

# **ELECTRONIC SYSTEMS DIVISION AIR FORCE SYSTEMS COMMAND**

**HANSCOM AIR FORCE BASE, MASSACHUSETTS**



MCI-75-1

December 1974

## **ESD 1974 COMPUTER SECURITY DEVELOPMENTS SUMMARY**

Approved for public  
release; distribution  
unlimited

**INFORMATION SYSTEMS TECHNOLOGY APPLICATIONS OFFICE  
DEPUTY FOR COMMAND AND MANAGEMENT SYSTEMS**

**20100827136**

LEGAL NOTICE

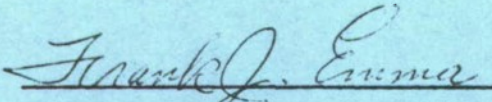
When U.S. Government drawings, specifications or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

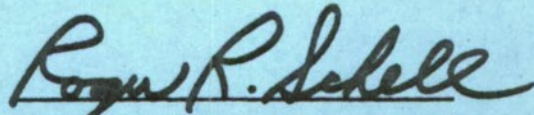
OTHER NOTICES

Do not return this copy. Retain or destroy.

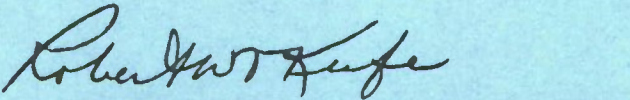
REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.

  
FRANK J. EMMA, Colonel, USAF  
Chief, Techniques Engineering  
Division

  
ROGER R. SCHELL, Major, USAF  
Techniques Engineering Division

FOR THE COMMANDER

  
ROBERT W. O'KEEFE, Colonel, USAF  
Director, Information Systems  
Technology Applications Office  
Deputy for Command & Management Systems

| REPORT DOCUMENTATION PAGE                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                               | READ INSTRUCTIONS<br>BEFORE COMPLETING FORM                                                              |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------|----------------------------------------------------------------------------------------------------------|
| 1. REPORT NUMBER                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 2. GOVT ACCESSION NO.<br>None | 3. RECIPIENT'S CATALOG NUMBER                                                                            |
| 4. TITLE (and Subtitle)<br><br>ESD 1974 COMPUTER SECURITY DEVELOPMENT SUMMARY                                                                                                                                                                                                                                                                                                                                                                                                                                |                               | 5. TYPE OF REPORT & PERIOD COVERED<br><br>Interim Report                                                 |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                               | 6. PERFORMING ORG. REPORT NUMBER<br>MCI 75-1                                                             |
| 7. AUTHOR(s)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                               | 8. CONTRACT OR GRANT NUMBER(s)                                                                           |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Deputy for Command & Management Systems (MCI)<br>Electronic Systems Division (AFSC)<br>L. G. Hanscom AFB, Bedford, MA 01731                                                                                                                                                                                                                                                                                                                                   |                               | 10. PROGRAM ELEMENT, PROJECT, TASK<br>AREA & WORK UNIT NUMBERS<br><br>PE 63728F/Project 5550/<br>Task 09 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br><br>See Item 9                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                               | 12. REPORT DATE<br>31 December 1974                                                                      |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                               | 13. NUMBER OF PAGES<br>34                                                                                |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)                                                                                                                                                                                                                                                                                                                                                                                                                                  |                               | 15. SECURITY CLASS. (of this report)<br><br>UNCLASSIFIED                                                 |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                               | 15a. DECLASSIFICATION/DOWNGRADING<br>SCHEDULE                                                            |
| 16. DISTRIBUTION STATEMENT (of this Report)<br><br>Approved for public release; distribution unlimited.                                                                                                                                                                                                                                                                                                                                                                                                      |                               |                                                                                                          |
| 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)                                                                                                                                                                                                                                                                                                                                                                                                                   |                               |                                                                                                          |
| 18. SUPPLEMENTARY NOTES The material presented was in large part originally prepared by Mr. Steven B. Lipner and Mr. Richard D. Rhode of the MITRE Corporation, Bedford, Massachusetts, in support of Project 5720 under Contract No. F19628-75-C-0001. Since this material is of an interim nature, this document does not qualify as an ESD Technical Report, and it is NOT AVAILABLE THROUGH DDC.                                                                                                         |                               |                                                                                                          |
| 19. KEY WORDS (Continue on reverse side if necessary and identify by block number)<br><br>Secure Computer Systems<br>Multilevel Systems                                                                                                                                                                                                                                                                                                                                                                      |                               |                                                                                                          |
| 20. ABSTRACT (Continue on reverse side if necessary and identify by block number)<br>This document presents a summary of ESD computer security development efforts that are currently underway or have been proposed. The purpose is to portray the nature of the relevant problems and viable technical approaches for their solution. The efforts described do not necessarily reflect approved Air Force development projects; therefore, details of resource allocations and schedules are not provided. |                               |                                                                                                          |

## SECTION I

### INTRODUCTION

This document describes a program intended to provide Air Force ADP users with the ability to process classified information securely and economically in computer systems. The lack of such an ability in today's systems has resulted in procedural "fixes" that generate significant costs and fail to address major operational requirements.

The document begins with an overview of the technical problem of computer security and of the Air Force user requirements that make this problem an important one. It then outlines a unified technical approach to solving computer security problems, and goes on to summarize major ESD-sponsored developments that use this approach. The final section summarizes the individual tasks that make up the ESD development program.

## SECTION II

### COMPUTER SECURITY PROBLEMS AND REQUIREMENTS

#### CURRENT ADP SECURITY PRACTICE

The problem of multilevel security in Automatic Data Processing (ADP) can best be introduced by discussing alternatives to technology solutions. These alternatives take the form of procedures that current ADP systems use for processing classified information. These procedures normally permit only one security level (1) of information to be processed at a time. The ADP system is housed in a facility cleared for a single security level, and access to it is restricted to appropriately cleared individuals. If remote users must be supported by the ADP system, the personnel at the remote site(s) must also be cleared and their terminals housed in secure areas. In addition, the remote terminals and central facility must be linked by encrypted or protected communications circuits.

If a present-day system has to process several levels of classified information, there are two alternatives:

- a. all security levels may be processed together -- provided that all users (and terminal areas and communications) are cleared for the highest level of information that could be processed on the system; or
- b. each level may be processed at a separate time, in which case the entire system environment (terminals, disk packs, tapes, printers) must be changed or sanitized at each change of security level.

The first alternative results in a proliferation of personnel clearances, secure terminal areas, and secure communications. The second, called "color-changing", does not. Even an uncleared terminal may be served provided it is detached before classified processing begins. But each change of level wastes a significant amount of system time while completing the change of environments. Regardless of which alternative is employed, the procedures necessary today to process multiple levels of classified information with computer systems involve increased cost, inconvenience, and/or system inefficiency.

---

(1) The terms "security level" and "level of information" are used here to designate a single National Defense Security classification level (Confidential, Secret, etc.) and one set of compartments (formal need-to-know classes).

## COMPUTER SECURITY REQUIREMENTS

This subsection summarizes the computer security requirements of some major Air Force ADP users. While it is not exhaustive, it does indicate the major problems that have been encountered to date with the use of current non-technology alternatives. Trends in future problems and requirements can be inferred from these experiences. The impacts of computer security requirements on system costs and on operational capabilities are stressed.

It should be noted in this introduction that computer security requirements have not yet made themselves apparent by the occurrence of hostile penetrations directed against computers processing classified data. The reason for this lack is not that such penetrations are impossible, but that current policies dictate the operation of computers in the modes described above that preclude such penetrations. Recent policy modifications have offered Air Force ADP managers the option of weakening these restrictions, but most installations have declined to implement the modifications, believing them inconsistent with their responsibilities for protecting classified information.

The following paragraphs address the impact of current alternatives for meeting computer security requirements on system costs and on operational capabilities.

### Cost Impacts

The cost impacts of computer security have been reflected in expenditures for increased protection and additional equipment, and in inefficient system utilization. Typical of the installations that have required increased protection is the Air Force Data Services Center at the Pentagon. There, additional personnel clearances, vault areas, and secure communications have been required to allow users to do unclassified processing on computers that handle classified data. The cost of securing each remote site (excluding terminal equipment) is estimated by AFDSC at \$50,000. At SAC, additional SIOP clearances and area protection were required when it was decided that the 4000th Aerospace Applications Group was to receive its computer support from the SAC World Wide Military Command Control System (WWMCCS) ADPE.

