top Secret, is collectively identified as *Q-information.* It is often called *Q-classified,* but note that this use of *classification* is an extended sense of the formal usage of the word.

Sometimes, special investigative procedures are stipulated for granting access to information in special categories. Thus, while formally there are only

three broadly defined national clearance levels, in practice there is a further structure within each level. In part, this reflects the separation of information into special categories, and, in part, the fact that many different agencies are authorized to grant clearances. For example, an individual functioning within the AEC domain and cleared to Top Secret will often be said to have a Q-clearance because he is authorized access to Top Secret information entrusted to the AEC for safeguarding and identified by the special category Q. These special types of clearances at given levels are not always specifically identified with a unique additional marking or label.

Caveat. A special letter, word, phrase, sentence, marking, or combination thereof, which labels classified material as being in a special category and hence subject to additional access controls. Thus, a caveat is an indicator of a special subset of information within one or more levels of classification. The caveat may be juxtaposed with the classification label, may appear by itself, or sometimes does not appear explicitly but is only inferred. Particular kinds of caveats are:

Codewords. An individual word or a group of words labelling a particular collection of classified information.

Dissemination Labels (Access Control Labels). A group of words that imposes an additional restriction on how classified information can be used, disseminated, or divulged; such labels are an additional means for controlling access. Examples: "No Foreign Dissemination," "U.S. Eyes Only," "Not Releasable Outside the Department of Defense."

Information Labels. A group of words that conveys to the recipient of information some additional guidance as to how the information may be further disseminated, controlled, transmitted, protected, or utilized. Examples: "Limited Distribution," "Special Handling Required," "Group 1—Excluded from Automatic Downgrading and Declassification."

Fully Cleared. An individual who has the clearance and all need-to-know authorizations granting him access to *all* classified information contained in a computer system. By extension, the term can be applied to equipment, in which case it implies that all necessary safeguards are present to enable the equipment to store and process information with many levels of classification and caveated in many different ways.

Security Flag. For the purposes of this Report, it is convenient to introduce this new term. It is a composite term, reflecting the level of classification, all caveats (including codewords and labels), and need-to-know requirements, which together are the factors establishing the access restrictions on information or the access privileges of an individual. By extension, the concept can be applied to equipment, and indicates the class of information that can be stored and processed.

Thus, the security flag contains all the information necessary to control access. One security flag is considered to be equal to or higher than a second if a requestor with the first flag is authorized access to information which has the second flag.

Security Parameters. The totality of information about users, files, terminals, communications, etc., which a computer system requires in order to exercise security control over the information that it contains. Included are such things as user names, clearances, need-to-know authorizations, physical location; terminal locations and clearances; file classifications and dissemination restrictions. Thus, a set of security parameters particularizes a generalized security control system to the specific equipment configuration, class of information, class of users, etc., in a given installation.

Part B

POLICY CONSIDERATIONS AND RECOMMENDATIONS

The policy recommendations that follow are intended to provide a security skeleton around which a specific secure computer system may be built. Additionally, these recommendations set forth the responsibilities and functions of the personnel needed to evaluate, supervise, and operate a secure system. This is a new field, and this Report represents the first major attempt to codify its principles. In some cases, the rationale behind a specific recommendation and appropriate examples are presented in a Comment.

I. FUNDAMENTAL PRINCIPLES

Automatic data processing systems shall accommodate, without exception, the responsibilities of individuals to ensure that certain official information affecting national defense is protected against unauthorized disclosure, pursuant to Executive Order 10501 (Amended), "Safeguarding Official Information in the Interests of the Defense of the United States."

A computer system shall grant access to classified information only to persons for whom it can determine that their official duties require such access, and that they have received the proper security clearances and need-to-know authorizations.

The means employed to achieve system security objectives shall be based on any combination of software, hardware, and procedural measures sufficient to assure suitable protection for all classification categories resident in the system.

To the maximum extent possible, the policies and procedures incorporated to achieve system security shall be unclassified. However, specific keys, passwords, authentication words, and specifically designated sensitive procedures shall require classification.

Comment: These principles reflect the constraint that the recommendations of the Task Force be consistent with generally accepted, existing security doctrine. The last item is considered relevant in order to permit maximum operational convenience.

II. SYSTEM PERSONNEL

Depending upon the nature of the individual computing installation, some or all of the following categories of personnel will be associated with it. It is recognized that a given individual may have more than one responsibility, and either simultaneously or at different times perform more than one function. It is also recognized that the scope of responsibility may imply a substantial organizational group for each function.

Responsible Authority. The head of the department or agency responsible for the proper operation of the secured computer system.

User. Any individual who interacts directly with the computer system by virtue of inserting information into the system or accepting information from it. "Information" is considered to include both computer programs and data.

Comment: A user is thus defined whether he interacts with the system from a remote terminal or submits work directly to the computing central through a batch-process mode.

System Administrator. An individual designated as responsible for the overall management of

all system resources, both the physical resources of the system and the personnel attached to it.

Comment: The users are generally excluded from the System Administrator's management purview, although personnel under his control may also be users at times.

System Certifier. An individual designated by an appropriate authority to verify and certify that the security measures of a given computer system and of its operation meet all applicable, current criteria for handling classified information; and to establish the maximum security level at which a system (and each of its parts) can operate.

System Security Officer. An individual designated by a Responsible Authority as specifically responsible for (1) proper verification of personnel clearances and information-access authorizations; (2) determination of operational system security status (including terminals); (3) surveillance and maintainance of system security; (4) insertion of security parameters into the computing system, as well as general security-related system matters; (5) security assurance.

Comment: The System Certifier will establish the maximum security level at which the system (and each part of it) can operate; the System Security Officer will determine on an operational basis the level at which it does operate. He will normally verify personnel clearances with the overall security officials of the organization, and need-to-know authorizations with the organizational element that has cognizance over the information in question (e.g., an Office of Primary Interest).

Security assurance implies an independent group that continuously monitors security provisions in the computer system. It includes such functions as continuously probing the system to ascertain its weaknesses and vulnerabilities, recommending additional safeguards as need is determined, and validating the security provisions in a system. Because of the technical expertise implied by security assurance, it is probable that this responsibility will be shared by the System Certifier.

System Maintenance Personnel. The individuals designated as responsible for the technical maintenance of those hardware and software system features that (1) must operate with very high reliability in order to maintain system integrity with respect to security matters, and (2) maintain the basic functioning of the system.

Comment: The hardware and software maintenance personnel are permitted to service not only the normal, basic features of the computing system, but also the security control features. However, there need be no prohibition on the assignment of these two classes of maintenance requirements to separate individuals or groups of individuals.

System Operators. Those personnel responsible for performing the manual procedures necessary to provide and maintain on-going service operations of the system.

Personnel Designations and Responsibilities

System Administrators, System Security Officers, and System Maintenance and Operations Personnel shall be formally designated by the Responsible Authority. The total number of such personnel should be kept to a minimum. Where necessary to meet special operational needs of a particular installation, special restrictions affecting personnel may be incorporated into the individual agency's procedures, formulated under the cognizance of the Responsible Authority.

Comment: This recommendation is intended to permit installations that have special operational needs, either because of mission or sensitivity of information, to impose additional constraints on system personnel or on their responsibilities.

As a general approach, it is desirable that persons designated as System Personnel have sufficient clearance and need-to-know authorization for all information resident in the computer system. However, it is conceivable that even for System Personnel, access could be segmented so that such clearance would not be absolutely necessary. For example, Operators and Administrators may not have access to the keys or mechanism that allow access to the interior of the hardware. This policy will accommodate either approach as found to be necessary by the exact nature of the computer system involved and the information to be protected. A typical user-agency decision might be to limit System Personnel to U. S. Government personnel, or to special two-man teams, each of which

may be limited to partial access. Another user-agency decision might be to require some degree of sanitization preliminary to the performance of certain types of system maintenance, especially if the person capable of performing such maintenance is not or cannot be cleared adequately. Sanitization refers to the protection of classified information resident in computer files either by deliberate erasure or by physically removing and/or protecting the storage medium or device.

Although it is recognized that System Personnel may fulfill more than one responsibility, this option may not be exploitable in practice because of the significantly different skills required. For example, skilled and experienced system programmers will be required to maintain the software, whereas computer engineers will be required for the hardware, and communication engineers for the communications.

User Designation

Each user (or specific group of users) shall be administratively designated (identified) to the computer system by the System Administrator, with the concurrence of the System Security Officer. The designation shall include indicators of the user's status in sufficient detail to enable the system to provide him with all material to which he is authorized access, but no more.

Comment: As will be seen in the Appendix, which defines a language and schema for identifying both a security structure and security parameters to a computing system, the number of parameters that must be kept within the system for each user will reflect the kind of classified information with which the system deals. In some instances, it will be necessary to verify more than a user's clearance and need-to-know status before access to classified information can be granted; e.g., it may be necessary to verify his agency of employment. It may also be desirable to keep within the computing system extensive information on each user, not for routine verification of his access privileges, but for the convenience of the System Security Officer when he finds it necessary to intervene in the system's operation.

User Authentication

Each user shall be required both to identify himself and to authenticate his identity to the system at any time requested by it, using authentication techniques or devices assigned by the System Security Officer. Such techniques or devices shall be sufficient to reduce the risk of unauthorized divulgence, compromise, or sabotage below that required by the sensitivity of the data resident in the system.

Comment: Identification is for the purposes of system accounting and billing, whereas authentication is the verification procedure necessary before the system can grant access to classified information. The choice of technique or device obviously will depend on the sensitivity of the data resident within the computing system, the physical location of the user terminal, the security level to which it and its communication links are protected, the set of users that have access to it at any time, etc.

User Responsibility

A properly authenticated user is responsible for all action at a given terminal between the time that his identity has been established and verified, and his interaction with the system is terminated and acknowledged. Termination can occur because he notifies the system of his departure, or because the system suspends further operation with him. The user is responsible for observing all designated procedures and for insuring against observation of classified material by persons not cleared for access to it; this includes proper protection of classified hard copy. Furthermore, he is responsible for reporting system anomalies or malfunctions that appear to be related to system security controls to the System Security Officer, especially when such occurrences suggest that system security control measures may be degraded, or that a deliberate attempt to tamper with or penetrate the system is occurring. Other system anomalies should be reported to System Maintenance Personnel, who, in turn, must report to the System Security Officer those hardware or software malfunctions that investigation shows have affected security controls.

Access

Access to classified information stored within the computer system shall be on the basis of specific authorization from the System Security Officer to

receive such information, or by automatic processes operating under his control and authority. The authority of the System Security Officer to authorize system users to have access to classified information stored in the system does not implicity apply to the System Security Officer himself. Separate and specific restraints over his access to classified information shall be established by the System Administrator. A specific algorithm (or combination of algorithms) for controlling access to all classified information shall be specified and embedded in the system. Moreover, a specific protocol and mechanism shall be specified for inserting into the computer system those security parameters that grant and rescind access privileges. For both purposes, hardware, software, and procedural mechanisms shall be implemented that insure that neither the access control algorithm nor the security-parameter insertion mechanism is circumvented, either accidentally (through component failure) or intentionally.

Comment: This recommendation establishes the general principle on which user access to classified information within the system is granted. The details of the algorithm that permits access to classified information obviously will depend on that part of the total security structure with which the computer system is concerned, and also on the status information kept within the system for each user. The Appendix illustrates a particular algorithm that appears to be sufficiently comprehensive to cover all requirements known to the Task Force. It should be noted that this recommendation attempts to incorporate redundancy into the access control mechanism, and also into the parameter insertion mechanisms, by requiring a combination of hardware, software, and procedural mechanisms.

III. INFORMATION STRUCTURE AND TRANSFORMS

Data storage shall be organized and controlled at the level of the basic computer system in terms of information units, each of which has a classification descriptor plus applicable special-access categories (as required by the presence of caveats) and other labels that apply to the information unit as a whole. It is the explicit responsibility of the individual directing a computational process to declare and verify the classification and any applicable caveats and other labels for an information unit produced as a result of some computer process (e.g., calculations of bomber ranges or weapon effectiveness), or as a result of a transformation of some previously existing unit (e.g., merging or sorting of files).[1] This responsibility extends to security control and management of information subunits. Procedures analogous to those in force for controlling introduction of information from or release of information to entities outside the system must be observed, and are described in Sec. VI below, "Information Security Labels." Since a hierarchical structure of information classification will usually exist, a composite unit must be at least at the highest level of classification of the units contained in the composite, but, in fact, may be higher. Automatic algorithms may be used to aid the user in the execution of these responsibilities.

Comment: The intent of this recommendation is to provide procedures analogous to those for handling documents, as specified in Section 3 of Executive Order 10501 (Amended). The recommendation on information structure and transforms leaves unspecified whether a computer-based file is classified as an entity, or whether the individual entries or elements of the file are separately classified. The design of the file structure and the details of how it shall be classified are operational matters, not a problem of providing security control mechanisms. However, where the security structure of the file is established, the procedures outlined in this recommendation will apply.

This recommendation also permits the use of computer algorithms to assist in classifying new information. In the Appendix, examples are given which suggest how such algorithms may be applied, but the computer system may not be able to establish classification level or applicable special caveats and labels in every circumstance. At most, the system can tell a user that he has had access to classified information

---

[1]This statement is not adequate for nongovernmental organizations, nor in some government situations. For example, an employee of an industrial contractor can only suggest the classification of information which he creates; the formal declaration of classification is made by a designated, appropriate authority, sometimes external to the contractor company. Some secure computer systems will require a supplementary procedure to validate classifications suggested by users.

with given caveats and labels; it will be his responsibility to confirm to the computer system the classification, special caveats, and labels that should apply. If the sensitivity of the information warrants, audit information should be made available to the System Security Officer, informing him that a user has taken some specified action in establishing or modifying a clearance level, applicable caveats, or labels.

IV. SYSTEM TRANSACTION ACCOUNTING

Logging of Transactions

All relevant transactions between users and the computer system shall be automatically logged (including date and time) by the computer system so that an audit of transactions involving access to and generation, classification, reclassification, and destruction of files is possible. The provisions of this paragraph also apply to unclassified information that resides in a system containing, or cleared to contain, classified information. Supplementary manual logs (including date and time) must record all significant events that cannot be automatically logged.

Comment: Transaction as used here includes such things as a user logging onto or off the system; the system granting a user access to a specified file; the merging of files by a user; the generation of new information to which a user assigns classification; changes made in a classified file by a user; and exchanges of information with another computer. The inclusion of unclassified information is intended to provide for the case where "unclassified" information becomes upgraded, and to protect against unobserved activity in the manipulation of the system by users. The audit-trail data should be made available to the System Security Officer to aid him in the continuous monitoring of the security of the system.

It may prove operationally desirable to aggregate information of this type and present it in various periodic reports. Thus, for example, the System Security Officer could be informed at the end of each shift as to which files have been addressed by or released to each user, or which files have been updated or had their classification changed. The control of security overlaps somewhat the control of file integrity, and it may prove desirable for some of the audit information to be made available to the System Administrator.

The number and kinds of audits and the periodicity with which they are made will depend on such factors as sensitivity of the information contained in the computer system, the class of users it services and their clearance status, the operational requirements of the system, etc. Some portions of the status log will be only historical, others will be used operationally. It is conceivable that in some installations it will prove desirable to provide the System Security Officer with a visual display of the system transaction log.

It should be noted that when the System Security Officer is interacting with the system (e.g., inserting new security parameters), he is considered by the system to be a user. Thus, even though his actions are privileged and executable only by himself, his activities will be automatically logged. Furthermore, maintenance personnel will also be considered users when their activity can be accomplished with the system in an operational status, and their actions will also be automatically logged. Finally, the interactions of the operating personal, especially the console operators, will be considered as user activity and logged.

Receipting

Where required by applicable regulations, a receipt shall be obtained from any user who has received classified information from the system. Receipting shall require an overt action on the part of the user following delivery (or presentation) to him of the classified information. The purpose of the receipt is to insure that the user is aware that he has received classified data. For the purposes of this requirement, the bounds of a dialogue between a user and the computer system are defined to be based on the beginning and ending of access to a particular unit of information contained within the system or transferred to or from the system.

Comment: While a properly functioning system already knows, to the degree adequate for logging of system activity, where information should be or to whom it has been delivered, the requirement for a receipt recognizes a need for an acknowledgment

from the recipient (person or program) that he is aware that he has received classified information of a particular level. It is essential for system efficiency and man-machine effectiveness that the receipting procedure not be imposed excessively. Thus, definition of appropriate transaction boundaries is crucial. Although it is undesirable to burden the user with unnecessary actions, nonetheless it may be to his advantage to require a receipt for all information. He will be aware of, and the system transaction log will reflect, precisely the information to which he has had access. His liability is therefore defined, and any investigation which later may arise because of a system malfunction or divulgence of classified information would be facilitated.