Additional equipment has been required by computer installations that must provide responsive support to user communities of varied clearance levels. At AFDSC, a time-sharing system (a Honeywell 635) was acquired to provide unclassified computing services to AFDSC's users in open office areas, supplementing the classified processing systems (with secure remote terminals) mentioned above. One of the

two SAC WWMCCS dual processors was split into two single processor systems so that development, on-line support and planning applications, each of differing security level, could each have their own computers. An additional Honeywell 6080 WWMCCS processor is now to be installed at MAC, to satisfy MAC's need to provide timely support to classified crisis management applications. This added equipment costs approximately \$4 million (an estimated \$2 million for the 635 and \$1 million each for the dual processor split and additional 6080). Additional Air Force WWMCCS (and other) computer facilities can be expected to require similar additions of equipment as major classified processing applications become operational.

Inefficient equipment utilization is reflected in the phenomenon of classified processing systems known as the "color change". In a color change, all work of one security level is completed, print queues are drained, and media dismounted. Then system memories are cleared, new media (including the operating system residence) are mounted, and a version of the system is brought up to process the new level. The actual time required to perform the change of media and clear and restart the system ranges from twenty to forty-five minutes. The color change may be propagated over one to two hours' processing by the refusal of long jobs and by the saving of files on backup tapes. Color changes are usually used in cases where responsiveness and workload do not require dedication of a computer to a given level. Thus SAC, with its many WWMCCS computers, performs several color changes each day. MAC and the SAC intelligence computer (a 360/85) also perform color changes, and so do smaller Air Force WWMCCS installations. These changes can easily absorb ten to twenty per cent of a system's processing capacity. (2)

#### Operational Impacts

Where possible, operational requirements for secure computers are met either by adding equipment so that there is a computer for each required level, or by clearing all users for access to all information processed. There is, however, a significant class of operational requirements that cannot be satisfied by today's computer systems using these alternatives.

For example, during the 1973 Middle East War, MAC was required to transport military supplies and equipment into Israel. Because of the sensitive nature of the operation, its details were classified Secret. Because of the operation's classification, it was impossible to support, at the same time with available equipment, both operation of

---

(2) Based on current examples where the system is in use ten hours a day, there are two color changes at 1/2 hour each, and there is 50% system degradation for an hour before each change.

normal unclassified command functions and operation of the contingency management functions. A small portion of the flight-following data base became classified and this portion had to be processed manually to avoid contaminating the entire data base. Addition of a processor at MAC has eliminated the requirement for manual processing of classified information, but manual re-entry and integration of information are still necessary. Consequently, even though additional equipment is available, MAC lacks an integrated system for the management of its aircraft force.

A second class of operational requirement concerns the integration of intelligence and operations data. Such integration is required for responsive force management, but it must be done so as not to jeopardize intelligence sources. In this case, it is often impossible to clear all system users for the intelligence data, so manual intervention is used -- a cleared intelligence officer hands a subset of the data to the operations element. As automated, timely integration of such data becomes necessary, this option becomes unacceptable, and a direct technological solution to the multilevel security problem is required.

#### Requirements Summary

What has been said summarizes the major impacts of current alternatives for meeting the requirement for computer security. Experience has indicated that the cost may run to ten to twenty percent or more of the total operating cost of the Air Force computer installations that process classified data -- perhaps \$20 to \$40 million per year. Operationally, many requirements are met by buying additional equipment and facilities, but a significant requirement for real-time information sharing is arising and this requirement cannot be met even by buying such equipment.

#### THE TECHNICAL PROBLEM OF MULTILEVEL SECURITY

The case against relying on the costly, restrictive procedures outlined above is strong. Economic and operational considerations argue for developing the ability to process an arbitrary mix of classified and unclassified information simultaneously with a single computer, serving cleared and uncleared users and relying on the computer's and operating system's internal controls to enforce security and need-to-know requirements. Such a computer would be operating in a multilevel security mode; the presence of uncleared users (or users at unsecured terminals) would define an open multilevel mode. Unfortunately, the costly procedures used today continue to be necessary -- made so by the inability of current hardware-software systems to protect the information they process. The only sound assumption that can be made about a current computer

system concerning information protection is that any program that runs on the system can access any information physically accessible to the processor, and can retrieve, alter or destroy the information as the programmer wishes.

While the assumption stated above may appear radical, it is amply supported by facts and experience. On numerous occasions, programmers have conducted formal or informal projects aimed at testing the security of operating systems by penetration -- by writing programs that obtain access to information without authorization. ESD personnel have directly participated in several of these penetration projects and have observed the results of others. In each case, the result has been total success for the penetrators. The programmers involved in these efforts have not been "insiders" but simply competent system programmers armed with user and (sometimes) system level documentation for the computer and operating system under test.

No "real" hostile penetrations of military computers processing classified information have been reported. However, this is because such computers operate under protective procedures of the sort just described, not because it is difficult to make a programmed penetration against them.

Given experience in the penetration of computer systems, one might ask "why not simply modify the operating system programs to correct the flaws that permit the penetration?". Two problems prevent this approach (often referred to as "patching holes") from being effective. The first is that in many cases operating system or application programs will not work if a hole is patched. Thus, correcting a security flaw may render the computer system inoperative unless a long, costly series of program modifications is made. This problem is compounded at the practical level by the fact that complex, expensive program modifications intended to patch existing operating system holes, may themselves introduce new holes in previously sound areas.

The second problem, a fundamental one in the field of multilevel computer security, is that of completeness. Even if every hole that allows a known penetration approach to work were repaired, one still could not consider the resulting operating system secure, because a given collection of penetration programs exposes only the holes that those programs exploit. Short of constructing the (astronomically large) set of all possible penetration programs, one can make no statement at all about undiscovered holes, or about the penetration programs that would exploit them.

The problem of completeness, as stated above, might lead the reader to rebel and proclaim that completeness is not necessary. Nowhere else is perfect security required; physical, personnel and

even communications security measures have finite probabilities of penetration. Is it not then possible to accept a degree of computer security less than a hundred per cent? Unfortunately, the usual analogy between operating system security problems and those of physical, personnel or communications systems is not a correct one. If even one error in an operating system program allows a penetration program to work, that program will work every time it is executed -- typically retrieving without detection any information accessible to the computer. The probability of a successful penetration is then unity; the level of security, zero. The likelihood that a hostile agent will write the penetration program is the only uncertainty. This likelihood is hard to assess, since it depends on the agent's motivation and competence. However, experience with penetration tests leads to the conclusion that the penetrator's chances of success are very high.

Restricting access to operating system documentation is not a safeguard. Although concealing the structure of the operating system may seem to obscure the weaknesses of the security controls, such a primitive encoding scheme does not effectively deter penetration; knowledge of the basic processor hardware and any standard operating system provides an adequate starting point for the penetrator's efforts.

A final point about the vulnerability of current computer systems concerns the cost of penetration. Most penetration efforts have been completed successfully with very few (perhaps two) man-months of effort. Typically, the bulk of the effort expended is directed toward exploitation -- finding information to be retrieved and building programs to retrieve it. Development of the basic approaches that assure successful penetration has usually required only a man-week or two. In comparison, the effort expended in patching operating system holes is rumored (3) to be in the tens or hundreds of man-months.

This brief overview of the technical problem of multilevel computer security is not intended to portray the problem as hopeless. Rather, the intention is to show that the problem is difficult and that the alternative of patching holes in current operating systems is futile. The next section introduces a unified technical approach to the development of secure computer systems.

---

(3) Most agencies that have performed such patches are reluctant to report costs.

## SECTION III

### A UNIFIED TECHNICAL APPROACH TO MULTILEVEL COMPUTER SECURITY

#### INTRODUCTION

This section introduces the foundation of the computer security development effort. Its three subsections describe the history and origin of the technical approach; briefly summarize the approach and its main implications; and discuss the technique for verifying the security of a computer system that solves the problem of completeness.