V. RELIABILITY AND AUTO-TESTING

All security control or assurance mechanisms and procedures shall be designed to include sufficient redundancy and independent checks so that the failure of one control mechanism will not allow an undetected compromise to occur. Frequent automatic checks of these protection mechanisms by the computing system itself, and periodic checks of the procedures by system personnel shall be made. The computing system shall have the capability of guaranteeing that some specified minimum fraction of its time is spent on performing automatic system checking. The percentage of time spent on automatic checking shall be a design parameter of the computing system (capable of change at the local installation as necessary), and shall be established with the concurrence of the System Certifier. The interval between automatic internal self checks may depend on the classification and sensitivity of the information that the system is designed to accommodate. The System Security Officer shall be provided means for establishing what fraction of the time the installed system spends in self-checking and be responsible for controlling the time so spent, depending on the classification and sensitivity of the information that his system is handling. Means shall be provided for the System Security Officer to initiate these checks manually.

A detected failure of the protection mechanisms shall cause the system to enter a unique operating mode wherein no information may be transmitted to or accepted from the user community. In order that there be no unnecessary interruption of services, the system must concurrently check all its internal protection mechanisms. Should the detected failure prove to be the consequence of a transient error, the system should so notify the System Security Officer and be returned to its full operational status by an overt action of the System Security Officer. In the event the failure persists, it shall be the responsibility of the System Security Officer to take any action indicated. He may return the system to full or partial operational status in spite of impaired security controls; he may attempt to remove malfunctioning equipment and restore a modified configuration to full status. In any event, the action required of him must be sufficiently overt that the possible security implications of his action will be patently clear.

Special instructions shall be provided to the System Security Officer in those installations that deal with information of high sensitivity, and for which special procedures are deemed necessary in order to insure that the system is not allowed to operate in a manner that increases the risk of compromise or unauthorized disclosure.

Comment: The issue raised by this recommendation is a delicate one because it addresses a conflict between policy objectives of the system: maintaining service to the users of a computing system, and maintaining proper security control over the information stored within it. If an agent knows how to create an error on demand, total shutdown of a system when trouble is detected is a serious vulnerability. Thus, a capability for flexible response, depending upon the conditions of the moment, is essential. The action taken by the System Security Officer, perhaps in conjunction with the Responsible Authority or the System Administrator, must reflect the operational situation that the system supports. In a military command and control system where delay can mean disaster, operational urgency may dictate that a calculated risk of unauthorized divulgence be assumed in order to maintain continued service to users. On the other hand, a technical information system can afford to suspend service totally in case of trouble, especially if it deals with very sensitive information.

The fraction of its time that a computing system must spend in self-checking, and the scope and depth

of such self checks are not matters that can be assessed readily by the local System Security Officer. Hence, this recommendation requires that the problem be addressed at the level of design and installation certification. However, it is reasonable that the System Security Officer have the option of adjusting the periodicity and depth and scope of self-checking, according to the level of information that his system must accommodate.

It is not possible to make positive statements about the frequency with which internal self-checking must be performed. In part, this reflects lack of insight into and experience with the security control mechanisms to be installed in the computing systems under consideration. It may be desirable to perform internal self-checking on some scheduled periodic basis, or, perhaps more wisely, the internal self-checking should take place on an aperiodic basis, such as when a user from a terminal requests access to a file. Aperiodic checking denies a potential penetrator the assurance that he has guaranteed intervals of time in which to attempt to subvert or bypass the security control mechanisms, but it also increases the self-checking load on the machine as the user load increases. In any event, the maximum interval between internal self-tests should be chosen jointly by the user-agency and the System Security Officer. The objective is to find an acceptable balance between system efficiency and the amount of classified information that could be compromised between tests, while maintaining a risk acceptable to the user-agency.

In the event of an automatically detected failure of a control mechanism, it is clear that the computing system must shift to a degraded mode of operation because of the risk of unauthorized divulgence. However, the system design must be such that the system attempts to maintain maximum service to the greatest number of users. It is also clear that the issue transcends the computing central and its procedures; a response to malfunction can also involve communications, remote terminals, other computers, etc.

The degraded mode suggested by the wording of this recommendation seems to be reasonable, but it is not the only possibility. Another, for example, is to bring the System Security Officer into the access control procedure and let him manually verify each user request for access to a given file. If such a procedure were to be implemented, the System Security Officer would need to be provided with a great deal of visually displayed information and with appropriate manual controls over system performance.

Typical actions that the System Security Officer might take, depending on the type of failure detected and upon the operational urgency of the moment, include:

(a) Disabling the system completely—i.e., closing it down and requesting maintenance.
(b) Continuing to operate the system in the degraded mode, but under his continuous manual surveillance.
(c) Prohibiting new users, while allowing current users to continue interaction with files presently accessible to them.
(d) Restricting access to classified files to those terminals over which he or some other responsible authority has visual cognizance. Alternatively, he might suspend all but fully-cleared users.
(e) Denying all user requests to access files of special sensitivity.
(f) Electrically severing malfunctioning storage devices, thus permitting the balance of the system to continue in operation. If these devices contain the security control and checking programs and authentication words, etc., then a choice must be made between this option and point (g) below.
(g) By-passing all security checks and operating the system "wide-open."
(h) Electing to operate with unprotected communications.

It is reasonable that the system be designed so that the action options available to the System Security Officer can be automatically presented to him by the system itself. It is also reasonable that each option displayed be accompanied by instructions detailing the manual and procedural actions that he ought to take.

Ultimately, the amount of self-checking incorporated into a system, the frequency with which self-checking is done, and the precise details of how the system functions in a degraded mode, will represent a design compromise between maintaining maximum service to the users and maintaining maximum safety of the information resident within the system. When circumstances warrant, the system can be designed to automatically go into a more extensive

mode of internal self-checking, or even to switch automatically to alternate software packages that can substitute for malfunctioning hardware or software protection mechanisms.

VI. INFORMATION SECURITY LABELS

Information Input

The system shall not accept information, even for temporary use, without first receiving from the user a declaration of the relevant security parameters, which in this case include classification, all caveats, and labels. These parameters will be used by the system to control further use or dissemination of the information. The security parameters can be handled as a declaration covering a definable set of interactions between a user and the system—e.g., the totality of a dialogue between user and system, beginning when the user logs on and ending when he logs off. The capability for specifying security parameters as a declaration covering a set of interactions is provided in order that the user not be burdened with specifying security information more often than absolutely necessary.

Comment: The requirement that the security parameters be specified before the system will accept information is simply a fail-safe mechanism to avoid oversight on the part of a user. It is reasonable that the system assist the user by asking him in turn for level of classification, codewords, dissemination labels, and information labels (as applicable). Where possible, the system should automatically apply any caveats, labels, etc., implied by information already supplied. It is also reasonable that, on request, the system provide the user with a listing of labels so that he can assure himself that nothing has been overlooked.

Information Output

Each user shall be notified of at least the classification level and special access caveats of all information being furnished him by the system. Where physical limitations prohibit or discourage presentation of all caveats and labels associated with each separate page or display of information, means must be provided for the user to obtain them at his request.

Comment: Ideally, all information provided a user, whether printed out in hard copy or electronically displayed, should be accompanied by all relevant security parameters. However, practical limitations in the capabilities of display devices or printers may make alternative procedures necessary. At the minimum, the classification level must be displayed or printed with each page. The user must be able to obtain the complete set of security parameters associated with information when he is being asked to receipt for it.

VII. MANAGEMENT OF STORAGE RESOURCES

User-to-User Leakage

Allocation, use, and erasure of storage resources of all types in the computing system shall be handled both by the system and by operational procedures in such a way that no information from a prior use of the storage medium can leak to the current use.

Comment: The consequence of this recommendation is to require that appropriate schemes for management of storage allocation and erasure of storage be incorporated into the system software and system operational features. The problem of leakage concerns both complete and fragmentary pieces of information, and entire as well as partial quantities of storage. For example, the scratch space on a magnetic disc assigned to one classified job must be satisfactorily sanitized before assigning it to a second job. The problem of leakage would be greatly facilitated if magnetic tape transports contained a rewind-and-erase feature, and magnetic discs a read-and-erase feature.

Residual Information

A storage medium shall carry the same classification as the most highly classified information stored on it since the most recent sanitization. All sanitization (e.g., degaussing) shall be done in such a way as to insure that even if the medium were removed

from the computing system and subjected to tests under laboratory conditions, no residual information could be extracted from it. The alternative to sanitization is to treat the storage medium as classified until destruction.

This requirement does not imply that all information read from a storage device must be treated as if it were classified to the highest level of any data ever recorded on the medium. Information extracted from the device by normal means (e.g., via the computer system) may be properly handled at the classification of the information *per se*, provided, however, that all other criteria that relate to handling of information at that classification level are satisfied.

Sanitization Procedures

The specific techniques and tests required to insure sanitization of storage media, as required in the preceding paragraph, shall be at the discretion of a Responsible Authority.

Comment: Currently, there is no sanitization technique or equipment generally available that will consistently degauss any and all media so thoroughly that residual information cannot be extracted under specialized laboratory conditions. Additional research and testing are needed to determine the validity of various procedures now used, and to develop new procedures, equipment, and tests. It is recommended that research continue, and, to the maximum extent possible, that duplication of efforts be avoided. Results should be made available through the Department of Defense. Meanwhile, responsible authorities must have leeway to select the degaussing technique proven best for the particular media under their control.

VIII. SYSTEM CERTIFICATION

Certification is the process of measuring, testing, and evaluating the effectiveness of the security control features of a system. It must be accomplished before a system can be used operationally with classified information. The three types of system certification are Design Certification, performed before and during system construction; Installation Certification, performed prior to authorizing a system for operational use; and Recertification, performed after major changes or correction of failures.

Comment: The problem of certifying that a computer system contains a properly functioning set of security safeguards and is operated under an appropriate set of operational procedures is complex and difficult. The issue is considered at this point in connection with policy and operational recommendations, but is also discussed later in the context of hardware recommendations. The precise details of an adequate certification procedure, including the necessary inspections and tests, are difficult to define, although it is clear that the details of such procedures will depend, in part, on the type of computer system in question, and on the scope and type of service that the system furnishes its users. System certification is the crucial process in establishing the classification level permissible in a secure system.

Certification of an overall system, determined on the basis of inspection and test results, shall be characterized in terms of the highest classification or most restrictive specific special-access categories that may be handled. Where tests show that the overall system can effectively maintain the integrity of boundaries between portions of the system, certification may differ for various portions (i.e., for "subsystems").

Comment: This recommendation establishes a convenient way to characterize the certification of a system or portions of it. By permitting certification to differ for portions of a system, we have in principle permitted part of a system to function in an uncertified condition, but subject to tests that demonstrate that the system can effectively maintain the integrity of subsystem boundaries. It is not certain at the present time that tests can adequately establish the integrity of boundaries, thus permitting inclusion of an uncertified portion in a system. In general, the more highly classified and sensitive the information in a system, the more carefully one should consider the risks before permitting an uncertified portion to operate in the overall system.

Tests and Inspections

Any computer system used to process classified information shall be subjected to inspection and test by expert technical personnel acting for the Responsible Authority. The extent and duration of the in-

spections and tests shall be at the discretion of the Responsible Authority. The inspections and tests shall be conducted to determine the degree to which the system conforms to the requirements here recommended, any derivative regulations, and other applicable regulations.

Comment: This recommendation does not specify the details of tests and inspections to be conducted, nor does it specify when such tests and inspections are necessary. Furthermore, it does not prohibit the Responsible Authority from using expert technical personnel from an external agency or department. On the contrary, some of the tests and inspections should be conducted by an external group. Where the sensitivity of the information in the system warrants, some of the tests, inspections, and deliberate diagnostic attempts at penetration should be conducted on an unannounced basis. It is not implied that the extent and nature of the tests and inspections necessarily be the same for each of the types of system certification.

Types of System Certification

Design Certification. A series of tests and inspections that establish that the safeguards designed into the hardware and software of the system are operative, function as intended, and collectively constitute acceptable controls for safeguarding classified information. Production models of a given design need be tested only to verify that all safeguards are present and properly functioning. It is recommended that this certification be performed by an agency or a special team not part of the using agency and separate from design or maintenance groups. Specifications (procedures, tests, inspections) for subsequent certification reviews must be produced as part of the design certification process.

Installation Certification. A series of tests and inspections performed according to specifications established during the design certification phase to insure that the required set of security safeguards (hardware, software, and procedural) are in fact present and operational in the installed equipment, and on all communication links that will carry classified information to remote terminals or other computers. This certification must also examine the operational procedures and administrative structure of the organization that controls the equipment, and must establish that the procedural and administrative environment supplements and complements hardware and software safeguards, and that physical safeguards are appropriate. It is anticipated that certification review will be most extensive and thorough at the time of initial installation of the system. Installation certification will probably be conducted by a special team, not necessarily under the control of the Responsible Authority. Ideally, the System Security Officer will participate in this certification so that he becomes familiar with the safeguards in the system and with the process and intent of certification in order that he can conduct subsequent certifications.

Recertification. Some level of recertification must be accomplished periodically, as indicated by operational circumstances. These instances are as follows:

Periodically during the operational life. It is desirable to recertify the system at intervals during its lifetime. This is in the nature of a preventive procedure to establish the continuity of security safeguards, to make gross checks on system functioning, and to search for loopholes in the protection. It is conceivable that some level of recertification might be desirable at the beginning of each scheduled shift of operation or on some other periodic basis, as dictated by the needs or sensitivity of the computing installation.

After system malfunction. Depending upon how the system has malfunctioned and on what remedial action has been taken, some recertification procedures are desirable to re-establish that the security controls are fully functioning. The responsibility for determining which recertification tests and inspections are necessary rests with the System Security Officer, although he may solicit expert opinion from System Maintenance Personnel or the System Administrator.

After scheduled or unscheduled hardware or software maintenance or modification. As with system malfunctions, some level of recertification undoubtedly is necessary after modifications have been made in the computing equipment or the system software. The scope and depth of these tests and inspections should reflect what maintenance has been performed and what changes have been made. The ultimate judgment as to which recertification procedures are necessary must be the responsibility of the System Security Officer,

although he may solicit expert opinion. For sufficiently extensive modifications or maintenance, the recertification procedure may well approximate the extensive set of tests and inspections made at the time of initial installation.

Comment: The Task Force does not recommend any particular recertification periodicity, but suggests that initially, at least, the question of periodic inspection and recertification be jointly determined by the System Security Officer and the Responsible Authority. As each acquires confidence in the capability of the system to maintain satisfactory security control, it is likely that the intervals between tests and recertifications will be adjusted accordingly.

Automatic internal self-testing previously described can be regarded as a form of recertification that takes place on a short time scale (e.g., milliseconds), as opposed to the type discussed above which occurs on a long time scale (e.g., hours, days).

Operational Security Parameters

The necessary operational security parameters of the overall system, or of each portion of it, shall be inserted into the system by the System Security Officer.

Comment: This recommendation is consistent with the view that the security apparatus of the agency that operates a computing system has the necessary overall view to be able to specify the relevant security parameters for the system. The recommendation also reflects the requirement that the System Security Officer be responsible for the currency and accuracy of the parameters in his system. The point is included as part of certification because proper tests and inspections must be conducted in order to ascertain that the security parameters have in fact been correctly inserted into the system (and accepted by it), both initially and each time the security parameters of the system are modified.

Protection at Boundaries

Information shall be passed to or accepted from any portion of the system only at a security level commensurate with the security parameter for that portion of the system. The use by an uncleared person of a terminal certified for highly classified information is permissible without the need for recertification as long as precautions (escorting, continuous surveillance to prevent tampering, etc.) are taken to prevent subversion of the security mechanisms needed (and previously certified as effective) to protect the stipulated classification of the terminal.

Comment: The impact of this recommendation on the clearance specified for a remote terminal is complex. In effect, it requires that the clearance assigned to a given terminal be determined by appropriate tests and safeguards that are commensurate with the highest classification of information to be handled. Temporary operation of the terminal with information of a lower classification is acceptable, providing that adequate measures are taken to maintain the integrity of the certified status of both the terminal and its environment. There must be safeguards that insure that the system responds to each user appropriately to his clearance, and tests must be applied during the various certification phases that verify the presence and efficacy of these protection mechanisms. Extra precautions must be taken before and after the use of a terminal by an uncleared person. Following use of a terminal by a person not cleared to receive information classified equivalent to the terminal's maximum clearance, authentication of a new user is mandatory before initiating transactions involving higher classifications. In establishing his authenticity, the new user is also tacitly indicating that the former user is no longer in a position to monitor the higher classification transactions.

Post-Certification Changes

Changes in the hardware or software of the system shall be installed for normal operations only by the designated System Maintenance Personnel or personnel operating under their observation and supervision, with the concurrence of the System Security Officer. An explicit report of all such changes shall be made to the certifying authority for the particular system, in addition to the normal manual and/or automatic logging of system transactions.

Comment: This recommendation requires explicit reporting of all changes in system hardware or software. If such changes are sufficiently minor in the opinion of the System Security Officer or the System Certifier, then reporting may be sufficient. However,

if, in the opinion of the System Certifier or the System Security Officer, the changes are sufficiently major that security safeguards may have been affected, then some level of recertification tests and inspection will be essential.

Continuity of Physical Protection

Equipment and associated materials (e.g., media containing copies of programs) used for handling classified information must be continuously protected against unauthorized change commensurate with the security level at which they most recently have been certified. Copies of operating software that is not itself classified and which is not to be used for actual insertion into the system or to generate programs for insertion into the system need not be subject to this requirement.

Comment: This recommendation is intended to guard against the implantation of intelligence sensors or software changes that might aid penetration of safeguards. Note that it does not require the items to be classified, nor does it require physical protection for all copies of an item. For example, several copies (e.g., on card decks or magnetic tapes or discs) of the operating system software will usually exist. Only that copy to be inserted into the machine for actual running of the system and the master copy from which it was made must be physically protected as required; even then, protection need commence only after a copy has been certified to be correct. Other copies, which are for the convenience of maintenance personnel or system operators and which will not be used to make additional copies or used operationally in the system when it contains classified information, need not be protected. This recommendation should also aid in avoiding unnecessary classification of equipment or software.

Part C

TECHNICAL RECOMMENDATIONS

I. INTRODUCTION

It is important to understand what present technology can and cannot do in protecting classified information in a resource-sharing system. Present technology offers no way to absolutely protect information or the computer operating system itself from *all* security threats posed by the human beings around it. As a consequence, procedural and administrative safeguards must be applied in resource-sharing computer centers to supplement the protection available in the hardware and software.

As could be observed in the policy recommendations, there are two types of environments in which secure computing systems operate. One is an environment consisting of only cleared users who function at physically protected terminals connected to a physically protected computing central by protected communication circuits. The main security problem in such a *closed environment* is largely one of maintaining the data and program integrity of each individual user. An inadvertent divergence of classified information by the system is analogous to a cleared person finding a classified document for which he is not authorized access. The other type of environment is one in which there is a mixture of uncleared users working at unprotected consoles connected to the computing central by unprotected communication circuits, and cleared users with protected consoles and protected communication lines. The security problem with such an *open environment* is that the system must be able to withstand efforts to penetrate it from both inside and outside.

For purposes of this Report, the terms *closed system* and *open system* are used to indicate security controlled computing systems that operate in these wholly different but realistic environments. From a technical point of view, a secure closed system (i.e., one acceptably resistant to external attack, accidental disclosures, internal subversion, and denial of use to legitimate users) while presenting difficult problems, can be provided by contemporary technology; but a secure open system cannot be provided by contemporary technology. In fact, there is special concern about the risk of compromise of classified information and the vulnerability of an open system to potential penetrations because, as of today:

(a) It is virtually impossible to verify that a large software system is completely free of errors and anomalies.
(b) The state of system design of large software systems is such that frequent changes to the system can be expected.
(c) Certification of a system is not a fully developed technique nor are its details thoroughly worked out.
(d) System failure modes are not thoroughly understood, catalogued, or protected against.
(e) Large hardware complexes cannot be absolutely guaranteed error-free.

Since adequate controls cannot be provided by technology alone, it is necessary to rely on a combination of hardware, software, and procedural safeguards. Thus, some of the recommendations below refer to issues already discussed in Part B.

The precise mix of controls and safeguards necessary in any given case will depend on the operational environment, sensitivity of information, class of users, and types of service rendered, as noted above. We believe that these recommendations are both

necessary and sufficient for a closed secure system. However, their sufficiency for an open system cannot be guaranteed in the abstract. Only by intelligent adaptation to a specific open environment utilizing experience from closed systems and by extremely objective and stringent testing and evaluation can their adequacy be established for a specific open system.

II. CENTRAL PROCESSOR HARDWARE

Central processor hardware must provide some or all of the following mechanisms, depending on the class of service it renders its users: user isolation; supervisory software[1] protection; and assurance against unanticipated conditions.

User Isolation Mechanisms

Each user (or worker) program[2] must be isolated from all other programs in the computing system. The currently known principal hardware mechanisms for isolating programs include base-addressing registers and various forms of hardware checking circuits to assure that memory addresses generated within the processor are in fact restricted to those permitted for the programs of a particular user. In addition, some contemporary machines provide memory protection through length-check registers, bounds registers, and storage locks.

The characteristics of the system software determine whether or not user-isolation hardware features are required on systems that provide the user with a file-query capability (Type I in Fig. 2), or with full programming capability through an interpretive mode or in a restricted set of languages with checked-out compilers; (Types II and III in Fig. 2). Sometimes, the hardware features are not necessary in principle, but as a practical matter the use of relevant hardware features greatly simplifies the achievement of isolation. It is recommended that hardware user-isolation mechanisms be required for all resource-sharing systems of Types I, II, and III (in Fig. 2).

It is recommended that isolation hardware be mandatory in systems that provide extensive programming capability to the user in any language and with any compiler of his choice, including the machine language of the computer (Type IV in Fig. 2).

While many contemporary machines designed for multiprogramming or time-sharing environments incorporate hardware safeguards that provide user isolation, there is very little internal hardware self-checking to guard against malfunctions. Older machines operating in a security controlling mode may not be able to fully meet these recommendations. To some extent, user isolation achieved by means of hardware mechanisms can be exchanged for isolation via software mechanisms. This should be done with caution, for the protection mechanisms effected by software-means must themselves be safeguarded against collapse due to a hardware or software malfunction.

Supervisor Protection

The objective of Supervisor protection is to deny a user program the ability to penetrate the Supervisor (which contains security control safeguards) without detection by the Supervisor. A user program might attempt such a subversion for the purpose of manipulating supervisory information in such a way as to disable security control barriers, or to pre-empt the system and so deny service to other users.

It is recommended that computer systems that provide for programming via interpretation or via limited languages and checked-out compilers, and systems that provide extensive programming capabilities (Types II, III, and IV in Fig. 2), incorporate hardware techniques that have the effect of providing at least two distinct operating states: the user state and the supervisor state (also called worker or slave, and master or privileged, respectively). Any hardware configuration is acceptable if it can create one internal operating state that cannot be penetrated by any software that a user program can execute.

---

[1] *Supervisory software*, or the Supervisor (also called the Executive or the Monitor) includes that portion of the software that internally manages job flow through the computer, allocates system resources to jobs, controls information flows to and from files, etc.

[2] *User program* (or worker program) is a computer program that performs some task for a user of the system. The Supervisor handles scheduling of the user program into the job stream of the system, the allocation of resources to it, control of its security aspects, etc.

In the *supervisor state,* the machine is able to execute all instructions, including those which affect security controls. In the *user state,* any instruction that initiates an input or output operation (such as a reference to a file), that attempts to modify a register used to isolate users or to protect the Supervisor, or that attempts to suspend or modify security controls must not be executed. Thus, in the user state, a user program will not be able to execute certain instructions and operations that are prohibited to it. Entrance to the supervisor state must be hardware controlled. This frequently is established by providing a facility to detect a special instruction, and creating by hardware means an interrupt signal that returns the computing system to its supervisor state.

If a user program attempts to execute a prohibited instruction, the attempt must be thwarted by immediately suspending the user program and returning control to the Supervisor. Furthermore, if a user program attempts to execute an undefined instruction, this too must be thwarted by immediately suspending execution of the user program and returning control to the Supervisor.

Comment: There are two technical points involved in this recommendation, as well as a delicate question of balancing tight security control against user service. A user program may accidentally attempt to execute a prohibited instruction because the user has made a mistake in his programming; similarly, a sequence of instructions in a user program can inadvertently create a "false instruction," one whose bit-pattern is undefined in the machine; this can give rise to unpredicted results, including bypassing security safeguards. As an aid to the Supervisor in determining which event has occurred, it would be convenient for the hardware to generate unique interrupt signals for each. Conversely, a user program can deliberately create either of these actions as part of a penetration attempt.

From a security point of view, the safe thing is to suspend execution of the user program whenever it behaves suspiciously. However, if the user is attempting to debug a program, he is likely to have errors in his program that will result in his suspension, and consequently interfere with his work. Possibilities for handling this conflict include imposing a time delay on the user before allowing him to continue (one minute, for example), but imposing a shorter delay (10 seconds, for instance) if he has stated that he is in a debug mode and this statement has been verified by the System Security Officer; imposing successively longer delays on the user as the frequency of his infractions increases; notifying the System Security Officer when a user has exceeded a certain number of violations.

Assurance Against Unanticipated Conditions

Since it is virtually impossible to determine in every situation whether a computing system is working as designed, it is obvious that a machine not operating properly is not only of doubtful utility, but also poses a grave risk to the security of the information being handled by it. Thus, it is desirable to incorporate safeguards that protect the system against unanticipated conditions that might arise. As a minimum condition, it is mandatory that the computer produce a known response to all possible instructions (both legal ones specifically in the machine repertoire, and undefined ones), together with all possible combinations of tags or modifiers, whether legal or not.

Comment: This condition is required to prevent the exploitation of undefined instruction bit patterns that might by-pass normal isolation and protection mechanisms.

Summary Comment: There are many other hardware features that are not absolutely essential for implementing security controls, but which can help protect against certain threats or can increase the assurance that controls are working properly and have not been inadvertently by-passed. For example:

Program-readable status switches on the hardware can assure that the program is aware of the hardware configuration in which it resides. This feature can protect against loading of the wrong software, and against some actions of the operator.

Key switches on all important peripheral-device controllers can protect against accidental change in their status or in security safeguards.

Program-readable hardware clocks assist in controlling and maintaining audits and recording ac-

tions by date and time.
An interrupt system can give first priority to hardware errors, malfunctions, and undefined instruction bit patterns.

III. SOFTWARE

The software of a resource-sharing system includes the Supervisor, the language processors (compilers, assemblers, etc.), the program library, and the utility programs (e.g., sort programs, file copying programs, etc.). The design of a computer system must consider all software components of the system, as well as the hardware on which the software will run.

Language Processors and Utility Routines

While a Supervisor of some sort is required on all types of systems enumerated in Fig. 2, the broad range of user software capabilities inherent in systems of Types III and IV implies that a much more complex Supervisor is required for them. With respect to language processors and utility programs, very little can be said that will be of assistance in the design and development of secure resource-sharing systems. In a Type III system (permitting programming via limited languages and certified compilers) the care and thoroughness with which the language processors are examined prior to approval can limit the threat that a user of the system might be able to mount against the classified information it contains. A careful analysis of all language translators, and particularly the assumptions that have been made regarding the execution environment of user programs, is essential on all four types of computing systems.

Assembly languages and the processors for them impose a particularly difficult problem because of the manifold opportunities for the user to create seemingly safe instruction sequences that, in turn, construct executable instruction sequences designed to disrupt service or to by-pass security controls in the operating system. Little more can be said about language processors or utility programs except to require that they be thoroughly tested by the user agency for correct operation and for detection and rejection of incorrect sequences of instructions or other errors. As recommended earlier with respect to hardware, language processors should provide to the maximum extent possible known responses for various error conditions.

Comment: This discussion applies only to the structure of the software components. Additional safeguards against misuse of the software or malfunction by it can be incorporated with appropriate procedural controls. Examination of the software is really an aspect of certification and it is conceivable that, because of the technical expertise implied, examination and testing of software can most efficiently be done by a certifying group.

Supervisor Program

The detailed structure of the Supervisor for a resource-sharing computer system is a function of the hardware configuration and of the type of service provided by the system to its users. Because of the variety of Supervisors and the fact that most resource-sharing systems are delivered by the manufacturer with a Supervisor, it is difficult to specify requirements in detail. In general, however, the software design should be clean, in the sense that it is as modular as possible. There are some aspects to Supervisor design that are sufficiently important to qualify as requirements.

It is recommended that Supervisors designed for a resource-sharing system include the following features:

1. As much of the Supervisor as possible must run in the user state (as opposed to the supervisor state); each part of the Supervisor should have only as much freedom of the machine as it needs to do its job. This should provide the Supervisor more protection than is given to user programs against faulty programming or machine errors. Supervisor functions should be separated into individual, self-contained modules with explicit communication[3] between modules. Each module must be fully described

[3]For example, we would discourage writing a subroutine that on its own initiative reaches into another subroutine for information without the knowledge of the second one. We would insist that some communication require that the first module ask information from the second, and that the exchange take place in an information-exchange area within neither.

with flowcharts to assist in its security analysis.[4]

2. The Supervisor must assure, to the extent technically feasible, that no classified information can remain as program-accessible residue in either primary or secondary storage. This includes all forms of secondary storage (magnetic drums, magnetic discs, magnetic tapes), as well as the primary core store and all registers. One technique is to have the Supervisor erase any segment of primary (core) storage before making that segment available to another program.

Comment: For systems with sufficiently small amounts of secondary storage, the requirement to erase-before-reuse will not be burdensome, but systems with voluminous secondary storage will suffer in terms of efficiency. A possibility for handling the situation (which, however, may be costly in terms of system efficiency) is as follows. If the user program requires some temporary secondary storage, the Supervisor can keep track of how much of the store is assigned, and also of how much information has actually been transferred into secondary storage. Subsequent read-out of such information by the user program will be restricted by the Supervisor to only that volume that has been written. This procedure can be applied to so-called scratch tapes or disc space. It should be noted, however, that tapes, drums, or discs controlled in this fashion must be classified and protected appropriately for the highest level of classification of the information written on them until erased by an acceptable method. Any arrangement that guarantees that a user program cannot read secondary storage beyond material that it wrote originally avoids unnecessary erasure of secondary storage, and also unnecessary computer-erasure of the information. This issue is one which requires attention in future machine designs; features such as bulk-erasure of magnetic discs will be valuable in maintaining system efficiency.

3. The Supervisor must have provision for bringing the computing system into operational status in an orderly manner. There also must be provision for orderly shutdown of the system (including such features as automatic logging out of users and access closure to all files of classified information). Furthermore, it must be possible for system personnel, working at a control console, to pre-empt selected users or to deny access to a given user or terminal (e.g., if an attempt to access the system with improper authorization has been detected).

4. The Supervisor must have a certified capability to control access to files. This point is so critical that it is treated separately below.

Summary Comment: The detailed design of the Supervisor and the protective safeguards that it contains and that are afforded it are vital to adequate security control. Since commercially designed Supervisors and operating systems have not included security control, it is to be expected that the average commercial software will not provide the standards, conventions, and capabilities required. A number of potential design guidelines are suggested here.

The Multics time-sharing software[5] utilizes the concept of concentric circles of protection. The most sensitive part of the Supervisor (sensitive in the sense that penetration of it will open the machine completely to the user) is conceptually at the innermost circle. Surrounding it in successive rings are decreasingly sensitive parts of the Supervisor. A user program seeking access to some portion of the Supervisor must specifically thread its way through the concentric rings until it reaches the desired portion. Thus, there is no direct route from a user program to, for example, the file-access control mechanism.

In the case where the Supervisor is responsible for data segregation, it must check the authority of terminals that originate traffic, must properly label (internally) all traffic, must label all tasks whose execution is required in order to service a user request, must keep track of all tasks and of the programs that execute them, must validate the security markings (including security flags) on all tasks and control access to files on the basis of the markings, and must validate (by reference to internal tables or files) the au-

[4]For an example of this type of design and the level of documentation required, see the software maintenance documentation for the GE 625/35 GECOS III time-sharing system.

[5]V. A. Vyssotsky, F. J. Corbato, and R. M. Graham, "Structure of the Multics Supervisor," *AFIPS Conference Proceedings*, Vol. 27, Part 1, Spartan Books, Washington, D.C., 1965, pp. 203-212; also R. M. Graham, "Protection in an Information Processing Utility," *Communications of the ACM*, Vol. 11, No. 5, May 1968, pp. 365-369.

thority of a remote location to receive output information with a given security marking or flag.

The system programs that collectively form the Supervisor must not be allowed to execute with complete freedom of the machine. Ideally, such system programs should execute only in the system's user state; otherwise, these programs should execute with as many restrictions as possible. Only the minimum number of system programs should be allowed to execute without any restriction. Relaxation of this philosophy in order to facilitate execution of a system program can lead to a serious weakness in security.

An essential aspect of access control is the security flag that identifies the classification level of the program, the data, the terminal, and the user. The basic philosophy of a program executing in the user state is that it is able to process anything that it has available within the region of core memory (or logical address space) assigned to it. Thus, satisfactory security control depends upon careful monitoring and control of what a user program brings within its memory region (physical or logical). Specifically, it must not be allowed to bring security flags into its region. If an unusual program has the privilege of writing outside its core region, it can in principle modify security flags. Obviously, such programs must be carefully designed and must be faultless.