#### THE COMPUTER SECURITY TECHNOLOGY PANEL

In 1970, the Air Force Data Services Center (AFDSC) asked the Electronic Systems Division to support development of open multilevel secure operation for AFDSC's Honeywell 635 computer systems. The 635's operate under control of the standard GCOS III operating system. ESD and MITRE personnel shortly reached conclusions substantially identical to those given above: that no set of modifications to GCOS III would render it suitable for multilevel operation, much less for open operation with uncleared users and terminals.

To determine the reasons for the difficulty with GCOS III, and to identify ways of solving future multilevel security problems, the Air Staff directed ESD to convene a computer security technology planning study panel. The panel, composed of recognized experts from industry, universities, and government organizations, convened in early 1972. The panel operated under a contract from ESD to James P. Anderson and Company, and was tasked with preparing a development plan for a coherent approach to attacking the problems of multilevel computer security. The panel was supported by a working group of computer system staff officers from ten Air Force commands who identified the operational and economic impacts resulting from the lack of computer security technology.

The panel's report (4) described an earlier version of the development effort described here. Further, it identified the problem of completeness and recognized the futility of "patching holes" in existing operating systems. It recommended as a technical approach "to start with a statement of an ideal system, a model, and to refine and move the statement through various levels of design into the

---

(4) James P. Anderson, Computer Security Technology Planning Study, ESD-TR-73-51, October 1972.

mechanisms that implement the model system". (5)

#### THE REFERENCE MONITOR

The basic component of the ideal system proposed by the security technology panel is the reference monitor -- an abstract mechanism that controls access of subjects (active system elements) to objects (units of information) within the computer system. Figure I schematically diagrams the relationships among the subjects, objects, reference monitor, and reference monitor authorization data base. The figure gives examples of typical subjects, objects and data base items.

An implementation of the reference monitor abstraction permits or prevents access by subjects to objects, making its decisions on the basis of subject identity, object identity, and security parameters of the subject and object. The implementation both mechanizes the access rules of the military security system and assures that they are enforced within the computer.

The security technology panel stated that, in order to provide the basis for a multilevel secure computer system, a mechanism that implements a reference monitor must meet three requirements:

- a. Completeness -- the mechanism must be invoked on every access by a subject to an object.
- b. Isolation -- the mechanism and its data base must be protected from unauthorized alteration. .
- c. Verifiability -- the mechanism must be small, simple and understandable so that it can be completely tested and verified to perform its functions properly.

The requirements for completeness and verifiability demand that the reference monitor implementation include hardware as well as software -- the former because software validation of every access by a subject to an object would add intolerable complexity and overhead to the reference monitor, the latter because certain hardware architectures preclude construction of a simple, understandable operating system.

The panel recognized the importance of hardware architectures and

---

(5) Op. cit., Volume 1, p. iv.

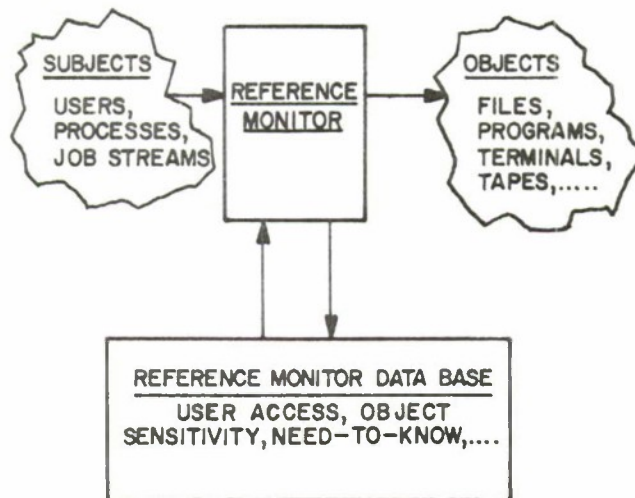


Figure 1. REFERENCE MONITOR

recommended for secure computer systems the use of descriptor-driven (6) processors that implement segmented memories. With such processors, the objects of the reference monitor can correspond to the segments supported by the hardware. A properly organized segmented memory merges primary (core) and secondary storage management functions, eliminating from security consideration any complex "file system". Further, the subjects of the reference monitor correspond to processes (address space-processor state pairs) supported directly by a descriptor-driven processor.

The hardware-software mechanism that implements the reference monitor abstraction is called the security kernel. When the computer hardware is predetermined, the software that must be designed to implement the reference monitor abstraction is frequently referred to as the security kernel for that computer. The paragraphs below discuss the problems of designing a security kernel and validating its effectiveness.

#### MODELS AND TECHNICAL VALIDATION

Recognizing the importance of the panel's "ideal model" as a starting point, ESD initiated development of a mathematical model of computer security in 1972. Preliminary efforts were performed in-house (7) and subsequent contributions were made by The MITRE Corporation and by Case Western Reserve University.

The completed model of secure computer systems (8) represents a secure computer system as a finite-state mechanism that makes explicit transitions from one secure state to the next. The state of the system is defined by:

- a. the classifications and compartments of all subjects and objects;

---

(6) A descriptor-driven processor is one whose hardware interprets each "virtual" address issued by a program in terms of a set of descriptors that specify the real physical address and permitted access modes (e.g., read, write, execute) to be associated with every possible "virtual" address.

(7) R. Schell, P. Downey, G. Popek, Preliminary Notes on the Design of a Secure Military Computer System, MCI-73-1, January 1972.

(8) D. E. Bell and L. J. LaPadula, Secure Computer Systems, ESD-TR-73-278, Vol. I-III, The MITRE Corporation, Bedford, Massachusetts.

- b. the need-to-know relationships of subjects and objects;
- c. the hierarchical organization of objects (in a storage system); and
- d. subjects' current ability to access objects.

The rules of the model formally define the conditions under which a transition from state to state can occur. The rules are proven to allow only transitions that preserve the security of information in the system.

A significant property of the model is that all but a special collection of proven and trusted programs are restricted from writing information at a lower classification (or proper subset of compartments) than they read. The restriction prevents information obtained at the higher level from being transferred to a lower level where it can be accessed illegally. This property eliminates the need to verify that all programs (such as editors and utility routines) do not act as "Trojan Horses" (9) and downgrade classified information.

The model of secure computer systems specifies requirements for the operation of a security kernel. The requirements identified by the model are taken directly from the Defense Department regulations on handling sensitive information (DoD Directive 5200.1-R). The problem of validation is then reduced to providing complete assurance that the security kernel behaves as the model requires.

For some time after the basic security model was developed, there was doubt as to the appropriate technical approach to providing the assurance mentioned above. In 1973 it was recognized that the work of Price (10) identifies a methodology for providing the required assurance. This methodology involves preparing a formal (or Parnas) specification for each function of the security kernel. The collection of specifications is then proven to be internally consistent and to implement the rules of the model. The descriptions

---

(9) A Trojan Horse is a computer program that is typically developed by one individual for use by another. When the program is operating on behalf of the intended user, it accesses that user's sensitive data, then makes it available to the program's author (for example by writing it in a "hidden" file). See D. K. Branstad, "Privacy and Protection in Operating Systems", Computer, Vol. 6, No. 1, January 1973.

(10) W. R. Price, Implications of a Virtual Memory Mechanism for Implementing Protection in a Family of Operating Systems, PhD Thesis, Carnegie-Mellon University, June 1973.

of the functions in the specification language are close to a programming language and facilitate proof or verification of the code that implements the specified kernel design. A more detailed description of the validation methodology has been prepared by MITRE and is contained in (11) and (12) .

While the basic methodology developed by Price applies to validation of small security kernels (up to perhaps 1000-line computer programs), the consistency proof may become cumbersome for larger kernels. Therefore, a structured specification and proof technique that divides the specification modules into manageable subsets is being explored in addition to the basic Price methodology. (13)

The paragraphs above have summarized the basic elements of ESD's approach to the design and technical validation of secure computer systems and security kernels. While the administrative certification that a computer is secure must be based on formal policy, it is likely that a technical validation approach such as that outlined provides the only adequate basis for such formal certification.