Since system programs are very sensitive with respect to security controls, they must be carefully debugged before becoming resident in the permanent program library. Those of particularly high sensitivity, such as routines for controlling access to classified files, must be given extraordinary attention during the debugging phase.

It is desirable that system programs which have unusually broad capabilities (such as being able to access all permanent files in secondary storage or in temporary working stores) be programmed so as to print console messages notifying the System Operators of the specific privileges being extended; before proceeding to implement such privileges, the system should require explicit permission. All such events should be logged automatically, together with the operator's response and, when deemed necessary, the concurrence of the System Security Officer. This restriction is a double check to prevent unauthorized execution of broad-capability programs with malicious intent.

IV. ACCESS CONTROL THROUGHOUT THE SYSTEM

In a resource-sharing computer system, access to the system itself and access to the information (files and programs) contained in the system must be separately controlled. If the resource-sharing system is a multiprogrammed computer operating with only local (as opposed to remote) access, operations personnel can visually identify an individual before granting him access to the system. Furthermore, the operations people can perform whatever verification procedure is necessary before releasing particular files or programs to that user. Alternatively, if such user information as authentication words or access protocols must be protected when in punchcard form, an arrangement can be made to have the card deck read under the visual surveillance of its owner, and immediately returned to him. For remote batch and resource-sharing computer systems, such functions must be performed by security-controlling mechanisms in the system software and hardware.

User Access

In a terminal-oriented system, a user must announce himself to the system through a log-on procedure that requires standard identification and accounting information, and a specific user authentication step so that the computer system can verify the identity of the individual at the terminal. For systems that have point-to-point permanent and protected communication links, physical control of access to a terminal may be used in lieu of authentication. In this case, responsibility for authentication is transferred to the administrative jurisdiction which has cognizance over the terminal. For systems that utilize dial-up communication links, or in which physical access control is undesirable, a password scheme or its equivalent must be used to provide authentication.

Authentication words or techniques must be classified and protected by the user in accordance with the highest level of information to which it permits him access. Authentication words or techniques must be obtained from an approved source, or, alternatively, must be generated and distributed under the cognizance of the System Security Officer by approved techniques. Specifically, a user cannot gener-

ate his own passwords. Depending on the sensitivity of information or operating conditions (circuit noise, interruptions, etc.) contained within a system, a user may be required to reauthenticate himself from time to time during a single terminal session. Authentication words must be changed as frequently as prescribed by the approved issuing source.

Provided that techniques approved by the appropriate cognizant agency are used, the resource-sharing system can itself be utilized to generate authentication words, provided the output is available only at a designated terminal and that the procedure is carried out under the cognizance of the System Security Officer.

The Supervisor software must be so constructed that user identification and authentication word lists can be maintained as part of the normal operation of the system from the terminal designated for the System Security Officer who has sole responsibility for such lists.

Information Access

The fact that a user is granted access to a system does not imply authorization to access classified files of data and programs contained in that system. For example, he may be authorized to perform only on-line computation, but not on-line file processing. Before a user is given access to a classified file, the user's clearance level, need-to-know, and access privileges must be checked against the access restrictions of that file. If information from this file is to be delivered to the user's terminal or to a terminal designated by him, the status of the designated terminal must also be verified. To do this, the computer system must have an internal catalog of user clearance levels and access privileges, as well as a catalog of the characteristics of all terminals connected to the system. Each file must be marked with any clearance, need-to-know, or other restrictions on its use. Finally, there must be an explicit and separate capability to update such an internal catalog. If the responsibility for maintaining this catalog is divided among several people, each must be restricted to only that part of it for which he is responsible.

Comment: The Appendix describes a system for implementing a file-access control mechanism. It also discusses a scheme whereby the System Security Officer can describe to the computing system that part of the total security structure with which his system must deal, as well as a means for inserting security parameters into the system.

In addition to the security reasons for controlling access to files, it is necessary also to control access so that unauthorized changes cannot be made, particularly if the file management responsibility is assigned exclusively to some individual or group—e.g., the Office of Primary Responsibility. For example, even though a given user might qualify for access to a particular file in terms of proper clearance and need-to-know, he might be granted access to read it but denied the right to change the file because this privilege is reserved to a designated file manager. Thus, in part, security control and file integrity overlap. Both features are essential, and common software can conveniently accommodate both.[6]

Denial of Access

A user must not be able to acquire information about the security controls or the files when access is denied him for any reason. Assuming inadvertence on the part of the user, the system should assist him in identifying his mistakes or procedural errors. However, the system logs should record all unsuccessful attempts to access classified files.

Comment: The point of this prohibition is to guard against acquiring incidental information by browsing. Thus, an improper access request must result in some innocuous reply, such as, "File not found." However, the restriction that the system not reveal the existence of a file creates a potentially awkward situation because the user might inadvertently create a file (perhaps public and unclassified) with the same name as one whose existence is unknown to him. Since different files of the same name are unacceptable in a system, the system must (1) inform the user that his proposed name is unacceptable (without giving a reason), (2) prefix all file names with a user-unique code to guarantee dissimilarity of names, or (3) use some pseudo-random process to automatically generate file names.

---

[6]For example, see R. C. Daley and P. G. Neumann, "A General-Purpose File System for Secondary Storage," *AFIPS Conference Proceedings*, Vol. 27, Part 1, Spartan Books, Washington, D.C., 1965, pp. 213-229.

Maintenance Access

Because systems are vulnerable to security threats posed by operations and maintenance personnel, it is strongly recommended that for systems handling extremely sensitive information all software and hardware maintenance be performed as a joint action of two or more persons. In particular, on-line debugging of the Supervisor software is expressly prohibited except when (1) all on-line storage devices containing classified files not needed in the performance of the maintenance are physically or electrically disconnected, and (2) only fully-cleared maintenance personnel have access to the system.

In order to maintain good security control, it is recommended that modification of installed system software currently in operation be done from specifically designated terminals; that system software maintenance personnel be assigned unique access privileges, including authentication words to permit them access to test files, system functions, etc.; and that all actions from such specially privileged consoles be under the continuous, positive control of a responsible individual who maintains a written log of the console use, including positive identification of the individuals using it. Such special hand-maintained logs should be in addition to the automatic logging performed by the system.

File Classification Determination

The system can and should be designed to assist the user in determining the appropriate classification and applicable caveats for each new file. In many cases, this can be determined algorithmically by the computer through a consideration of the classifications and caveats of all files referenced, programs utilized to create the files, and inputs.[7] In other cases, it can only be determined by the user. Whenever a user is notified by the system that, based on internal information, it has assigned a tentative classification status for a newly created file, he must indicate that he has verified and accepts this status or desires to change it. If a user chooses to change the classification, either raising or lowering it, or to add or remove caveats, the system should record the transaction in its log and specially note it for review by the System Security Officer. In either case, the user's action must be recorded in the system log. If the classification has been lowered or caveats have been removed, the file must not be released to other users before the System Security Officer has verified that the new status is correct. In some operational situations, it may be prudent to limit downgrading authorization to only those users who are entitled to write into a file.

When a new file is created by combining information from existing files and adding interpretations of the combined results, it is conceivable that a purely algorithmically determined maximum classification and caveats may exceed the user's access privileges. In such a case, the access control mechanism must be designed to withhold the information from the user and to bring the situation to the attention of the System Security Officer.

Comment: The reason for requiring the user to confirm or modify the computer-determined status, rather than permitting the user to specify his own, is that he may not be aware of the totality of all file classifications and caveats that he has referenced; thus, he would be unaware of the classification status of the composite information. Classification of a large collection of classified documentary information always requires extensive manual analysis and evaluation; a corresponding action on large computer files would be unreasonable.

Input/Output Limitation

It is recommended that software traps be incorporated to detect any input or output information identified by a security flag that exceeds that authorized for either the user, his terminal, or any file specified in his job. Such a condition must immediately suspend service to the terminal, notify the System Security Officer, and record the event in the system log.

Comment: This implies that all input/output operations are buffered through a storage area assigned to the Supervisor on the way to or from a user program. For example, information from a terminal must be moved into buffered storage, its security flag detected and compared with the user privileges, and then it must be moved again into the user program area.

Typically, the Supervisor is designed to receive

[7] See the Appendix for one such scheme.

Clearance	Classification		
	Input	Job	Output
User	≥	≥*	≥
I/O Device	≥	Independent	≥

*Except for certified execute-only programs.

Figure 4

remote input information only from the terminal that originates the job and, correspondingly, to output information only at that terminal. If operational requirements dictate otherwise, the Supervisor must be so designed that it can identify and authenticate terminals and users other than the originating one and with which information will be exchanged.

Job Security Interaction

As a user's job actually runs in the computer, it will carry a security flag that initially is determined from the security flags of the user and of the terminal from which he works unless the user specifically designates otherwise at the beginning of the job. In either case, as the job unfolds, the security flag may have to be modified automatically by the system to reflect the security flags of files of information or files of other programs that are used. The job flag need not be limited by the terminal flag. For example, an individual cleared for Top Secret might run an entirely Top Secret job through a Secret terminal if there is to be no Top Secret input or output through the terminal; the output, for example, might be directed to a Top Secret printer. A situation such as this might be common for remotely initiated batch operations, and no deception is indicated since the user is cleared for the job even though his terminal is not. The basic point is that the security flag of the user is the absolute limit on his access privileges, unless the program in question has been certified to have access to higher security flags but to produce information that does not exceed the flag of the user.

The access control limitation just outlined can be represented as shown in Fig. 4. It is read: user (device) flag should be greater than or equal to (>) the input (job, output) flag.

It may prove too difficult in a specific case to certify that a program can access highly classified information but produce results of a lower level. If so, it is strongly recommended that a user's job never be allowed to access information—either data or programs—whose security flag exceeds that of the user. Since parts of the Supervisor will run in the user state as a user program, access in such a case to accounting and control files must be excluded from the restriction.

In principle, the following items can each carry a security flag: user, terminal, job program, job data, file data, input, and output. The question of which jobs a user can run in each possible circumstance can become very complex. Unfortunately, the Supervisor will have to determine user privileges algorithmically; it cannot exert judgment. Thus, the issue must be examined carefully in each operational environment, with appropriate rules formulated to match user needs and security restrictions of the installation.

Comment: A program might be intrinsically classified because it implements classified algorithms, and, thus, its classification establishes a lower bound when it runs as part of a job. On the other hand, a classified program might access data more highly classified, and, hence, the job classification can exceed that of the program that is executing.

Multilevel Utilization

It is possible to demonstrate that many resource-sharing computer systems may be safe from direct user attacks from terminals by proving that a particular hardware/software component is effective in blocking attacks of various kinds. However, there is the recurring question of the risk of inadvertent disclosure of classified information through software, hardware, or a combination of failures; in such a case, it would be necessary to prove that a single failure or a combination of failures cannot occur. Since a complete proof of protection is not within the present state of the art, particularly for existing computer systems, it is recommended that the system designer estimate the probability of occurrence of a single failure or the combination of failures that could result in a disclosure of classified information. Based on this information, the Responsible Authority can determine whether the risk probability is acceptable or not. If the decision is that the risk is too great, a *segregated mode of operation* should be used, and the system certification made accordingly.

A system functioning in a segregated mode requires that all users are cleared to a specified level, all terminals are physically protected to that level, and all communication lines are secure to that level. If, within any level of classification, special caveat information is introduced, a new determination must be made as to whether the risk and consequences of exposure of the special caveat information to cleared but not authorized persons operating

User Clearance	Current Classification of System					
	Special Access "A"	Special Access "B"	Top Secret	Secret	Confidential	Unclassified
Special Category "A"	●		●*	●*	●*	●
Special Category "B"		●	●*	●*	●*	●
Top Secret			●	●	●	●
Secret				●	●	●
Confidential					●	●
Uncleared						●

● —Access authorized.

●* —Access may or may not be authorized, depending on the relation of the Special Category to the given national classification.

Figure 5

within the system warrants segregated operation of the entire system at the special caveat level. If the classification level at which the system is certified to function hierarchically subsumes other levels of classification, then authorized users of the system may execute programs of such lower levels of classification. However, if the scheduled mode for the system establishes a level of classification which is mutually exclusive of other levels, the users are restricted to programs classified at the current mode of the system. Fig. 5 illustrates these relations.

The concept of segregated operational modes requires that users of various clearance levels be scheduled separately. In addition, special controls are needed to assure that highly classified or caveated material does not become accessible when a lower-level classification or differently caveated mode begins operation. The precise procedures and mechanisms necessary to change the operational status of a system must be tailored to the precise hardware/-software configuration. The following steps are representative of the procedures necessary to maintain segregation when system status changes.

(a) When file information is permanently resident in the system (e.g., on disc files or mass storage devices), the information must be protected by disconnecting such devices (by certified electronic switching, unplugging cables, or manual operation of switches) if the classification or special-access categories of the file information are such that the file must not become accessible to unauthorized users under any circumstances.

(b) Before a file device is made available to users with more restricted access privileges than those who have been using it, it must be sanitized (and checked) by approved procedures of any classified information more highly classified or restricted in access than appropriate to the new mode of operation.

(c) Each user must be notified of any change in the operational status of the system, whether scheduled or not. This notification should be transmitted prior to the change to all active terminals that will be able to access the system in its new mode of operation. However, a terminal not authorized to access the system in the new mode should not be given any information about the specific classification status of the new mode. A change in the mode of operation must be accomplished by recessing or logging off, as appropriate, all active users and forcing a new log-on procedure, including authentication, for the new level.
A change in the operational status of the system will obviously inconvenience users. While some will be required to terminate their work completely, all will be required to momentarily suspend operation until the change in status and the new log-on have been accomplished. To the maximum extent possible, the procedures for changing the status of the machine should be designed with user convenience in mind.

(d) Since the operational clearance status of the system can change in a segregated style of operation, any user who is granted access to the system must be informed by the system of its current status.

(e) When initiating a new operational mode, terminals in work areas not cleared to receive the information at the forthcoming level of operation must be disconnected from communication links with the computer (by certified electronic switching, unplugging, or manual operation of switches).

(f) When initiating a new operational mode, any special software relevant to the new mode must replace that of the previous mode.

(g) In the event of a failure in the Supervisor software or in the hardware resulting in an operational malfunction, the system must be restarted at the appropriate clearance level by an approved restart procedure as a part of returning it to operational status in the same mode.[8] Depending upon the nature of the malfunction, it may be necessary to verify the security flags of on-line data files in order to assure that the malfunction did not affect them.

The recommendations above indicate in a general way what is required; additional issues, such as the following, must be considered.

(a) Indicator lights visible to the operator may be

[8] See Part D.

needed so that the status of on-line file media is readily discernible.

(b) The disabling of read heads of magnetic disc devices may be required.

(c) Appropriate key locks may be needed so that an operator is assured that certain actions have been taken; the action of these locks must be electrically reported.

(d) Checklists are helpful to assure that system operating personnel methodically verify each step of the process.

(e) Storage of such classified material as punchcards, printed paper, magnetic tapes, etc., must be provided.

(f) Printers or punchcard equipment must be sanitized by running out blank paper or blank cards; ribbons must be changed or protected.

(g) Positive control procedures should be used to assure that magnetic tapes or magnetic disc packs containing classified information of one level of classification or special category are not accidentally used at some other inappropriate level.

(h) There must be detailed instructions to the system operating personnel for each mode, relative to such things as console actions, on-line file status, memory-clear procedures, mode shut down, mode initiation, message insertion via the console typewriter, etc.

(i) There must be continuous surveillance of the operations area by fully cleared personnel.

It is not possible to consider explicitly all the changes that must take place in a computer system for a change in operational clearance level. In general, the recommendations given parallel practices common in existing security doctrine. At a particular installation, the System Security Officer will be aware of the levels of classification and special access categories in his system, and must be able to formulate the detailed procedures for shifting the operational mode of the system from one to another.

V. COMMUNICATION LINES

Any communication line that passes classified information between a terminal and the central computer facility or between computer systems must be protected in accordance with Government-approved communication security methods. They may include provision of approved secure cable between the terminal and the central location, or of approved cryptographic equipment. Intelligent deception of the link (i.e., spoofing) must not be possible.

Emergency Communication Arrangements

There may be an operational requirement to maintain continuity of service to a remote user in spite of communication circuit failure. If so, there must be emergency provisions and procedures for establishing alternate channels to remote locations, and such actions must be accomplished by properly cleared and authorized individuals, in accordance with established operating procedures for secure communications.

High-Risk Areas

If the resource-sharing computer system operates in an environment wherein there is a reasonable probability of one or more terminals being captured, then it is essential to employ the technique of cryptographic isolation (i.e., use of a unique key for each terminal). In the event of capture, this confines the operational and information loss to the captured terminal, and prevents the captor from intruding on other communication links in the system and intercepting classified information intended for other terminals.

VI. TERMINALS

Terminal Protection

Any terminal through which a user can gain access to classified information in the central computing facility must be physically protected in accordance with the highest classification of information processed through the terminal. Furthermore, if protection requirements are specified for any cryptographic equipment collocated with the terminal, the physical protection must be in accordance with the protection requirements specified for that cryptographic equipment. In addition, if the system is

closed, the protection must be consistent with that specified for the overall system.

To guard against the covert emplacement of illegal intelligence sensors or recorders, terminal maintenance personnel must be cleared for the highest level of classified information handled at the terminal, or the terminal maintenance must be performed under surveillance of an appropriately cleared and technically knowledgeable person.

Terminal Identification

Because present security doctrine depends heavily upon identification, it is necessary that a remote-access, resource-sharing system require positive identification of each terminal with which it communicates, and that the system be able to interrogate a terminal for its identification at any time.

Comment: Terminal identification is particularly important when a computing system is being brought into operational status initially, or when it is being recertified as a secure configuration. This recommendation also applies to all remote equipment, such as other computers.

If remote terminals are connected into the central processor via a dial-up connection rather than permanent hard wire, this requirement for terminal identification may require a separate authentication method despite the use of cryptographic equipment on the circuit. This recommendation will also apply to the situation in which a user at a terminal connected to one system wishes to access a second system. In some systems it may be permissible for the user to authenticate himself to his own system, which then passes the authentication to the second system via their mutually authenticated and protected communication link. In other cases, a unique arrangement may be necessary to enter the second system.

VII. CERTIFICATION

Certifying that a resource-sharing computer system is secure represents a very difficult issue. It involves an examination of the safeguards—hardware, software, procedural, administrative—that have been provided, and, ideally, a quantitative estimate of the probability of inadvertent disclosure of classified information. It is almost impossible to identify and protect against all possible failure modes of a system.

Design certification is the process of measuring, testing, and evaluating the probable effectiveness under operating conditions of the security control features of a stable system—i.e., one whose software and hardware have been completed. In order to make the measuring process meaningful, the security protection designed into a system must be quantified to the maximum extent possible. It is strongly recommended that design certification be performed by a group other than that responsible for the design, construction, or maintenance of an operational system. A suggested procedure is given below:

(a) Identify all hardware elements (such as registers, base address registers, counters, etc.) that provide or are depended upon for direct operation of a security control function. Identify all system software features, barriers, and components that have a security control function. For each of these determine:
   (1) Its logic;
   (2) Hardware failures that will cause incorrect operation and any inherent checks that are intended to detect such failures—e.g., a parity check on register-to-register transfer;
   (3) The probability of failure of the hardware upon which a security control depends;
   (4) Possible software checks on the consistency of its operations and the accuracy of parameters, addresses, etc., used by the function;
   (5) Combinations of data (parameters, tables, etc.) that will result in incorrect operation;
   (6) Its dependence on other functions for its own operation;
   (7) The probable effect of its failure;
   (8) Specific tests—either software or electronic—that can be made to determine if the function really works as specified.

(b) Based on the determination of these factors and test results, make an overall estimate of the probability of failure of the total function.

(c) Based on the probability of failure of each security function, estimate the overall probability of a system security failure that would result in a compromise of classified information or an illegal entry into the system.

The matter of overall equipment configuration becomes especially important in large systems containing many computers, either collocated or geographically distributed. The overall hardware configuration must be examined in order to establish the consequences to the security controls of a total or partial loss of a major component in the system. For example, if the controller for a group of magnetic discs were to fail, it is necessary to determine whether a crucial segment of the software would be made unavailable for security control. Whenever possible, security controls should be designed so that failure of a portion of the system does not invalidate or weaken the controls in the balance of the system remaining operational. Conversely, the design should permit rapid and simple physical disconnection of an inoperative portion of the system. Following are some other points that should be considered.

(a) If the failed component (such as a magnetic drum, a section of core, or a second computer) contains information required for security control and not available elsewhere in the system, the entire system must shut down or operate in a degraded mode. The decision should be made jointly by the System Security Officer and the System Administrator.
(b) The loss of some components may so seriously affect the operational performance and accuracy of the remainder of the system that it should be shut down for that reason, even though significant security controls continue to function.
(c) Loss of communciation between elements of the system may force it to be shut down if data critical to security control in the system cannot be transferred.
(d) If the Supervisor software is designed to monitor the operating status of each remote station before sending information to it, the loss of a remote station is not a security threat, although such incidents must be reported to the System Security Officer.
(e) Loss of an operator console may require that the associated computer must be shut off if it cannot be properly controlled, or if alternate locations for operator control are not available.

At the time of installation certification, the administrative and procedural environment in which the system is to function must be examined to verify that it supports the controls present in the hardware/software complex, and that it provides the additional controls on the people, paper, magnetic tapes, etc., of the system. Also at installation certification, the communications arrangement must be verified to be secure, the level of spurious emanations must be demonstrated to be acceptable, physical protection must be shown to be adequate, and all controls over remote equipment (physical, personnel, emanation) must be verified.

Complete certification should be performed before changing a closed system into an open system even though it may be operated in a segregated mode, as previously described, when processing highly sensitive information. After a system has been certified, all changes to the system must be similarly examined before being incorporated. Such an examination is required whether the changes originate with the user-agency or with either the hardware or software vendors.

After the general reliability of a system has been established by operating successfully for a reasonable length of time, a limited recertification process should be performed at appropriate intervals, consisting only of tests and inspections intended to reveal changes surreptitiously made in the system, or to detect inadvertent changes made in the system during maintenance, or to validate the continuing performance of system security controls.

Audit Trails

The audit-trail technique can be used to verify that a system is operating correctly and, more importantly, that it is being used properly. For purposes of monitoring security controls, it is recommended that the system contain software that automatically records (with date and time) at least the following:

(a) All user log-ons and log-offs, including each user's name, identification, and terminal;
(b) All maintenance log-ons and log-offs for whatever purpose, including the names of maintenance personnel, the nature of the maintenance, and any files accessed;
(c) All operator-initiated functions, including his name and the function (from the point of view of the logs, the operator should be treated as a user);
(d) Each attempt by a user or his program to access files or programs for which he is not authorized, including his name, terminal, and an identification of his program;
(e) All program-abort incidents, including the name of the program, the user, terminal, and time of abort;
(f) Any special usage of the system—e.g., generation of passwords, changing of the classification, or modifying security parameters; a record of the type of transaction, including the authority or person under whose cognizance the usage is conducted, and the terminal used;
(g) Groups of output operations that the system performs at the request of a user, including those which he directs to be sent to a terminal other than the one from which the request was made; including identification of the file accessed and a measure of the amount of information read out from the file, and the requesting and receiving terminals. Similar information should be logged for all input operations that create or destroy files or instructions, or that change file classifications or security parameters.

To the extent deemed necessary by the System Security Officer, the log records must contain sufficient detail to permit reconstruction of events that indicate an unsuccessful attempt to penetrate the system or that clearly resulted in a compromise of information or a security violation. For example, repeated unsuccessful attempts to gain access to the system software or to a file should be promptly reported by the Supervisor software in order to alert system operations personnel and, if necessary, the System Security Officer. The audit trails should enable security investigation personnel to identify the terminal involved, the user, the target file or program, and the system reaction. In general, the log should be complete enough to permit the System Security Officer to monitor system performance on a real-time or periodic basis, as needed. The data collected by the system log can also be aggregated at intervals to provide performance statistics that indicate the efficacy of existing security safeguards, and to develop new or improved procedures and controls.

Comment: If a system contains unusually sensitive information or must operate in an unusually hostile environment, more extensive automatic logging of system activity may be desirable. Furthermore, in some cases the presence of special machine instructions whose execution might modify or by-pass security controls, or the existence of an unusual configuration, etc., might require logging of additional activity—e.g., any use of a diagnostic instruction that can lead to subsequent errors because of change-of-mode in the machine.

Supplementary manual logs kept by the operators to record such events as the following may be useful.

(a) Machine faults, failures of internal checks, power losses, environmental malfunctions;
(b) Restarts of the system, including details of the loading of system software and by whom, checking or verification of files, manual operations taken, etc.;
(c) All changes to the Supervisor, the program library, or any system files made by way of the operator console;
(d) Each running of unusually privileged system programs and by whom;
(e) Each instance of hardware or software maintenance, by whom, and for what purpose.

Comment: A system will also log much information for purposes of accounting for resources assigned to users, for scheduling events and users in the system, for allocating charges to users and to accounts, etc. Such information may also be useful for monitoring the security controls. Since a large volume of information will be available through the various logs, it is clear that special data reduction programs, event-correlation programs, and data-summary programs will be required by the System Security Officer.

Self Surveillance

As a means of verifying the continued correct operation of the security safeguards in a resource-sharing computing system, a system self-inspection and testing program must be inserted into the system with the status of a user program. The function of this program is to verify that the hardware and software safeguards are operative. At a minimum, the testing program should attempt to violate security controls, and should verify that the correct response was received in all cases. The security testing program must communicate with the computer system by directing its information through a turnaround channel (i.e., one that leaves the central processor proper, traverses a channel controller, turns around, and re-enters) in order to verify the integrity of the channel controllers as well.

If the test program succeeds in any attempt to violate either a hardware or software safeguard, the system shall immediately enter a unique (degraded) operating mode, in which it withholds all information from the user community until the situation has been assessed and appropriate action taken (see Part B, pp. 14-25).

Security Violation and Auto-Testing

If a user program violates any security controls while running operationally (i.e., not during debugging), the program must be immediately suspended and the System Security Officer notified. Appropriate remedial action must be taken and verified before the program is returned to operational status.

If the violation occurs during on-line debugging of application programs, and the program has not accessed files of sensitive information, it is sufficient to notify the user, alert the System Security Officer, and record the event in the system log, while allowing the program to continue after the user acknowledges the event and responds with any appropriate remedial action. In any such conflict between a user program and security controls, but especially in the case of an open system, it may be advisable to interrupt all system operations at the first feasible opportunity and run a security testing program to verify correct functioning of all security controls.

Comment: This situation is a delicate one in that it reflects a compromise between user convenience and security of information. A complete abort could leave the user in an awkward position from which it may be difficult to restart his program or recover any completed work. Similarly, it is an inconvenience to other users to be interrupted even briefly in order to recertify the system. Obviously, the seriousness of the violation and the potential security risk are matters that the System Security Officer is responsible for judging.

VIII. OPEN ENVIRONMENT CONSIDERATIONS

As stated earlier, it is simpler to create a secure system in a closed environment than an open one, largely because of inadequacies in the present state of technology. The foregoing recommendations present techniques and methods relevant to protecting information in an open environment, but which may not assure security in such a situation. A few comments are in order on the practicability of reducing the degree of openness as a means of coping with the security problem. The system can be closed to uncleared users when classified information is resident; this is a simple and possible course of action. However, it may be impractical because the workload and population of users in many installations will be such that a single computer system is required to economically serve both cleared and uncleared users.

On the other hand, it might also be true that the volume of classified and the volume of unclassified work are such that an economic solution might be a separate machine for each part of the workload. A modification of this approach is to schedule a system to operate alternately in uncleared and classified modes, with appropriate operational procedures to sanitize the system and to certify it between modes. All information within the system might be rendered unclassified, which implies that internal encryption is used. Finally, it might be possible to find special configurations of hardware that could be certified secure even in an open environment—e.g., duplex-redundant processors and input/output controls with management of the system and of the security controls vested completely in a third and independent machine. With respect to internal encryption, it should be noted that the principal threat countered is recovery of information. The threats of system denial or intelligent deception must be coun-

tered by other controls. A possible benefit of internal encryption may be that it reduces the scope of system certification to more manageable proportions. A possible drawback is the possibility of a malfunction in the encryption device permanently "freezing" the information in an encrypted, impenetrable state.

Internal encryption could be applied not only to the primary magnetic core storage, but also to secondary file storage. All programs and all data resident in core storage could be in encrypted form and decrypted only as they pass from storage to the processing unit for execution. As information is returned from the processing unit to storage, it would be reencrypted. Incorporation of this technique into a system would protect against unauthorized access to data resident in primary storage. In addition, information in secondary storage could be protected by an encrypting mechanism connected directly to the encrypted primary storage in such a way that information could be transferred from primary to secondary storage without an intermediate plain-text stage occurring. The purpose of securing secondary storage in this fashion is to protect against physical access to storage devices. On the other hand, encryption of secondary storage greatly complicates the file management problem.

IX. RESEARCH NEEDED

In addition to continuing research into internal encryption devices, as mentioned above, other research requirements include special hardware configurations to maintain absolute segregation between uncleared and other users, special software for such configurations, automatic recertification procedures to be used by the system itself between configuration changes, comprehensive automatic monitors (hardware and software) for security controls, more reliable self-checking hardware architectures, methodology for identifying failure modes and accurate prediction of failure probabilities, and new machine architectures whose security controls minimally affect the efficiency or cost of the system.

X. OVERALL SYSTEM PROBLEMS

Security control in a computer system, especially a resource-sharing one, is a system-design problem, and solutions to it must be based on a system point of view. A number of problems covered in the preceding discussions are brought together here briefly because of their importance to the system as a whole.

Redundancy

Given the present state of computer hardware and software technology, we can expect that even the best designed systems will have relatively frequent malfunctions. While system designers can be very ingenious in attempting to arrange safeguards so that malfunctions do not result in serious consequences, nonetheless, given the present lack of experience with computer systems that contain security safeguards, it is strongly recommended that redundancy be incorporated throughout the system safeguards. Redundancy might take such forms as duplicate software residing in different parts of the memory; software checks that verify hardware checks, and vice versa; self-checking hardware arrangements; error-detecting or error-correcting information representations; duplication of procedural checks; error-correcting internal catalogs and security flags; or audit processes that monitor the performance of both software and hardware functions.

A particular point to note is that the absence of a parity check in the memory or in information transfers can permit errors which perturb, disable, or mislead security controls. In the absence of parity checks throughout the machine configuration, equivalent error-detecting procedures must be incorporated into the software.

Certification

As system designers and system operators acquire insight into the behavior of resource-sharing configurations, new and revised certification tests will have to be developed to check one or another aspect of system behavior. Certification is a continuing process. It is the experience of designers of multiaccess, resource-sharing systems that even with the best and most ingenious designs, users of a system find ways of chaining together actions that were not foreseen by the designers and which, in many cases, lead to undesirable or disastrous consequences. Therefore, in order to establish confidence in the

security controls, the certification procedure must include a phase that deliberately attempts to penetrate our best designs, and that is conducted by technically competent individuals not part of the design group or of the operating agency, and not administratively responsible to either.

Debugging and Testing

During debugging of a new program or testing of a program with new data, the likelihood of an error is much greater. It is inappropriate to levy security violations against a user for security errors occurring during a debugging phase; but it is dangerous to risk having an agent conceal his activities as debugging errors. Possibilities for dealing with the problem include: requiring the user to state his intention to be in a debugging mode and to have this fact noted (and possibly authenticated to the system) by the System Security Officer; requiring all debugging to operate through a certified interpreter; requiring all debugging of programs to operate on dummy and unclassified data; reflecting all errors and violations of security control back to the user with an enforced delay before he can resume work.

System Component Isolation

Each system component—individual user, operator, maintenance person, etc.—must be isolated from all other components of the system to the maximum practicable degree, except as needed to do its job. Strict adherence to the principle of isolation is necessary in order to avoid undesirable or unpredictable side effects in case of failure or malfunction of a particular item in the system.

Fault Detection

System design must be such that faults—malfunctions of either the equipment or the Supervisor software—are readily detectable. The damage resulting from a fault depends upon the importance of the faulting element to the security control structure and the length of time that the fault goes undetected and unremedied. Intermittent faults may go undetected because of error-correcting procedures in the system, or because the system may automatically repeat a faulting operation. Faults in the Supervisor tend to be subtle and not immediately detectable; as a general principle, it is desirable to design the Supervisor so that faults result in gross misbehavior, thus facilitating detection. However, in practice, this principle is difficult to apply because of the complexity of the Supervisor software and because only after-the-fact operational experience will indicate the general manner in which a given software design faults.

Cross-checking

Where possible, security controls should be designed to cross-check each other; e.g., operator input actions should be recorded automatically in the log, which is transmitted to the System Security Officer, thus minimizing the opportunity for an operator to take any undetected hostile action. Also, to the maximum extent possible, checks between security controls should cross system components; e.g., manual actions should be checked by equipment records, software checks of hardware should not depend on the hardware being checked.