---

(11) E. L. Burke, Synthesis of a Software Security System, MTP-154, The MITRE Corporation, Bedford, Massachusetts, August 1974.

(12) D. E. Bell, E. L. Burke, A Software Validation Technique for Certification: The Method, ESD-TR-75-54, The MITRE Corporation, Bedford, Massachusetts, November 1974.

(13) L. Robinson, P. G. Neumann, K. N. Levitt, and A. Saxena, "On Attaining Reliable Software for a Secure Operating System", to appear in 1975 International Conference on Reliable Software, Los Angeles, California, 21-23 April 1975.

## SECTION IV

### SECURE COMPUTER SYSTEMS

#### INTRODUCTION

This section presents an overview of four major secure computer system developments that apply the technical approach described above. They are aimed at providing the Air Force with immediate improvements in its ability to meet computer security requirements, and with long-term solutions to very general computer security requirements. The first is the "brassboard security kernel", a general-purpose security kernel for an off-the-shelf minicomputer. The next is the jobstream separator, a mechanism that provides a reference monitor external to a large unsecure computer system. The third is the development of a security kernel for Multics, a large general-purpose computer system. Finally, there is a description of some developments and applications in the areas of secure computer networks and secure communications processors.

#### THE BRASSBOARD SECURITY KERNEL

The computer architecture requirements in the previous section are stated in terms of required features, rather than specific computer types. The reader may have the impression that computer security requires development of special "military" processors with the requisite features. Fortunately, that is not the case. Several manufacturers make small-, medium- and large-scale computers that have the required memory segmentation, as well as multiple domain hardware that can be used to isolate a kernel, a non-security operating system, and user programs from each other. Furthermore, current trends towards virtual memory and reliable programs make it likely that the future systems developed by other manufacturers will also have these features.

One processor that is available now and is suitable for use in a multilevel secure computer system is the Digital Equipment Corporation PDP-11/45, a relatively large, moderately priced minicomputer that can optionally be outfitted with hardware to implement segmentation and domains. To verify the viability of the secure system model, ESD directed MITRE Corporation in January 1973 to begin implementing a prototype security kernel for the PDP-11/45. (14) This kernel was

---

(14) W. L. Schiller, Design of a Security Kernel for the PDP-11/45, ESD-TR-73-294, The MITRE Corporation, Bedford, Massachusetts, June 1973.

initially intended to serve as the base for a front-end communications processor for use with a secure general-purpose computer system to be developed later. It was soon realized that the kernel could also support stand-alone secure computer applications requiring only a minicomputer and, most important, could serve as a "brassboard" to prove out the model and kernel concepts long before developing a kernel for a large general-purpose system. (15)

The kernel design for the PDP-11/45 was developed by applying Dijkstra's levels of abstraction (16) to separate the parts of the kernel that implement the security rules, objects and subjects required by the model. The kernel design has gone through one major revision due to an increase in the model's representational power. The revision simplified the kernel, increased the functional utility of the environment provided, and entirely eliminated several security problems that had previously been handled on an ad hoc basis. (17)

The kernel design provides for a potentially very large segment storage system with a hierarchical organization. The kernel implements separate sequential processes that can cooperate and communicate in accordance with the rules of the model. (Formally, interprocess communication channels are treated as objects and constrained by the security rules.) The kernel itself manages direct-access (disk and drum) storage and magnetic tape. Handling of terminals and other low-speed input/output devices is delegated directly to user (or non-security operating system) programs. Provision of much input/output control by programs outside the kernel is possible because the PDP-11/45 segment control hardware allows the kernel to allocate input/output devices to processes as it does segments. (18) The PDP-11/45 kernel design is a foundation for

---

(15) While a kernel for a large, general-purpose computer need not be much larger than that for a minicomputer, the amount of non-security operating system software needed to effectively use the large system is far greater.

(16) E. W. Dijkstra, "The Structure of the THE Multiprogramming System", Communications of the ACM, Volume II, Number 5, May 1968.

(17) An example is the elimination of the security problems of input/output (I/O) operations. The revision treats external I/O devices simply as additional security kernel objects. User programs can then execute I/O operations with the same security protection afforded other operations (such as the "add" instruction).

(18) The segment control hardware has a few limitations in allocation of input/output devices; if they were not present, magnetic tape too could be handled outside the kernel.

operating systems and application programs that will function in a secure environment.

The design of the revised security kernel for the PDP-11/45 was completed in early 1974. The kernel programs were implemented in a higher-order language (the Project SUE Systems Language) and compiled and tested in spring 1974. Since then development of application and demonstration programs that exploit the kernel has proceeded at a moderate pace.

Verification of the brassboard kernel's security began in summer 1974 with completion of formal specifications. Proof that the specifications are consistent and implement the model was initiated then, and a single module was proven to verify the feasibility of the proof method. Proof of the remaining modules was then deferred, and reinitiated in late 1974. Exhaustive mechanized testing of the kernel programs against their formal specifications began in fall 1974; verification of the methodology again preceded complete testing.

In summary, the PDP-11/45 security kernel provides an early demonstration of the feasibility of building a security kernel that implements the model. Each step in the sequence from model to kernel code is subject to proof or verification. The final kernel will be available for performance tests, penetration tests (which will undoubtedly be desired even though their failure is not a proof of security), inspection, review, and application.

#### THE JOBSTREAM SEPARATOR

The basic principles identified above have led to an understanding of appropriate ways of achieving secure computer systems. Unfortunately, these ways do not apply to most existing computers; the hardware is simply wrong for development of a reference monitor. However, some security solution is clearly desirable for a large number of installed systems that bear the economic burden imposed by today's alternatives to security technology.

The jobstream separator provides a reference monitor outside the main computer. The objects managed by the reference monitor are physical objects that can be controlled from without -- objects such as disk and tape drives and communications circuits. The subjects are entire job streams of uniform security level. The jobstream separator makes the complete change of environments used in current color-changing procedures, but automates it under the control of a secure minicomputer. The main computer exercises no programmed security control function.

Figure 2 shows the configuration of a computer installation that uses a jobstream separator. The different levels of information are segregated on separate storage devices (for example, Secret on devices marked S and Unclassified on devices marked U). In operation at the Unclassified level, the minicomputer closes switches to the Unclassified disk and tape drives, and opens the switches to all other drives. It also forwards information to and from the Unclassified communication lines and blocks flow on those of Secret level. When the main computer is to serve the Secret job stream, the minicomputer signals the main processor to "shut down". The main processor stops its jobs and saves the state of memory, processor and control units on an Unclassified storage device. (The save process may be complicated, but it is not security-related -- the main processor program that does the stopping and saving can access only Unclassified devices, and thus cannot fail so as to compromise information.) The minicomputer stops forwarding Unclassified communications and opens the switches to the Unclassified drives. It then attaches a read-only "clear" bootload tape (B in Figure 2) to the tape controller and sends a signal to the processor's bootload control line. The clear program -- the only security-related main processor program -- initializes the main processor, memory and control units to a neutral state. Once the main processor is cleared, the minicomputer opens the switch to the clear tape, closes switches to the Secret drives, and initiates another bootload, this time from a drive containing a previously saved Secret system state. The minicomputer then begins forwarding characters to and from the Secret communications circuits.

The minicomputer, drive access switches and bootload program of Figure 2 implement a reference monitor as defined in the previous section. Every access to information by the jobstream in the main processor is mediated by the minicomputer-controlled switches or by the minicomputer itself. Objects that appear at all security levels and use common physical resources -- the processor and memory -- are explicitly controlled by the minicomputer via the bootload control line and clear program. The controls in the minicomputer are isolated and protected from the programs in the main processor. Finally, use of a secure minicomputer with a security kernel (such as a PDP-11/45) allows the implementation to meet the requirement of verifiability.

While the jobstream separator concept provides an automated secure computer system, it does not provide multilevel security. Several jobs of differing levels cannot be multiprogrammed together, and files cannot easily be shared across security levels. Further, the cost of the jobstream separator configuration (minicomputer, switches and extra storage drives) must be balanced against the time that the configuration saves. Thus the jobstream separator configuration must be viewed as an alternative to manual color-changing in tradeoff studies.