Gradation

In principle, the number, type, and depth of security controls in a system should depend on the sensitivity of the information in the system, on the class of users being served, on the geographical distribution of the system, on the nature of the service that the system provides its users, and on the operational situation that the system supports. In several places, it has been suggested that detailed decisions must be made by the System Security Officer, by the user-agency, or through a consideration of the sensitivity of the information and classification levels involved. The cost of providing security controls may turn out to be substantially independent of the factors noted above, or it may strongly depend on them. Thus, positive statements about gradation of security controls await the design, implementation, and operational experience with a few such systems. Examples of features whose presence, frequency of operation, completeness of checking, etc., might be subject to gradation are:

- The variety and amount of information recorded in the system logs for audit purposes;

- The manner in which user debugging and testing of programs is handled;
- The periodicity and completeness of the internal self-testing program;
- The frequency with which users must authenticate themselves;
- The amount of redundancy in the security controls;
- The number of events reported to the System Security Officer for his attention;
- The depth of operational control exerted by the System Security Officer;
- The frequency of recertification procedures;
- The internal events that are reported as security violations;
- The frequency with which authentication words must be changed.

User Convenience

At several places it has been indicated that the system must be designed to aid the user or to behave in a way helpful and convenient to him. This point must not be taken lightly. User convenience is an important aspect of achieving security control because it determines whether or not users tend to find ways to get around, ignore, or subvert controls.

Centralization of Vulnerability

Care must be exercised not to create inadvertently a system weakness by centralizing too much responsibility in one individual. For example, the System Security Officer oversees all the protective features of the system, as well as controlling its operational security status. Thus, he has broad and critical powers, and becomes a potential target for subversion. Appropriate administrative and procedural safeguards, plus division of responsibility and power in the System Security Office, will be required to offset such a threat.

Positive Alarms

A computer system can malfunction in ways that are not readily noticeable to its operators; thus, it is conceivable that security controls can also malfunction or fail without noticeable evidence. All security controls must be implemented in such a way that failure or malfunction is positively and unambiguously transmitted, preferably in a redundant fashion, to the System Security Officer.

Part D
MANAGEMENT AND ADMINISTRATIVE CONTROL

In addition to overall policy guidance and to technical methods, there must be an effective set of management and administrative controls and procedures governing the flow of information to and from the computer system and over the movement and actions within the system environment of people and movable components (e.g., demountable magnetic tapes and discs, print-outs). An essential aspect of effective control is standardization of activities and the need for standards throughout the system. Their presence will make attempts to subvert the system much more visible and detectable.

Comment: The importance of standards is a subtle philosophical point. They are effective in many ways: with rigidly prescribed procedures, operators will be inhibited from taking shortcuts that can result in leakage; "game players" who wish to subvert the system to their own ends will find it much more difficult in a highly standardized environment; records of system performance and human activities will be available so that the system can be tuned for improved service; etc.

The discussion below presents typical procedures that are required, and suggests some details of each. For each, it is necessary to provide forms for recording, initiating, and controlling events; definitions and documentation of procedures; checklists for aiding in the execution of procedures; training aids; periodic and archival summaries of activities; specifications and limitations of personnel responsibilities; etc.

Operational start-up. Procedures must be established for putting a resource-sharing system into operation, and must include provisions for loading a fresh, certified copy of the Supervisor software, for verification of its correct loading, for validation of system security checks, for inserting relevant security parameters, and for certification of system security status by the System Security Officer.

Scheduled shutdown. The procedures for a scheduled shutdown of operations must take account of proper notification of the System Security Officer, physical protection of demountable storage (tapes, discs) as required, orderly closing of internal files, validation of the suspension of operation of all terminals, demounting of all copies (or required parts) of the Supervisor software, erasure of any parts of the Supervisor software remaining in working storage, verification of erasure of the Supervisor, disconnection of remote communication circuits, and physical securing of the power controls.

Unscheduled shutdown. An unscheduled shutdown must initiate procedures for immediate surveillance and recording of all indicators to help ascertain what happened; any needed emergency actions in case of fire, water hazard, etc.; special surveillance or physical protection measures to guarantee that no demountable items are removed; immediate notification of the System Security Officer; and special security controls (for example, protecting all printouts, including those at terminals, in accordance with protection rules for the highest classification handled in the system until the situation can be resolved).

Restart after unscheduled shutdown. If a trouble condition has caused the system to shut down, it is necessary that there be procedures to handle restart, including the loading of a new, certified copy of the Supervisor software, clearing the internal state of the equipment in order to clean up

memory untidiness resulting from the shutdown, verifying correct loading of the Supervisor, validating security controls and security parameters, and certifying the system security status by the System Security Officer.

File control. File control procedures include those for identifying the cognizant agency of each file, scheduling changes for files, modifying access restrictions of files, giving operators access to demountable files, moving files into and out of the computing area, pre-operator handling of files (including mounting and demounting of tapes and discs), and sanitization of files.

Control of magnetic tapes and discs. These procedures must account for and control the circulation and storage of tapes and discs; their use, reuse, and sanitization; and their classification markings and entrance to and release from the area.

Control of paper-based media. Procedures for punchcards, forms, papertape, and printouts must cover their accountability, classification marking, storage, and entrance to and release from the area. Additionally, manuals, guides, and various system documents must be covered.

Personnel control. Personnel control procedures include measures for verifying clearances and special-access authorization for personnel entry to each area of the system, visual surveillance of operating and maintenance areas, and logging and escorting of uncleared visitors. The reporting of suspicious behavior and security infractions is included among the personnel control procedures.

Terminal control. Various procedures are required with respect to the operation of remote terminals. These include provisions for logging user entry to the terminal area, removal of hardcopy, proper marking of hardcopy not marked by the system, clearing of displays, and securing as required during orderly shutdown.

Security parameter control. Procedures must be provided for authorizing security parameters to be entered into the system; for verifying correct entry; for changing them on the basis of shift, day of the week, etc.; for receiving and processing requests to modify them; and for actions to be taken in case of a system emergency or an external crisis.

Software control. These include procedures for rigid control and protection of certified copies of the Supervisor and other software bearing on system security or threat to the system, for loading the Supervisor, for making changes to it, and for verifying the changes.

Maintenance. All maintenance to be performed on hardware or software must be covered by appropriate procedures, including measures for surveillance of maintenance personnel by properly cleared personnel, for verifying with the System Administrator any adjustments made to the system's configuration, and for manually logging all changes and adjustments made or errors discovered.

Certification. Certification procedures should embrace various personnel responsibilities, tests and inspections to be performed and their conduct, the responsibilities of the System Security Officer, etc.

User aids. The production, distribution, and document control of manuals, guides, job procedure write-ups, etc., must be covered by appropriate procedures; there must be approved ways of conducting personnel training.

Change of mode. These procedures include the provision of checklists for actions required in changing mode, removal and storage of paper media and demountable files, physical and electronic surveillance of the machine area, purging of printers by running out the paper, purging of punchcard equipment by running out cards, removal or erasure of Supervisor software from the previous mode and proper verification thereof, loading of the Supervisor for the new mode and proper verification thereof, clearing of all storage devices so that residual information from the previous mode does not carry forward, removal of print ribbons from printers and terminal typewriters for storage or destruction, mounting of files for the new mode, and certification of the security status of the new mode.

Assurance of security control. Security control assurance includes procedures for reporting anomalous behavior of the system or security infractions; for monitoring security controls, including those on communications; for assuring continuity of security control; for devolution of responsibility in case of personnel nonavailability; and for auditing user and system behavior.

Appendix

AUTOMATION OF A MULTILEVEL SECURITY SYSTEM

INTRODUCTION

The basic multilevel security problem consists of determining whether an individual with a particular clearance and need-to-know can have access to a quantum of classified information in a given physical environment. While this problem exists independently of computer systems, the introduction of an automated decision process requires a formal specification of the decision rules used to answer this question. This Appendix addresses itself to one solution to that problem, detailing a language for defining security clearance structures, and a system that, given such a definition, will automate it and protect its integrity. This system provides for the classification and protection of information through a series of authorization checks which verify that an impending user action is permissible to the user in his current operational context.

The operating environment in which the proposed system will exist is not discussed, and will certainly vary depending on the equipment configuration of the installation. It is assumed, though, that the operating environment possess the following features:

- Integrity for both itself and the security system;
- Multiprogramming and/or on-line, interactive capability;
- A basic file system;
- Protection (read, write, and execute) for users from each other;
- A secure method of identifying and authenticating users;
- An interface with the security system that permits input/output for any user only after authorization by the security system.

Since the operating environment is not discussed in further detail, the implementation of the security system is specified only at the level of the logical processing that insures the integrity of the security system. The details of a monitoring system with which the System Security Officer can observe activity within the security system are also not treated here.

One important implementation issue that is covered, however, is the table-driven nature of the security system, facilitating on-line modification of system security parameters and minimizing the problem of separate certification of the system at each installation. Because of the complexity of the overall scheme for controlling access to classified information, it may be that the full range of security control mechanisms will not be necessary at each installation. Furthermore, as a matter of precaution, it would be undesirable to divulge unnecessarily to programming personnel the details of the security control methods. Therefore, the approach has been to conceive a scheme in which only the structure of the security control procedures need be described to programming personnel. The specific security parameters should not be available to such programmers, and must be inserted by the local System Security Officer.

It is proposed that a multi-access, remote-terminal computer system contain the following information:

- For each user, a list of certain parameters relevant to him;

- For each file, a list of certain access parameters relevant to the information contained in that file;
- For each terminal connected to the system, a list of certain parameters relevant to it.

The details of these parameters and how they are used are developed below.

Certain assumptions and definitions have been made for the purposes of this discussion:

(a) The System Security Officer must be aware of the structure of that portion of the total security system that is of concern to his installation.

(b) Access authorizations must be verified by explicit reference to a name check, organization check, other check, or combination of checks, etc., as may be required by security procedures. This is in addition to verification of the clearance status of the user requesting access to a given file.

(c) A *clearance* [1] status must be associated with both a user and a terminal; a *classification* [1] status must be associated with a file of information.

(d) The word *accesses*, when used below as part of the security structure language, is defined to be semantically equivalent to *permits access to information labelled as.*

(e) The phrase *national clearances* is taken to mean the normal defense clearances of Top Secret, Secret, Confidential, and Uncleared, which are hierarchical in that order. The national clearance status of an individual will be taken as the major parameter in controlling his access to classified information.

(f) If an individual is authorized to have access to information of Type A at one or more national clearance levels, then it is assumed that he is (in principle) granted access to Type A information up through the level of his national clearance. This is intended to rule out the following case, which we believe is common in present manual practice. An individual with a national clearance of Top Secret is authorized access to (say) cryptographic information (i.e., is granted Crypto access) only to the Secret level. This is regarded as an illegal use of the clearance control structure. For the purposes of the computer records, an individual granted (say) a national Top Secret clearance and access to information of Type A is automatically assumed to be cleared for all Type A information through the Top Secret level; this does not imply, however, that he is automatically authorized access to all levels of Type A information. Thus, it can be said that a national clearance factors or distributes over all special information types. The phrase *Type A* can refer to a special clearance system, a compartment or special grouping that may be within a special clearance system, or any major or minor segment of any clearance system that may have to be specified.

Comment: The above-mentioned special situation was ruled out for two reasons. First, discussion with several security officers indicated that it is, in fact, a misuse of the security system. Second, the inclusion of this case would introduce a logical inconsistency in the security control processing described herein, thereby making it possible to circumvent the system. While this could be corrected, the cost, in terms of computer processing, would be prohibitively high, and the first reason makes it unnecessary.

(g) As a consequence of the above, the computer algorithm which matches the parameters of the user against the parameters of the file to be accessed will first compare the user's national clearance and the file's national classification. If a user is to be granted access to a given file, then his national clearance level must equal or exceed the national classification level of the file. Note that this is a necessary but not sufficient condition for access. Additional controls, such as code words, special access categories or compartments, etc., will be regarded as controlling access to specific information types within the framework of the national clearance structure.

(h) A *dissemination label* is regarded as an additional means of access control, and will require verification against the user's status. Examples of such labels are "No Foreign Dissemination" and "Not Releasable Outside the

[1] These terms are defined on p. 12.

Department of Defense."

(i) An *information label* is regarded as not controlling access to information, but rather giving guidance to the user on how the information may be further disseminated, controlled, utilized, etc. Examples of such labels are "Limited Distribution," "Special Handling Required," "Downgrading Group 1."

(j) All names, code words, etc., are assumed to be unique.

COMPUTER SYSTEM CATALOGS

The computer system will maintain a catalog of all terminals that may be connected to it. For each terminal, it will maintain the following information:

(a) The highest classification level of information that may be transmitted to or from the terminal—i.e., the terminal clearance level.

(b) Special code words, group names, or other names that modify the clearance level of the terminal to receive other classes of information.

(c) A list of the users authorized to use the terminal (this may be "ALL").

(d) The electrical address.

(e) The permanent identification number.

(f) Physical location, including building location, room number, and the cognizant agency.

(g) Person responsible for the terminal and (perhaps) his telephone number.

The first three items above may be time and date dependent; different parameters may be specified for different periods, such as normal working hours, holidays, weekends, and night shifts.

The computer system will maintain a catalog of all users authorized to have access to it, and for each user will maintain the following information:

(a) His national clearance level, its date of expiration, and its granting agency. (If necessary, its date of issuance can be included.)

(b) Special code words and groupings or other words that extend his access to other classes of information, and the date of expiration of each such special name.

(c) His agency affiliation.

(d) His citizenship.

(e) His agency assignment(s).

(f) His permanent identification number (Social Security or other).

(g) Special need-to-know designators other than those explicitly contained in the first and third items.

The computer system will maintain the following information for each file:

(a) Its national classification level.

(b) Special names, such as code words, compartment names, handling labels, etc., that serve to control access to the file.

(c) Access authorization lists, including one or more of the following as may be required:
- Universal authorization lists (i.e., everyone is authorized access);
- Name lists;
- Group designator authorizations (group membership information is maintained by the system in support of access authorization processing);
- *Specific exclusions from* access authorization by such things as groups, names, explicit lists of names.

(d) Dissemination labels.

(e) Information labels.

(f) Background information on the file; examples of information that might be desired are:
- Its date of creation;
- Its downgrading group, and any downgrading actions applied to it;
- Name of individual who created the file and his agency;
- Predecessor files (if any) from which the file was created.

SECURITY CONTROL SYSTEM GENERATION

The system for automating multilevel security classification and control here described is entirely table driven. As such, the same software implementation can be used at all installations using the same machine. The generation process described below creates the tables used by the system, but does not

affect the software or any of its built-in checks. Thus, installation personnel need not know about or implement any part of the security control system; nor should they be expected or allowed to modify it. Each installation, through the security control system generation process, particularizes the security tables to its environment (with built-in validity and consistency checks), and thus can minimize recertification of the security control system.

The card deck (or magnetic tape or magnetic disc) detailing the security control system and the tables produced during the generation process contain the most sensitive information resident in the computer system. As such, no provision is made for directly classifying or accessing this information via the file system; rather, special mechanisms must be provided to limit access to this information to only the responsible authorities.

System Access Definition is the vehicle for describing to the computer system those parameters that will affect an individual's access to information. This consists of a *Personnel Definition*, describing all relevant parameters for the individuals permitted to use the system, *except* information dealing with security; a *Terminal Definition*, describing all relevant parameters for any terminals that may be connected to the system, *except* information dealing with security; and a *Security Control Definition*, describing all relevant security parameters. The Personnel and Terminal Definitions are not discussed here, since they are installation dependent and are not within the scope of this Report.

Security control system generation is the process whereby the System Security Officer (or other responsible authority) specifies the Security Control Definition to the computer system. The computer system will process this information, doing such things as validity checking and internal table storage generation, and thus render the system ready for actual use. After the initial security system has been generated, changes to the Security Control Definition can (in almost all cases) be handled directly by the system without cause for regenerating the security control system.

The Security Control Definition consists of five separate specifications: Security Structure Definition, Personnel Security Definition, Authorization Group Definition, Terminal Security Definition, and Releasability Definition. The *Releasability Definition* specifies the dissemination labels and the way they are processed. It is not discussed here because we have been unable to determine any standardized, rigorous order in the current practice of using such labels. We recommend that this area be further explored. Note that the processing of the dissemination labels will depend upon the Personnel Definition. For example, a "DoD Only" file will necessitate the ability to determine the agency that the individual represents.

The other four specifications of the Security Control Definition are discussed below. The reader is directed to Annex A for the formal System Access Specification in a slightly modified Backus-Naur Form (BNF). In addition to the language specification, it is necessary to specify the algorithms for processing this information. These are discussed below in all but the obvious cases. The reader should reference the Annexes as he reads the remainder of the discussion, particularly Annex B, which contains examples of Security Component Definitions.

SECURITY STRUCTURE DEFINITION

The *Security Structure Definition* formally defines the structure of that portion of the security classification and control system that is applicable to the particular installation in question. The language presented in Annex A is sufficient to describe all special clearances and compartments with which we are familiar, although actual examples demonstrating the completeness of this approach cannot be presented at this level of classification.

The Security Structure Definition consists of any number of *Security Component Definitions*, followed by any merge rules relating different components. A *component* may be a compartment, a special category, or a special access. It is reasonable to expect that changes to the Security Structure Definition will necessitate a new system generation.

The security structure language formally defines a set of relations among entities, including names of clearances or classifications, code words, labels, etc. The structure below can be thought of as defining a set of decision rules that the computer system can consult when it wishes to make a decision concerning security parameters. It is immaterial as to how these decision rules are actually stored in the com-

puter, and this is (for the present) left to the individual software system designers.

Following is an example of a Security Component Definition:[2]

DEFINE: NATIONAL CLEARANCES;

CLEARANCES: TOP SECRET, SECRET, CONFIDENTIAL, UNCLEARED;

SYNONYMS: TOP SECRET = TS, SECRET = S, CONFIDENTIAL = C, UNCLEARED = UR, UNCLASSIFIED = U;

INTERNAL STRUCTURE: TS IMPLIES S, S IMPLIES C, C IMPLIES UR;

ACCESS RULES: TS ACCESSES TS, S ACCESSES S, C ACCESSES C, UR ACCESSES U;

REQUIRED LABELS: NONE;

EXTERNAL STRUCTURE: NONE;

REQUIREMENTS: NONE;

MERGE RULES: TS **AND** (S **OR** C **OR** U) YIELDS TS, S **AND** (C **OR** U) YIELDS S, C **AND** U YIELDS C;

END;

The component name (as specified in the *DEFINE* statement) is the name normally applied to a classification system, compartment, or special category. It, and all *CLEARANCES* within the component, are listed in the definition. Note that a component name and a clearance name may be the same. *SYNONYMS* allows for commonly used abbreviations or synonyms.

The *INTERNAL* and *EXTERNAL STRUCTURE* statements (i.e., internal and external to the particular component in question) are handled the same way by the system software. They are stipulated separately in the definition merely to assist the System Security Officer in organizing his thoughts as he defines the security structure. A possible use of the *EXTERNAL STRUCTURE* statement is to create Universal Privileges, as discussed below; its use is also illustrated in Example 4 of Annex B. These statements describe hierarchical relationships that exist between one of the clearances being defined in the component, and either another clearance within that component or a clearance from another component, respectively. This is interpreted to mean that access authorized by a given clearance implies the automatic access (unless otherwise limited) authorized by other clearances lower in the hierarchy. For example, if an individual has a Top Secret clearance, Top Secret implies Secret *(TS IMPLIES S)* in the sense that an individual cleared for Top Secret also has access to information to which an individual cleared for Secret has access.

Under *ACCESS RULES*, there is only one operator, called *accesses*, which has been previously defined as *permits access to information labelled as*. These rules explicitly state the relation between the names of the clearances in the security component being defined and the labels on the information to which that security clearance permits access. In many cases, the *same* word is used to specify a clearance and a label indicating classification of information (as in the example above).

The *REQUIRED LABELS* are those other than the normal classification labels on a file. For example, certain security components require all information within the component to be handled via special channels, and this fact is explicitly stated on any piece of information protected by the component. In effect, a required label can be regarded as a pseudo-classification, accessed by any of the clearances listed in the Security Component Definition (or their synonyms). The necessity of this view is indicated in the Crypto example of Annex B (Example 1), where administrative traffic not having the Crypto classification label, but still confined to Crypto-authorized people, must be recognized by the system.

Note that information and dissemination labels, although required on information, are not included here as *REQUIRED LABELS* because at present their usage is neither standardized nor logically consistent. When their usage becomes standardized, it will be possible to revise slightly the scheme here described to accommodate them and handle them automatically.

The *REQUIREMENTS* statement is the vehicle for describing situations in which a particular clearance requires the simultaneous existence or nonexistence of other clearances or access authorizations (see Examples 2-4 in Annex B). Note that classification labels are not mentioned, since the particular labels accessed by a given clearance can always be determined.

[2] Additional examples are found in Annex B.

MERGE RULES, discussed more fully below, contain the information that allows the system to determine automatically the classification of information that results from merging information of various classifications. Standard logical relationships (utilizing the Boolean connectives **AND** and **OR**) are permitted.

The operator *YIELDS* means that the combination of classifications (or labels) on the left requires the classification (or labels) on the right to be placed on the merged information.

Security Structure Preprocessing for Minimization of Clearances

After the complete Security Structure Definition has been entered into the computer, an augmented set of Requirement statements will be automatically constructed as follows. For each implication statement of the form *A IMPLIES B* in either an Internal or an External Structure statement, the Requirement statement of *B* will be modified by the conjunction of **NOT** *A*. If there is no previous Requirement statement for *B*, then one must be created.

The purpose of this is to provide for consistency in the minimization of the user's clearance set. For example, if an individual is to be granted a Top Secret clearance after already possessing a Secret clearance, the system should rightfully expect that his Secret clearance be removed when the Top Secret is granted. Similarly, there are instances of interrelated components where it is mandatory that a clearance not mutually coexist with another clearance that implies it (see Example 4 in Annex B). The system includes this capability, and this results in the following rule:

> When upgrading any user clearance that is hierarchical, the security officer *must* first remove the lower clearance and then add the higher clearance.[3]

In the example just given, this means that the security officer must remove the user's Secret clearance *before* adding the user's Top Secret status to the system. (The system's consistency checking mechanism described below will prevent the Top Secret clearance from being accepted before the Secret clearance is deleted.)

[3]As described below, the user is not allowed to be logged onto the system while his clearance status is being modified, nor can his status be changed while he is logged on the system.

Consistency Check of the Security Structure Definition

After all Security Component Definitions have been entered into the computer and preprocessing has been completed, two consistency checks are made. The first insures that all clearances referenced have been defined and that no clearance is multiply-defined. The second insures that no chains exist that lead to contradictions. For example, *A requires B, B requires C, C requires* **NOT** *A*, would form an inconsistent set of clearances in which clearance *A* could never be granted.

The consistency check is performed as follows for *each* clearance in the Security Structure Definition:

(a) Form an expression, called the *consistency expression*, consisting of the clearance being tested.

(b) Moving through this consistency expression from left to right, pick up the next clearance in the expression and replace it by itself conjuncted with the right-hand side of the Requirements statement for that clearance (from its Security Component Definition), all enclosed in parentheses.

(c) Repeat step (b) above, each time moving to the next clearance appearing in the consistency expression (i.e., the next one to the right of the one just processed), until all clearances in the consistency expression have been processed.

(d) Assign the value of *TRUE* to the next (leftmost) clearance in the consistency expression (i.e., to the one being tested for consistency with the rest of the security structure).

(e) If any set of assignments of *TRUE* and *FALSE* can be made to the other clearances in the consistency expression which result in a value of *TRUE* (when the expression is evaluated according to the normal rules of Boolean expression evaluation), then the clearance being tested is consistent with the rest of the Security Structure Definition.

(f) If no such assignment can be found to make the consistency expression *TRUE*, then the clearance being tested is inconsistent with the

rest of the Security Structure Definition. The consistency expression and the inconsistent clearance must be output by the system to facilitate the correction of the inconsistency. The consistency check should continue to look for further inconsistencies, but the particular Security Structure Definition cannot be accepted by the system. (The system cannot allow any type of error in the Security Structure Definition.) After correcting the inconsistency, the *entire* process of Security Structure Definition must be restarted from the beginning. Also, because of the complex processing described above, there is no provision for on-line definition of new clearances.

(g) Repeat steps (d), (e), and (f) above, each time moving to the next clearance appearing in the consistency expression (i.e., the next one to the right of the one just processed), until all clearances in the consistency expression have been processed.

Merge Rules

Merge rules are provided to permit automatic determination of the classification of information that has been produced by the combination of information of dissimilar classifications (see the example above of National Clearances, and also Examples 2-4 in Annex B). Note that all relationships, including hierarchical ones, must be explicitly stated in terms of classification labels; the software cannot be expected to infer that one classification subsumes another.

Merge Rule Processing

The actual merge rule processing is as follows:

(a) Concatenate (i.e., conjunct) all the labels of each file accessed during the merge process (this includes required labels).

(b) Simplify resultant merge label by the following rules:

(1) Identity transformation. *A* **AND** *A* yields *A* for all *A;*

(2) Apply merge rules; i.e., if the left-hand side of a special merge rule matches the concatenated labels or a portion thereof, replace that portion by the right-hand side of the rule. (Treat the left-hand side of the merge as a Boolean expression and evaluate according to the normal rules. If a label appears in the concatenated label set, consider it *TRUE* in the expression; otherwise, *FALSE.* Hence, the right side is substituted for the left side of a merge rule when the left side is *TRUE.)*

In attempting to apply steps (1) and (2) above, the labels can be freely reordered to promote a simplification.

(c) If any simplification results from step (b), then repeat steps (b) and (c).

PERSONNEL SECURITY DEFINITION AND USER CLEARANCE UPDATE

The next step in system generation is *Personnel Security Definition.* It is possible to modify this information subsequently through the on-line use of the user clearance update language. The processing involved is the same for both initial system generation and subsequent updates, and is as follows:

(a) Update of a user's clearance status by the security officer can be done if and only if the user is not logged onto the system.

(b) The granting agency and expiration date may be specified for clearances and put into the user's information, but are not presently utilized. The cognizant agency is neither specified nor stored. This implies that within this automated security system, a Top Secret clearance granted from one agency also implies access to Top Secret information from another agency, unless additional labels that deny such access have been applied to this information.

(c) On each addition or deletion of a user clearance, a check will be made that the user exists; that (on addition) the clearance exists and has not already been granted to the user; and (on deletion) that the user does, in fact, have the clearance to be deleted.

(d) At the time of Personnel Security Definition, and at the time of granting an additional clearance to (or removing an existing clearance from) a user, a consistency check is made to insure that the Requirements statement for

each of the user's clearances is still satisfied after the addition (deletion) of the new (old) clearance; this is accomplished as follows:

(1) Generate the set of access privileges specified by the user's explicit clearances; this can be done as follows:
- Form the set of all the user's explicit clearances (called the *clearance set);*
- For each clearance in the clearance set, add all clearances implied by this particular clearance in either Internal or External Structure statements within the Security Component Definition;
- Apply identity transformation *(A* [AND] *A* yields *A)* to the clearance set (i.e., remove all duplicates).

Notice that this is the algorithm used in generating the set of all labels to which the user's clearance permits access (explained below in "File Access Processing") with steps (b), (c)(1), and (c)(3) deleted.

(2) For each explicit clearance the user has been granted, including the new one being added (or excluding the old one being deleted), check to see if the requirements as stated in the Requirements statement(s) in the Security Component Definition are satisfied by the occurrence or absence of the clearances in the clearance set just generated according to the normal rules of Boolean expression evaluation.

AUTHORIZATION GROUP DEFINITION

Authorization Group Definition occurs at system generation time, but, like Personnel Definition, also may be updated on-line. There is no special processing explicitly required for authorization groups. A user does not have to be authorized to use the system for his name to be in an authorization group. Updates are made via the authorization group update language.

Comment: Our concept of an authorization group is more general than the normal need-to-know concept associated with classified information. It also addresses the question of what a person can do to the information to which he has in fact been granted access. In the usual context, need-to-know is really need-to-know for reading. We have simply extended that concept to allow separate need-to-know groups for reading, changing, etc., and we call this extended concept "authorization groups" in order to avoid confusion.

UNIVERSAL PRIVILEGES

Under emergency conditions, it may be necessary to grant a user or a group of users unrestricted access to all files in the system or to a set of files regardless of clearances, special access categories, and/or need-to-know restrictions. Rather than turning off the file safeguards in the system, necessitating concern for user identification, protection of terminals, etc. (especially under emergency conditions), a special capability is provided within the system so that the system security controls are not impaired.

The System Security Officer in a normal Security Component Definition can define a universal or emergency clearance, which implies all other clearances or special-access categories in the system and which has no external requirements. It can be granted to a given user by first removing all his clearances (to prevent a clearance inconsistency check) and then granting the universal or emergency clearance. (Obviously, any number of such emergency clearances could be set up for any subsets of the overall security system by simply listing the desired ones in the External Structure statement.)

Universal authorization groups can be defined to handle the problem of overriding the system's file manipulation and access authorization restrictions. Membership in such a group authorizes the individual to take some action on the files to which he is permitted access, either on a standing or an emergency basis. Examples of universal authorizations are: universal right-to-read, universal right-to-change, etc.

Comment: The word "emergency" is used here in a limited sense; i.e., we refer mainly to the numerous unanticipated special situations that always seem to arise at any computer installation. Through appropriate forethought and predefinition, these situations can be handled routinely as they arise. Still, however, there may arise a true emergency (such as an enemy

attack) where there is no time to do anything but respond. The techniques discussed here are not intended to address that problem. Rather, we would assume some sort of fail-safe, joint-key mechanism whereby appropriately authorized individuals could turn off all access controls of the system in time of dire emergency.

Mechanisms such as described above should be sufficient for accommodating any specific situations that may arise, assuming the appropriate universal groups have been predefined. In addition, they allow routine handling of two situations normally requiring special provisions. These are the privileges of the System Security Officer and the file-backup mechanism. The System Security Officer should have, in addition to his normal clearance status, universal authorizations for read-only, right-to-change authorization lists, and right-to-change file classifications. The file backup program can be given the clearance status to handle all files for which it is to provide backup and universal authorization for read-only to enable it to read any of these files.

TERMINAL SECURITY DEFINITION AND UPDATE

Terminal Security Definition is handled in a manner similar to personnel security information. There exists the capability to update this information online. In the present specification, the capability to specify a terminal access list has not been included; i.e., a list of the authorized users of a given terminal. It appears, for the present, that this is an unnecessary complexity to add to an already burdened system, and we expect that physical access to terminals processing classified information will normally be controlled. Further control seems unnecessary, but should it be desired, mechanisms similar to those already specified can be used. For example, a special clearance status can be defined, access to which is permitted only for a particular terminal.

Specification of File Authorizations

Each time a file is created, the creator may specify which individuals or groups of individuals are permitted to access the file, as well as how they may do so; e.g., read-only. For each file, the author may therefore specify authorizations and an access list to be associated with each authorization.

If not specified, default access lists are assumed as follows:

> All authorization access lists have the default condition of null (i.e., unless otherwise specified, they are empty) *except* those associated with the following actions: unrestricted access, right-to-change authorization lists, and right-to-change file classifications. The access lists associated with these particular authorization types must be initialized by the system to contain the name of the author of the file.

It should be noted that the syntax of the authorization specification provides capability for the removal of the author's name from an access list. Unless this is explicitly done, however, the author of a file will be permitted unrestricted access to the file, as well as the privilege of changing the authorization specification and classification of the file.

At present, it is not deemed necessary to provide the capability to be able to syntactically distinguish between authorization group identifiers and user identifiers. Rather, it is assumed that the processing algorithms will have to check the identifier in question against master lists, and that the semantics will be obvious from the context.

Anyone who has the ability to write in a file can, in principle, add to it information of a higher classification than the file. Therefore, he must have some way of altering the classification status of the file. Whether this is provided by allowing anyone with write privilege to alter the file classification directly, or by requesting the original author of the file to alter the classification, or by requesting the System Security Officer to alter the classification, is an operational policy decision. The first alternative is simplest, but it may be operationally desirable to have a second person involved in change of classification. The mechanisms in the overall scheme provide capability to specify a separate group of individuals who can only alter the classification of a file.

FILE ACCESS PROCESSING

The system must follow certain procedures when attempting to determine whether or not a given user

may reference a particular file of information. First, the user's clearance must be sufficient to permit access to the file classification, and this is determined as follows:

(a) Obtain the file classification labels.
(b) Obtain the set of labels to which user clearances permit access. This set may be calculated as needed at log-on time or at security system update time (if the latter is used, on-line updating of a user's clearance by the System Security Officer cannot be allowed).
(c) If the set of labels to which the user's clearance status permits him access contains all the labels in the file classification status, then the formal security accessing requirements have been satisfied.

The method of generating the set of labels to which a user's clearance status permits him access is as follows:

(a) Form the set of all user's clearances and special access categories (called *clearance set).*
(b) Initialize to null the set of labels to which the user's clearance status permits him access (called the *accessible label set).*
(c) For *each* entry in the clearance set:
    (1) Add to the accessible label set all labels to which the particular entry permits access. These are obtained from the access rules in the Security Component Definition. Also, add all required labels for this particular clearance entry.
    (2) Add to the clearance set all clearances or special-access categories implied by this particular clearance entry in either Internal or External Structure statements within the Security Component Definition.
    (3) Delete this entry from the clearance set.
(d) Apply identity transformation *(A* **AND** *A* yields *A)* to the accessible label set (i.e., delete all duplicates).