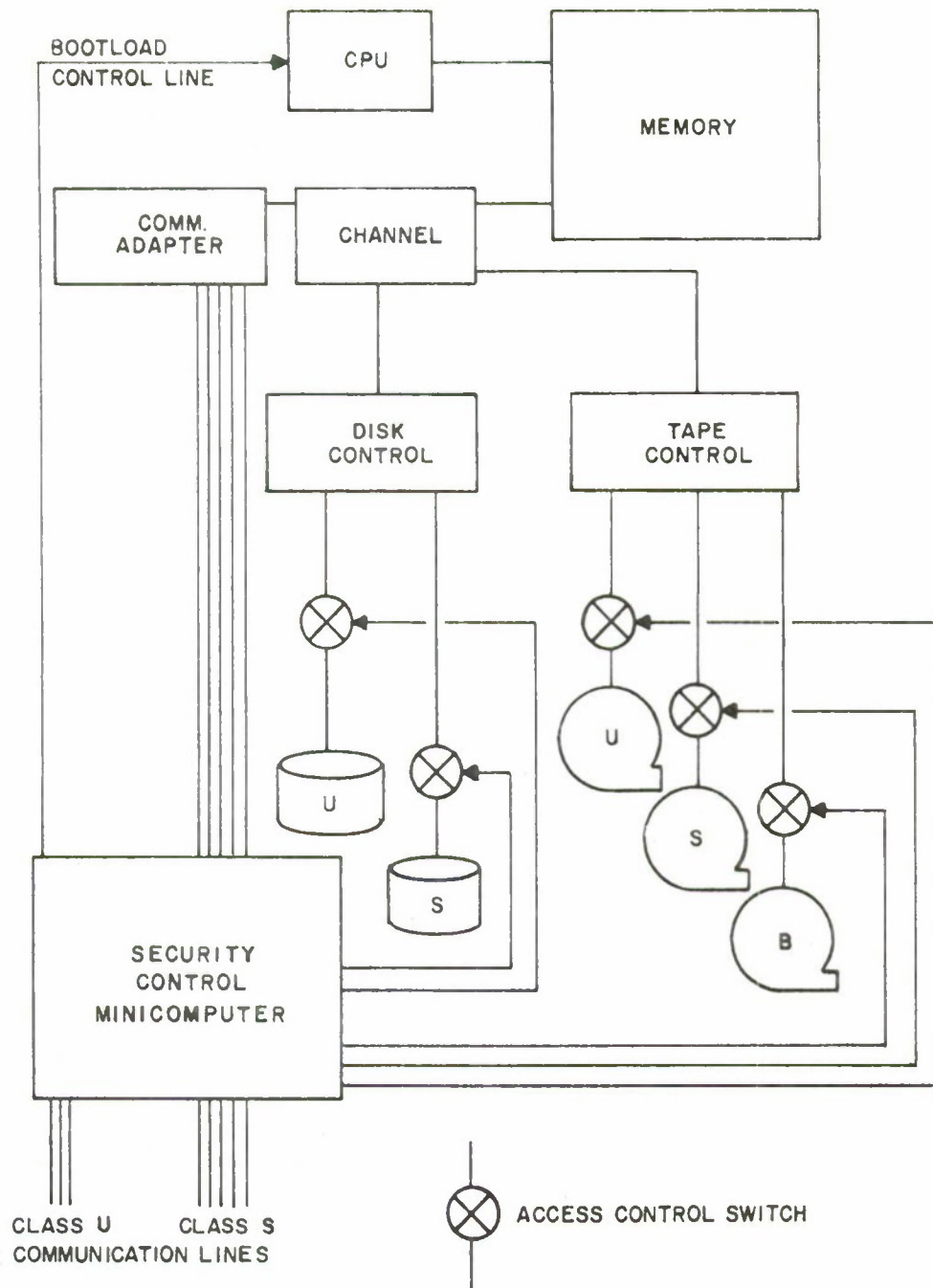


Figure 2. JOBSTREAM SEPARATOR

Although the jobstream separator concept was first devised in late 1970, it was not until late 1973 that it was seen to be an implementation of a reference monitor. (19) In mid-1974, at the direction of the Directorate of Data Automation at Air Force Headquarters, ESD began a tradeoff study of the sort mentioned above, with the specific objective of determining the costs, benefits and feasibility of applying the jobstream separator concept to the Air Force WWMCCS ADPE. Requirements for such a system have been identified and engineering issues examined. A report on the jobstream separator tradeoff study is to be published in early 1975, and development of a prototype could begin in spring 1975.

#### THE MULTICS SECURITY KERNEL

While the security kernel for the PDP-11/45 provides a small secure system, and the jobstream separator permits more efficient secure use of current ADPE, Air Force commands such as MAC and AFDSC need large multilevel secure computers. Furthermore, the reference monitor concept will be much more useful if it can be demonstrated in an efficient, as well as secure, large resource-sharing system. For these reasons, ESD has set as a goal development of a security kernel and operating system for a large computer.

The Honeywell 6180 (or successor 68/80) computer and its Multics operating system were chosen as the base for a secure large-scale prototype, for two prime reasons. First, the 6180 hardware supports a segmented virtual memory and multiple protection domains in a way that makes it well-suited to support a kernel. In fact, a study of hardware architectures for security completed in mid-1974 (20) determined that the 6180 was the off-the-shelf large computer best suited to support a security kernel.

The second reason for choosing the 6180 and Multics relates to the Multics operating system. Multics is written to implement a segmented virtual memory, and to use that segmented memory where possible within the operating system. Thus the existing user programs and many operating system programs are compatible with the environment that a security kernel is expected to provide. This fact should significantly reduce the cost of a Multics-based secure system, for it appears that the (non-security related) operating system software, rather than the security kernel, will be the major cost component in

---

(19) S. B. Lipner, A Minicomputer Security Control System, MTP-151, The MITRE Corporation, Bedford, Massachusetts, February 1974.

(20) L. Smith, Architectures for Secure Computing Systems, ESD-TR-75-51, The MITRE Corporation, Bedford, Massachusetts, 30 June 1974.

any kernel-based secure computer system.

Initial steps toward developing a secure system based on Multics were taken in conjunction with development of a Multics operating system for use in a two-level (Secret and Top Secret) environment at the Air Force Data Services Center. This system's design is aimed at providing security controls based on the military access rules, but it does not attempt to eliminate completely the prospect of hostile penetration. The risk of penetration is largely to be controlled by procedures and by personnel and environmental controls, rather than by the Multics hardware and software. The implementation of the access rules in the Data Services Center Multics was based on the secure system model described in the previous section, but no attempt was made to define a security kernel for the system.

The design of the Data Services Center Multics was begun in late 1973 and completed in mid-1974. Implementation is to be finished and the system in operation by mid-1975. During design and implementation of the Data Services Center Multics, a number of issues arose pertaining to the system's utility and security. The resolution of these issues provided information relevant to the design of a Multics security kernel. Furthermore, the user interface of the Data Services Center Multics has been designed to resemble that of a kernel-based system, so that the transition from the Data Services Center Multics to a kernel-based Multics will be relatively easy, and so that operational experience will be available to guide the Multics kernel design.

Design of a Multics kernel began in September 1974 with a concentrated one-month session involving staff members from ESD, the MITRE Corporation, Honeywell and the Massachusetts Institute of Technology (a codeveloper of Multics). The resulting kernel design was presented in rough detail to a meeting of government and Multics design personnel in early November 1974. It includes a segmented and paged virtual memory similar to that of the standard Multics operating system, with security controls and organization similar to those in the PDP-11/45 "brassboard" kernel. The input/output system required by the kernel is based on using a minicomputer front-end processor with its own kernel to provide a secure flexible interface to external devices. This approach allows for secure input/output control without requiring the (radical) development of a non-programmable secure input/output controller.

MITRE members of the kernel design team are preparing formal specifications for the kernel that are to be completed in spring of 1975. Honeywell has been involved in the kernel design since July 1974, via a cost-sharing contract with ESD. With MIT as a subcontractor, Honeywell is defining the revisions to the (non-security) Multics operating system that will provide a complete,

usable environment.