After a user's clearance status has been checked and successfully permits access to a file, the security system must determine whether the user satisfies the authorization limitations for the file. This check determines the user rights and specifies what types of manipulation he is allowed for the file in question. The process for carrying this out is as follows:

(a) Copy the user's universal authorization privileges (which are explicitly specified at log-on time by the universal authorization algorithm described below) into a memory area called his *file-access rights block.* If he has universal unrestricted access after specifying this in the file-access-rights block as explained in step (b)(2) below, then processing can stop (i.e., there is nothing that can be added to his access rights).
(b) For each authorization type (starting with unrestricted access):
    (1) If the user is in the access list either explicitly (by name) or implicitly (either by membership in a group specified in the list or because the universal set was specified), grant the user the specified type of access;
    (2) If the authorization is for unrestricted access and the user qualifies for it, grant him (in his file-access-rights block for this file) all the other authorization types, and stop processing these rights.

The file-access-rights information (in the file-access-rights block) is consulted by the Supervisor on every input/output operation in order to determine whether or not the operation on the file is legal. Thus, the authorization processing occurs during the linkage of a user to a file after clearance status checks have been made, and results only in the creation of the file-access-rights data, which is later used by the Supervisor for controlling access to the file.

The universal authorization algorithm consists of checking each universal group for the presence of the user in the set, either explicitly by name or implicitly by membership in another group specified as a member of the universal group. If the user is present in the set, then grant him the associated universal access privilege.

Comment: When access control labels are standardized and any precedence or combinatorial relations among them have been specified, the algorithms for handling them can be developed, and the restrictions resulting from the operation of such algorithms would be examined at this point in file access processing.

Annex A:

FORMAL SYSTEM ACCESS SPECIFICATION

Notation: Standard Backus-Naur Form (BNF), plus:

- **[x] means one or more occurrences of x separated by commas, with no initial or terminal comma.**
- **Also, if any <STRING> contains one of the fixed words appearing in the following BNF rules that could lead to an ambiguity, the <STRING> should be enclosed in parentheses.**

System Access Definition

```
<SYSTEM ACCESS DEFINITION> ::= <PERSONNEL DEFINITION>
     <TERMINAL DEFINITION> <SECURITY CONTROL DEFINITION>

<PERSONNEL DEFINITION> ::=    Not part of this specification.

<TERMINAL DEFINITION> ::=     Not part of this specification.

<SECURITY CONTROL DEFINITION> ::= <SECURITY STRUCTURE DEFINITION>
     <PERSONNEL SECURITY DEFINITION> <AUTHORIZATION GROUP DEFINITION>
     <TERMINAL SECURITY DEFINITION> <RELEASABILITY DEFINITION>

<RELEASABILITY DEFINITION> ::= Not part of this specification.
```

Security Structure Definition

```
<SECURITY STRUCTURE DEFINITION> ::=
     <SECURITY COMPONENT DEFINITION> <MERGE RULES> |
     <SECURITY COMPONENT DEFINITION> <SECURITY STRUCTURE DEFINITION>

<SECURITY COMPONENT DEFINITION> ::= <DEFINE STATEMENT>
     <CLEARANCE STATEMENT> <SYNONYM STATEMENT>
     <INTERNAL STRUCTURE STATEMENT> <ACCESS RULE STATEMENT>
     <REQUIRED LABEL STATEMENT> <EXTERNAL STRUCTURE STATEMENT>
     <REQUIREMENT STATEMENT> END;

<DEFINE STATEMENT> ::= DEFINE: <COMPONENT NAME>;

<CLEARANCE STATEMENT> ::= CLEARANCES: [<CLEARANCE NAME>];

<SYNONYM STATEMENT> ::= SYNONYMS: NONE; | SYNONYMS: [<SYNONYM PAIR>];

<INTERNAL STRUCTURE STATEMENT> ::= INTERNAL STRUCTURE: NONE; |
     INTERNAL STRUCTURE: [<CLEARANCE NAME> <BLANKS> IMPLIES
     <BLANKS> <CLEARANCE NAME>];

<ACCESS RULE STATEMENT> ::= ACCESS RULES: NONE; |
     ACCESS RULES: [<CLEARANCE NAME> <BLANKS> ACCESSES <BLANKS>
     <LABEL>];

<REQUIRED LABEL STATEMENT> ::= REQUIRED LABELS: NONE; |
     REQUIRED LABELS: [<REQUIRED LABEL>];
```

```
<EXTERNAL STRUCTURE STATEMENT> ::= EXTERNAL STRUCTURE: NONE; |
     EXTERNAL STRUCTURE: [<CLEARANCE NAME> <BLANKS> IMPLIES
     <BLANKS> <EXTERNAL CLEARANCE NAME>];

<REQUIREMENT STATEMENT> ::= REQUIREMENTS: NONE; |
     REQUIREMENTS: [<CLEARANCE NAME> <BLANKS> REQUIRES <BLANKS>
     <CLEARANCE EXPRESSION>];

<CLEARANCE EXPRESSION> ::= <PRIMARY> | <PRIMARY> <BOOLEAN OPERATOR>
     <PRIMARY>

<PRIMARY> ::= (<CLEARANCE EXPRESSION>) | <CLEARANCE NAME> |
     <BLANKS> NOT <BLANKS> <PRIMARY>

<BOOLEAN OPERATOR> ::= <BLANKS> AND <BLANKS> | <BLANKS> OR <BLANKS>

<SYNONYM PAIR> ::= <BASIC NAME> = <SYNONYM NAME>

<BASIC NAME> ::= <COMPONENT NAME> | <CLEARANCE NAME> | <LABEL NAME>

<LABEL NAME> ::= <LABEL> | <REQUIRED LABEL>

<SYNONYM NAME> ::= <STRING>

<EXTERNAL CLEARANCE NAME> ::= <STRING>

<COMPONENT NAME> ::= <STRING>

<CLEARANCE NAME> ::= <STRING>

<LABEL> ::= <STRING>

<REQUIRED LABEL> ::= <STRING>

<STRING> ::= <LETTER> | <LETTER> <CHARACTER STRING>

<CHARACTER STRING> ::= <NONBLANK CHARACTER> | <CHARACTER>
     <CHARACTER STRING>

<CHARACTER> ::= <NONBLANK CHARACTER> | <SPACE> | <HYPHEN>

<NONBLANK CHARACTER> ::= <LETTER> | <DIGIT>

<LETTER> ::= A | B | C | ... | Y | Z

<DIGIT> ::= 0 | 1 | 2 | ... | 8 | 9

<BLANKS> ::= <SPACE> | <SPACE> <BLANKS>

<MERGE RULES> ::= <MERGE RULE STATEMENT> END;

<MERGE RULE STATEMENT> ::= MERGE RULES: NONE; |
     MERGE RULES: [<MERGE RULE>];

<MERGE RULE> ::= <MERGE CONDITION EXPRESSION> <BLANKS> YIELDS
     <BLANKS> <RESULTANT STRING>

<MERGE CONDITION EXPRESSION> ::= <MERGE PRIMARY> | <MERGE PRIMARY>
     <BOOLEAN OPERATOR> <MERGE PRIMARY>

<MERGE PRIMARY> ::= (<MERGE CONDITION EXPRESSION>) | <LABEL NAME> |
     <BLANKS> NOT <BLANKS> <MERGE PRIMARY>
```

```
<RESULTANT STRING> ::= <LABEL NAME> | <LABEL NAME> <BLANKS> AND
    <BLANKS> <RESULTANT STRING>
```

Personnel Security Definition

```
<PERSONNEL SECURITY DEFINITION> ::= END; | <USER CLEARANCE STATEMENT>
    <PERSONNEL SECURITY DEFINITION>

<USER CLEARANCE STATEMENT> ::= [<USER ID>]:
    [(<CLEARANCE NAME>, <GRANTING AGENCY>, <EXPIRATION DATE>)];

<USER ID> ::= <NONBLANK CHARACTER> | <NONBLANK CHARACTER> <USER ID>

<GRANTING AGENCY> ::= <LETTER> | <LETTER> <GRANTING AGENCY>

<EXPIRATION DATE> ::= <MONTH> / <DAY> / <YEAR>

<MONTH> ::= <DIGIT> <DIGIT>

<DAY> ::= <DIGIT> <DIGIT>

<YEAR> ::= <DIGIT> <DIGIT>
```

User Clearance Update Language

```
<USER CLEARANCE UPDATE LANGUAGE> ::= <GRANT USER CLEARANCE STATEMENT> |
    <REMOVE USER CLEARANCE STATEMENT>

<GRANT USER CLEARANCE STATEMENT> ::= GRANT [(<CLEARANCE NAME>,
    <GRANTING AGENCY>, <EXPIRATION DATE>)] TO USER [<USER ID>]

<REMOVE USER CLEARANCE STATEMENT> ::= REMOVE <CLEARANCE SET> FROM USER
    [<USER ID>]

<CLEARANCE SET> ::= ALL CLEARANCES | ([<CLEARANCE NAME>])
```

Authorization Group Definition

```
<AUTHORIZATION GROUP DEFINITION> ::= END; |
    <AUTHORIZATION GROUP SPECIFICATION>
    <AUTHORIZATION GROUP DEFINITION>

<AUTHORIZATION GROUP SPECIFICATION> ::= <AUTHORIZATION GROUP NAME>:
    [<AUTHORIZATION TYPE>]
    ([<AUTHORIZATION GROUP ELEMENT>]);

<AUTHORIZATION GROUP NAME> ::= UNIVERSAL <AUTHORIZATION TYPE> |
    <AUTHORIZATION GROUP IDENTIFIER>

<AUTHORIZATION TYPE> ::= READ ONLY | CHANGE ONLY |
    APPEND ONLY | EXECUTE ONLY | UNRESTRICTED ACCESS |
    RIGHT-TO-CHANGE AUTHORIZATION SPECIFICATION |
    RIGHT-TO-CHANGE FILE CLASSIFICATION

<AUTHORIZATION GROUP ELEMENT> ::= <AUTHORIZATION GROUP IDENTIFIER> |
    <USER ID>
```

```
<AUTHORIZATION GROUP IDENTIFIER > ::= <NONBLANK CHARACTER> |
     <NONBLANK CHARACTER> <AUTHORIZATION GROUP IDENTIFIER>
```

Authorization Group Update Language

```
<AUTHORIZATION GROUP UPDATE LANGUAGE> ::= <DEFINE GROUP STATEMENT> |
     <ADD MEMBER STATEMENT> | <REMOVE MEMBER STATEMENT>

<DEFINE GROUP STATEMENT> ::= DEFINE GROUP <AUTHORIZATION GROUP NAME>:
     [<AUTHORIZATION TYPE>]
     ([<AUTHORIZATION GROUP ELEMENT>])

<ADD MEMBER STATEMENT> ::= ADD ([<AUTHORIZATION GROUP ELEMENT>])
     TO GROUP [<AUTHORIZATION GROUP NAME>]

<REMOVE MEMBER STATEMENT> ::= REMOVE ([<AUTHORIZATION GROUP ELEMENT>])
     FROM GROUP [<AUTHORIZATION GROUP NAME>]
```

Terminal Security Definition

```
<TERMINAL SECURITY DEFINITION> ::= END; |
     <TERMINAL CLEARANCE STATEMENT> <TERMINAL SECURITY DEFINITION>

<TERMINAL CLEARANCE STATEMENT> ::= [<TERMINAL ID>]: <CLEARANCE SET>;

<TERMINAL ID> ::= Installation dependent--not specified here (may
                  not include comma, colon, or semicolon).
```

Terminal Clearance Update Language

```
<TERMINAL CLEARANCE UPDATE LANGUAGE> ::=
     <GRANT TERMINAL CLEARANCE STATEMENT> |
     <REMOVE TERMINAL CLEARANCE STATEMENT>

<GRANT TERMINAL CLEARANCE STATEMENT> ::= GRANT <CLEARANCE SET>
     TO TERMINAL <TERMINAL ID>

<REMOVE TERMINAL CLEARANCE STATEMENT> ::= REMOVE <CLEARANCE SET>
     FROM TERMINAL <TERMINAL ID>
```

File Authorization Specification

```
<FILE AUTHORIZATION SPECIFICATION> ::= <FILE NAME>:
     [(<AUTHORIZATION TYPE>
     <AUTHORIZATION ACCESS LIST>)]

<AUTHORIZATION ACCESS LIST> ::= UNIVERSAL | UNIVERSAL
     <SET SUBTRACTION OPERATOR> <AUTHORIZATION EXPRESSION> |
     <AUTHORIZATION EXPRESSION>

<AUTHORIZATION EXPRESSION> ::= <AUTHORIZATION GROUP> |
     <AUTHORIZATION GROUP> <AUTHORIZATION OPERATOR>
     <AUTHORIZATION EXPRESSION>
```

```
<AUTHORIZATION GROUP> ::= ([<AUTHORIZATION IDENTIFIER>])

<AUTHORIZATION IDENTIFIER> ::= <AUTHORIZATION GROUP IDENTIFIER> |
    <USER ID> | AUTHOR

<AUTHORIZATION OPERATOR> ::= <SET ADDITION OPERATOR> |
    <SET SUBTRACTION OPERATOR>

<SET ADDITION OPERATOR> ::= +

<SET SUBTRACTION OPERATOR> ::= -

<FILE NAME> ::= Operating system dependent--not specified here (may
                not include colon).
```

Annex B

SECURITY COMPONENT DEFINITION EXAMPLES

Example 1

Consider a class of information called Crypto, which is to be regarded as a further restriction on access under the national clearance system. Since Crypto information is to be transmitted via special channels, and is labelled as such, administrative traffic without the classification label Crypto can still be confined to Crypto-authorized personnel by regarding the required label on the file as a pseudo-classification accessed by any of the clearances listed in the definition.

DEFINE: CRYPTO;

CLEARANCES: CRYPTO;

SYNONYMS: CRYPTO = CRP;

INTERNAL STRUCTURE: NONE;

ACCESS RULES: CRP ACCESSES CRP;

REQUIRED LABELS: HANDLE VIA SPECIAL CHANNELS;

EXTERNAL STRUCTURE: NONE;

REQUIREMENTS: CRP REQUIRES TS OR S;

MERGE RULES: NONE;

END;

Example 2

Consider a hypothetical refinement of the national clearance system called DATATEL as follows:

DEFINE: DATATEL;

CLEARANCES: III, II, I;

SYNONYMS: NONE;

INTERNAL STRUCTURE: III IMPLIES II, II IMPLIES I;

ACCESS RULES: III ACCESSES ABLE, II ACCESSES BAKER, I ACCESSES CHARLIE;

REQUIRED LABELS: HANDLE VIA DATATEL CHANNELS ONLY;

EXTERNAL STRUCTURE: NONE;

REQUIREMENTS: III REQUIRES TS, II REQUIRES S, I REQUIRES C;

MERGE RULES: ABLE AND (BAKER OR CHARLIE) YIELDS ABLE, BAKER AND CHARLIE YIELDS BAKER;

END;

Example 3

Now consider a hypothetical compartment of information within the DATATEL structure. It has been assumed that APPLE information is not labelled as such, but is to carry the codeword ALICE. The APPLE definition below relates APPLE to III; the DATATEL definition relates III to ABLE and also to Top Secret. Thus, the system can correctly determine that the proper classification label for APPLE information is TOP SECRET ABLE ALICE. Note also that such information has two required labels; some rule of precedence must be specified to handle such situations.

DEFINE: APPLE;

CLEARANCES: APPLE;

SYNONYMS: NONE;

INTERNAL STRUCTURE: NONE;

ACCESS RULES: APPLE ACCESSES ALICE;

REQUIRED LABELS: HANDLE VIA APPLE CHANNELS ONLY;

EXTERNAL STRUCTURE: NONE;

REQUIREMENTS: APPLE REQUIRES III;

MERGE RULES: NONE;

END;

Example 4

Consider a hypothetical example (named ROUND ROBIN) in which it is assumed that at the Secret level there are two categories of information, called

AGILE and BANANA, accessing information labelled respectively as ANN and BETTY. Further assume that an individual cannot be concurrently authorized access to both AGILE and BANANA information. Rather, assume that in order to have access to both, an individual must be cleared to Top Secret, in which case he will be said to have access to CHERRY information labelled CHICO, as well as to all AGILE and BANANA information. Furthermore, assume that having a CHERRY access also allows an individual to access all information that a person who has a III access authorization (see Example 2) may access.

DEFINE: ROUND ROBIN;

CLEARANCES: CHERRY, AGILE, BANANA;

SYNONYMS: NONE;

INTERNAL STRUCTURE: CHERRY IMPLIES AGILE, CHERRY IMPLIES BANANA;

ACCESS RULES: CHERRY ACCESSES CHICO, AGILE ACCESSES ANN, BANANA ACCESSES BETTY;

REQUIRED LABELS: NONE;

EXTERNAL STRUCTURE: CHERRY IMPLIES III;

REQUIREMENTS: AGILE REQUIRES NOT BANANA AND SECRET, BANANA REQUIRES NOT AGILE AND SECRET, CHERRY REQUIRES TOP SECRET;

MERGE RULES: ANN AND BETTY YIELDS TOP SECRET AND CHICO;

END;