## SECURE NETWORKS AND COMMUNICATIONS PROCESSORS

The results achieved thus far by ESD's computer security development programs apply to a wide variety of secure computer systems. However, to be of value to the Air Force, these general results must be translated into specifications for competitive acquisition of secure systems. The first paragraph below discusses progress in this direction. The remaining paragraphs discuss potential application of the secure system developments in the area of computer network security.

The SATIN IV communications network for the Strategic Air Command requires secure communications processors. During 1974 the secure computer technology described above has been translated into acquisition documents for the SATIN IV communications processors. The key to this translation lies in the recognition that the secure system model is fundamental to the Defense Department security system, and that the formal specification and programs may vary depending on system functions and choice of hardware. Verification of system security depends on expression of the secure system design in proper formal specification language. Thus SATIN IV requires that the communications processor internal access control mechanism be described by a formal specification and proven to correspond to the security model. It must then be verified that the computer programs correspond to the formal specifications.

Besides having an internal access control mechanism, each communications processor in the SATIN IV network must be able to determine the security level of information that it receives and transmits. Thus the SATIN IV security efforts have also emphasized the need for a secure path by which kernels can communicate security control information. The specifics of such a path depend on network protocols and functions.

An alternative to providing network security by secure communications processors is the use of end-to-end encryption. In this case, information is handled in enciphered form within the network, with encryption before entry and decryption at the destination. Most end-to-end encryption schemes require use of a control computer to direct the operation of the device that does the encryption and decryption. Such a computer is controlling the security of the network, and must therefore operate in a verifiably secure way. A minicomputer with a kernel is a logical candidate for such an application, and exploration of the interfaces between encryption devices and secure minicomputers began in late 1974.

## SECTION V

### OUTLINE OF THE DEVELOPMENT EFFORT

This section summarizes the entire development effort. The tasks of this effort produce techniques, prototypes and application aids, aimed at equipping Air Force computer users with the capability to do efficient secure multilevel computing; they should result in an immediate improvement in the ability of the Air Force users to meet its computer security requirements. The intent of this section is to present an overview of each of the more than fifty component tasks that make up the effort and to indicate how they fit together.

For the purpose of this section, the tasks have been divided into five groups:

- a. the prerequisite group;
- b. the secure general-purpose system development group;
- c. the technology transfer group;
- d. the application aids development group; and
- e. the secure computing environment development group.

Figure 3 depicts the relationship of the groups and tasks. Reference to this figure may prove helpful when reading their descriptions.

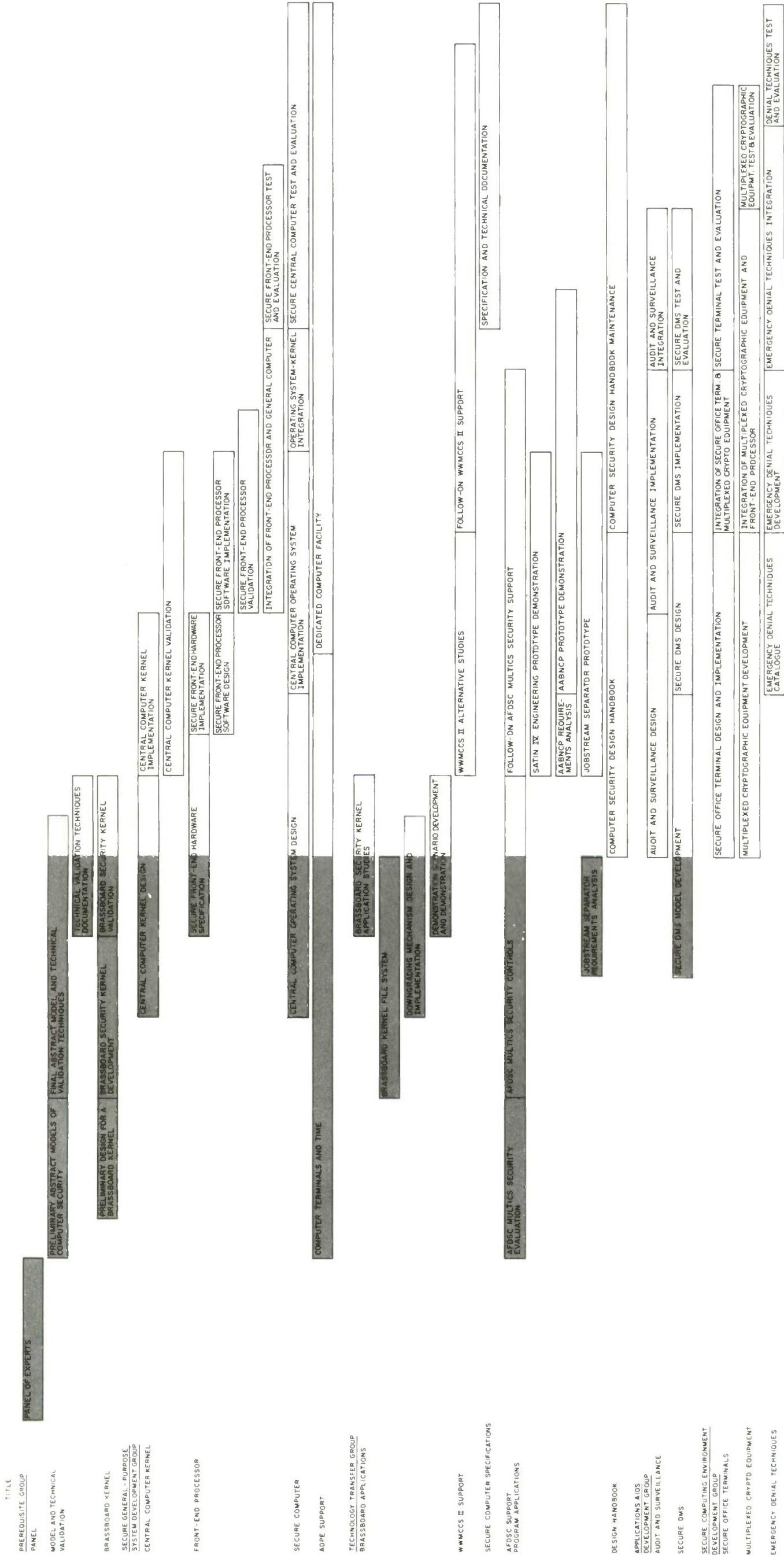
#### THE PREREQUISITE GROUP

The prerequisite group includes initial tasks necessary to the achievement of multilevel secure computing capabilities. Its tasks develop the plans, theories, technology and demonstrations necessary to solve the multilevel computer security problem. Most of the tasks have already been completed and are discussed in earlier sections.

Specific tasks in the prerequisite group include:

Task 1 -- Panel of Experts -- This task involved the formation and operation of the ESD computer security technology panel. This task is completed.

Task 2 -- Preliminary Abstract Models of Computer Security -- The preliminary model task involved the early phases of the security model developments by ESD, MITRE and Case Western Reserve University. This task is completed.



Task 3 -- Final Abstract Model and Technical Validation Techniques -- The final models describe objects that correspond to segments in a storage hierarchy. This task also addresses development and application of technical validation techniques that can be applied to kernel module formal specifications. This task is in progress.

Task 4 -- Technical Validation Techniques Documentation -- This task provides formal documentation and tools for verifying that a security kernel corresponds to the security model. This task is in progress.

Task 5 -- Preliminary Design for a Brassboard Security Kernel -- This task developed the first design iteration for the PDP-11/45 security kernel. This task is completed.

Task 6 -- Brassboard Security Kernel Development -- This task completed the design and implementation of the security kernel for the PDP-11/45. This task is completed.

Task 7 -- Brassboard Security Kernel Validation -- This task proceeds with the proofs and verifications required to effect technical validation of the Brassboard Security Kernel. This task is in progress.

#### THE SECURE GENERAL-PURPOSE SYSTEM DEVELOPMENT GROUP

The secure general-purpose system development group takes the models, tools, and concepts prepared by the prerequisite group and reduces them to practice by developing a large-scale, general-purpose secure system. The product is a prototype of a secure large-scale computer system (based on the existing Multics system) suitable for field use and capable of serving as a guide for Air Force users who have a requirement for such systems. The tasks in this group cover development and technical validation of kernels for the secure computer system and its front-end processor, and modification of the operating system software to provide a useful computing environment outside the kernels.

Task 8 -- Central Computer Kernel Design -- The mathematical model and brassboard kernel design are the foundation for design of a kernel for a secure general-purpose central computer. This task develops the design of a formal specification for a kernel for the Honeywell 6180 processor. This task is in progress.

Task 9 -- Central Computer Kernel Implementation -- Given a design for a central computer security kernel, this task develops

the code that implements the kernel.

Task 10 -- Central Computer Kernel Validation -- This task proceeds with the proofs and verifications (also penetration tests, if desired) required to effect validation of the kernel for the central processor of the secure general-purpose system.

Task 11 -- Secure Front-End Hardware Specification -- This task specifies a hardware architecture that provides a basis for implementation of a secure front-end processor for the secure central computer. This architecture must be capable of supporting its own security kernel. This task is in progress.

Task 12 -- Secure Front-End Hardware Implementation -- This task provides the hardware for the secure front-end processor.

Task 13 -- Secure Front-End Processor Software Design -- This task will result in a design for the secure front-end processor kernel and all other software necessary to interface the front-end processor with the central computer. Formal specifications will be used to define and aid in verification of the front-end processor kernel.

Task 14 -- Secure Front-End Processor Software Implementation -- The design prepared by Task 13 is implemented on the hardware made available by Task 12.

Task 15 -- Secure Front-End Processor Validation -- This task proceeds with the proofs and verifications required to effect technical validation of the front-end processor kernel.

Task 16 -- Integration of Front-End Processor and Central Computer -- This task integrates the front-end processor and the central computer into a cooperating unit. Special attention is paid to the interaction of the two processors' security kernels.

Task 17 -- Secure Front-End Processor Test and Evaluation -- This task performs the functional test and evaluation of the secure front-end processor in an environment that includes a secure central computer and secure communications peripherals.

Task 18 -- Central Computer Operating System Design -- The operating system for the secure central computer must exploit the environment provided by the kernel. This task designs a suitable operating system based as much as possible on the existing Multics operating system. This task is in progress.

Task 19 -- Central Computer Operating System Implementation -- This task modifies the Multics operating system to work with the

kernel, based on the design prepared by Task 18.

Task 20 -- Operating System-Kernel Integration -- This task integrates the central computer security kernel and operating system.

Task 21 -- Secure Central Computer Test and Evaluation -- This task tests and evaluates the utility, efficiency, and acceptability of the secure general-purpose computer in a user environment.

Task 22 -- Computer Time and Remote Terminals -- This task represents the requirement of the secure general-purpose system development group for time-sharing access to a Multics computer system. Such access is required during the early phases of the central computer kernel and operating system design and development. This task is in progress.

Task 23 -- Dedicated Computer Facility -- Once implementation of the central computer kernel and operating system begins in earnest, a dedicated secure facility is required to support development, testing and kernel storage. While such a facility can support users other than those involved in the secure system development, the nature of the kernel and operating system development will be such as to provide a rather dynamic and oft-changing software environment. If the kernel and operating system development tasks are to be pursued in an efficient and expeditious manner, they must have access to a development facility without excessive regard for impact on production users. This task defines the requirement for the dedicated secure facility.

#### THE TECHNOLOGY TRANSFER GROUP

The prerequisite group develops technology and initial products for achievement of multilevel computer security. The secure general-purpose system development group applies the technology and develops a prototype of a multilevel secure "computer utility". Tasks of the technology transfer group are the key to applying the results of the first two groups to meeting the specific computer security requirements of the community of Air Force computer users. These tasks provide specifications, usable products and engineering techniques in forms suitable for direct application by user commands and acquisition agencies. Specific sets of tasks in this group deal with providing support to the Air Force Data Service Center's multilevel secure Multics system, with developing an Air Force Computer Security Handbook, and with specifying security requirements and controls for other Air Force systems.

Task 24 -- Brassboard Security Kernel Application Studies -- The brassboard security kernel for the PDP-11/45 (or similar minicomputers) provides a secure (though small) computer system in an early time frame. A variety of proposed applications could benefit from the availability of such a secure computer. This task provides documentation and application guides for the brassboard kernel for direct use in operational systems. This task is in progress.

Task 25 -- Brassboard Kernel File System -- The automated processing and correlation of data from tactical sensors requires concurrent processing of data at various classification levels. This task is the first in a series that will result in a demonstration software system for securely processing data in a tactical environment. This task is directed towards the design and implementation of a file system for the brassboard kernel. This task is in progress.

Task 26 -- Downgrading Mechanism Design and Implementation -- A key requirement of the application discussed in the last task is the capability to selectively sanitize and downgrade sensor information. This task extends existing computer security technology and concepts to fit the downgrading requirement and will result in the design and implementation of a downgrading mechanism for the brassboard kernel. This task is in progress.

Task 27 -- Demonstration Scenario Development and Demonstration -- In order to substantiate the usefulness of the software system developed by Tasks 25 and 26, this task prepares a demonstration scenario for processing and downgrading information in a tactical environment. The scenario and demonstrations will illustrate situations and instances where the capabilities of the proposed system are necessary. This task is in progress.

Task 28 -- WWMCCS II Alternative Studies -- The planning for a second generation of WWMCCS ADPE must begin early and include explicit provision for multilevel security. This task supports the WWMCCS II planning by establishing specific Air Force WWMCCS II security requirements and by evaluating the alternative approaches to meeting WWMCCS II ADPE security requirements.

Task 29 -- Follow-on WWMCCS II Support -- This task continues the support initiated in the last task through the specification, acquisition and evaluation of security elements of WWMCCS II ADPE.

Task 30 -- Specification and Acquisition Guidance Documentation

-- The secure general-purpose system development group develops a verifiably secure "computer utility" system. While Air Force users can acquire secure computing capability by duplicating the prototype, it is vital that they, also be able to specify a secure system for competitive acquisition from any of a variety of vendors. This task translates the prototype design and experience into sample secure system specifications and associated guidance for acquiring agencies.

Task 31 -- AFDSC Multics Security Evaluation -- This task provided a preliminary evaluation of the suitability of the Honeywell Multics computer system for use in a multilevel (Secret-Top Secret) environment at Air Force Data Services Center. This task is completed.

Task 32 -- AFDSC Multics Security Control -- This task applies preliminary computer security modeling results to the specification, development, testing, and integration of security control enhancements intended to make Multics suitable for use in the two-level environment at AFDSC. The controls provide Multics with enhanced protection, and adapt it for use in a specific military security environment; however, they do not insure that the system can withstand malicious penetration efforts. This task is in progress.

Task 33 -- Follow-on AFDSC Multics Security Support -- Once the AFDSC Multics Security controls are installed and operational, they must be subject to continued validation, review and enhancement. (A true security kernel would not require such a degree of continuing support, as it would be compact, isolated, and relatively stable). This task provides the requisite support and assists AFDSC in planning for eventual transition to the complete and secure systems developed by the tasks already described.

Task 34 -- SATIN IV Engineering Prototype Demonstration -- This task provides a prototype communications network processor compatible with SATIN IV goals. The purpose of this prototype is to demonstrate the applicability of current computer security technology to SATIN IV.

Task 35 -- AABNCP Requirements Analysis -- This task analyzes the multilevel computer security requirements in the Advanced Airborne Command Post.

Task 36 -- AABNCP Prototype Demonstration -- This task provides a prototype verifiable computer system capable of providing the controlled data sharing required by the Advanced Airborne Command Post.

Task 37 -- Jobstream Separator Requirements Analysis -- This task investigated the application of a secure minicomputer to automation of the "color-change" process at various WWMCCS sites. The jobstream separator offers a practical, immediate solution to the inefficiencies inherent in present security level change procedures. This task is completed.

Task 38 -- Jobstream Separator Prototype -- This task will design and implement a prototype jobstream separator for the Honeywell WWMCCS computers. Included in this task will be development of the security control minicomputer, suitable modification of the main computer's hardware and software and design of additional necessary hardware to permit automation of the "color-change".

Task 39 -- Computer Security Design Handbook -- This task codifies available information to guide designers of computer systems faced with security requirements. The information is organized as a handbook suitable for periodic updating (Task 40).

Task 40 -- Computer Security Design Handbook Maintenance -- As development continues and new technologies become available, they must be transmitted to system designers. This task updates the computer security design handbook periodically (every six months to a year) to reflect new results, techniques and practices.

#### THE APPLICATION AIDS DEVELOPMENT GROUP

Certain subsystems, while not central to providing multilevel secure computer systems, will facilitate cost effective use of secure systems in the field. The application aids development group produces two such subsystems -- one to facilitate data base management in a secure computer environment, the other to provide for auditing of user actions in a secure environment. The former subsystem facilitates use of the secure system on a large data base of mixed classifications, while the latter helps enforce requirements for user accountability and responsibility.

Task 41 -- Audit and Surveillance Design -- This task establishes the requirements and design for a security audit subsystem for use with the secure general-purpose prototype system. Required kernel actions and appropriate audit strategies are defined by this task.

Task 42 -- Audit and Surveillance Implementation -- The audit and surveillance tools designed by Task 41 are implemented to operate in the kernel and secure system environment.

Task 43 -- Audit and Surveillance Integration -- This task integrates audit and surveillance tools into the secure general-purpose prototype system.

Task 44 -- Secure DMS Model Development -- If a data management system is to operate on files of several classifications simultaneously, and is to assure that a user accesses only a controlled subset of those files, the DMS must be based on a model which is compatible with the security kernel that controls it. This task provides a model on which such a data management system can be based. This task is in progress.

Task 45 -- Secure DMS Design -- This task prepares a design for a secure data management system that implements the model developed by Task 43.

Task 46 -- Secure DMS Implementation -- This task implements a secure data management system as an application subsystem of the secure general-purpose prototype system.

Task 47 -- Secure DMS Test and Evaluation -- This task evaluates the operational utility of the secure data management system in the secure computer environment.

#### THE SECURE COMPUTING ENVIRONMENT DEVELOPMENT GROUP

A secure multilevel computer system should extend the scope of the classified computing services provided to Air Force users. For example, individuals with small computing tasks to perform at the Secret level should be able to perform those tasks on a multi-user secure timesharing system. For computing service to be provided efficiently to users, it should be possible to place a terminal for Secret level processing in an office as one would a safe.

This group of tasks is aimed at developing more efficient terminal and communications security equipment for the interactive computing environment. While these developments are not necessary for multilevel computer security, they will provide for more cost-effective use of secure computer systems.

A second set of tasks within this group provides rapid, safe means of rendering classified information on storage media inaccessible. This set is aimed specifically at the problems of processing classified information in the tactical environment and of making it possible to store or transmit media that contain classified information using ordinary containers.

Task 48 -- Secure Office Terminal Design and Implementation --

This task develops a prototype of a secure terminal suitable for interactive computer applications, with integrated communications security equipment, for use in a general office environment (not a vault). This task builds extensively on experience gained in developing a secure terminal for use with communications systems.

Task 49 -- Integration of Secure Terminal and Multiplexed Cryptographic Equipment -- This task integrates the secure terminal developed by Task 48 with the multiplexed cryptographic equipment developed by Task 51.

Task 50 -- Secure Terminal Test and Evaluation -- This task tests and evaluates the secure terminal for application with the secure prototype computer system.

Task 51 -- Multiplexed Cryptographic Equipment Development -- A secure front-end processor can control a single cryptographic device that provides security for a number of separate secure terminals, or for many users in a computer network. This configuration can reduce the cost, space and power required for cryptographic equipment at computer sites that serve numerous remote terminals. This task develops the required cryptographic equipment.

Task 52 -- Integration of Multiplexed Cryptographic Equipment and Secure Front-End Processor -- This task integrates the cryptographic equipment with the secure front-end processor. Application programs for the front-end processor will be needed to drive the multiplexed cryptographic device.

Task 53 -- Multiplexed Cryptographic Equipment Test and Evaluation -- This task provides operational test and evaluation of the multiplexed cryptographic equipment in an environment including secure central computer, front-end processor and secure terminals.

Task 54 -- Emergency Denial Techniques Catalog -- This task begins the development of techniques for emergency denial of access to classified information with a survey and catalog of potentially suitable techniques. This task will specifically consider application of media encryption techniques.

Task 55 -- Emergency Denial Techniques Development -- This task selects promising techniques from the catalog developed by Task 54 and develops prototype equipment for evaluation.

Task 56 -- Emergency Denial Techniques Integration -- For evaluation, the prototype denial techniques will be used with the prototype secure general-purpose system. This task integrates

the denial prototype equipment into the secure general-purpose system.

Task 57 -- Denial Techniques Test and Evaluation -- This task assesses the reliability, effectiveness and compatibility of the prototype denial equipment. Not only must the equipment effect denial on demand, but it must also guarantee against accidental denial or loss of information.

## SECTION VI

### SUMMARY

This document has described the problem of multilevel computer security and a technological basis for its solution. Section II reviewed the current alternatives for processing classified information with ADP systems, and outlined the major economic and operational impacts of those alternatives.

The reference monitor concept introduced in Section III offers a technological basis for security controls whose effectiveness can be verified. The secure systems described in Section IV apply the reference monitor concept to meet the requirements of Air Force users. The PDP-11/45 security kernel is the heart of a small secure system that can be used in the near term. The kernel is based on a mathematical model and is already in experimental use. Its security is now being verified by a rigorous formal process.

The jobstream separator is a potentially cost-effective alternative to today's manual color-changing procedures. The fact that the jobstream separator implements a reference monitor allows its security to be verified. The Multics security kernel will provide the prototype of a large multilevel system for use in command control, administrative and intelligence applications.

Finally, the SATIN IV internal access control mechanism typifies the direct application of the reference monitor concepts to real Air Force programs and systems. Section V identifies several other such applications.

The reference monitor concept has been brought from an academic abstraction to a basis for security in real systems. The development tasks exploit the concept in an orderly manner -- first by developing prototypes of secure systems that apply the concept, and then by transferring the techniques proven by the prototypes to operational systems in the field. The basic approach is technically sound, and if the Air Force is to meet its pressing requirements for secure multilevel computing, these technical development efforts are necessary.

## BIBLIOGRAPHY

Anderson, James P., Computer Security Technology Planning Study, ESD-TR-73-51, October 1972.

Bell, D. E., E. L. Burke, A Software Validation Technique for Certification: The Method, ESD-TR-75-54, The MITRE Corporation, Bedford, Massachusetts, November 1974.

Bell, D. E. and L. J. LaPadula, Secure Computer Systems, ESD-TR-73-278, Vol. I-III, The MITRE Corporation, Bedford, Massachusetts.

Burke, E. L., Synthesis of a Software Security System, MTP-154, The MITRE Corporation, Bedford, Massachusetts, August 1974.

Dijkstra, E. W., "The Structure of the THE Multiprogramming System", Communications of the ACM, Volume II, Number 5, May 1968.

Lipner, S. B., MACIMS Security Configurations, WP-3697 (Internal Communication), The MITRE Corporation, Bedford, Massachusetts, January 1971.

Lipner, S. B., A Minicomputer Security Control System, MTP-151, The MITRE Corporation, Bedford, Massachusetts, February 1974.

Price, W. R., Implications of a Virtual Memory Mechanism for Implementing Protection in a Family of Operating Systems, PhD Thesis, Carnegie-Mellon University, June 1973.

Robinson, L., P. G. Neumann, K. N. Levitt, and A. Saxena, "On Attaining Reliable Software for a Secure Operating System", to appear in 1975 International Conference on Reliable Software, Los Angeles, California, 21-23 April 1975.

Schell, R., P. Downey, G. Popek, Preliminary Notes on the Design of a Secure Military Computer System, MCI-73-1, January 1972.

Schiller, W. L., Design of a Security Kernel for the PDP-11/45, ESD-TR-73-294, The MITRE Corporation, Bedford, Massachusetts, June 1973.

Smith, L., Architectures for Secure Computing Systems, ESD-TR-75-51, The MITRE Corporation, Bedford, Massachusetts, 30 June 1974