STATEMENT OF WORK

FOR

SECURE MULTICS DESIGN, DEVELOPMENT AND CERTIFICATION

Contract F19628-74-C-0193

22 June 1975 (Revised)

TABLE OF CONTENTS

1.0 Introduction

This contract is part of a cohesive program to develop the technology required for secure computing with classified information and various levels of user clearances. The technology required to implement secure computer systems has been demonstrated on a small prototype system by a prior ESD-sponsored effort. The aim of this effort is to demonstrate that the technology also applies to a large efficient general purpose system. To this end issues of efficiency and compatibility with the existing Multics must be addressed by this contract. However, these issues cannot be resolved at the expense of security, for the final product must be capable of certification and demonstration use in a true multilevel environment.

The overall objective of this contract is threefold:

1. This effort will demonstrate the feasibility of constructing a secure general purpose computer system by constructing a prototype system.

2. In the course of constructing a secure general purpose computer system, a minicomputer will be developed for use as a front end processor element in the secure general purpose system. This minicomputer will provide a suitable base for implementing secure systems in environments other than the general purpose system environment.

3. A prototype secure Multics system will be developed for test and evaluation at an operational Air Force Multics installation such as the Air Force Data Services Center.

The military has the responsibility for protection of information in its shared computer systems. The military must insure the security of its computer systems before they are put into operational use. That is, the computer system security must be "certified", since once military information is lost it is irretrievable and there are not legal methods for redress.

Most contemporary shared computer systems are not secure because security was not a mandatory requirement of the initial hardware and software design. The military has effective means of implementing and certifying physical, communication, and personnel security, so that the nub of our computer security problem is the certification of information access controls in the operating system and supporting hardware. The primary need is for an effective means for enforcing simplistic protection relationships, by implementing a certifiable "security kernel". Initially, solutions to some of the more complex protection problems such as mutually suspicious processes is not required.

The purpose of this contract is to design and develop the interfaces to the so-called security kernel of a general purpose computer system which provides controlled, direct sharing of information and programs between users with different authorized access. In addition, the results of this contract will define those non-kernel security related functions that will be required in such a computer system.

After the kernel interfaces have been designed and the non-kernel security related functions identified, the contractor shall proceed with the implementation of a full scale secure Multics system. This implementation involves the design and technical certification of a security kernel for Multics, the restructuring of the Multics operating system to interface with the kernel, and the design and development of both the hardware and software of a secure front end processor to extend the concept of the reference monitor to the input/output operation of Multics.

2.0 Scope

2.1 Objective

The contractor shall review the functional primitives of a security kernel for the Multics general purpose multi-user computer system. These primitives shall implement the necessary functions to meet the Department of Defense security requirements for the protection of classified information. These functional primitives shall deal only with the security of information objects and their access. Excluded from the security kernel functional primitives are operating system primitives which do not affect information security. Included in the class of excluded primitives are those providing system integrity such as preventing denial of service, page replacement and scheduling strategies, resource allocation policies, and other non-security related operating system functions. In reviewing the security kernel primitives, the contractor shall suggest alterations to ensure efficient interfaces with the non-security related portions of the operating system. A secure Multics would contain certain non-kernel security related functions including but not limited to login verification, input/output daemons, and security officer functions. The contractor shall define these functions and identify special problems posed by their interfaces to the security kernel.

In addition, the contractor shall conduct a study of the integration requirements for the design and implementation of security software. This study shall identify the scope and detailed integration tasks for a prototype design and implementation effort.

When the contractor has finished the integration study, he shall proceed to implement and provide the basis for the certification of a

secure Multics system. The implementation shall follow the recommendations of the integration study and the technical direction provided by the government.

As part of the development of a secure Multics, the contractor shall design, develop, and provide the basis for the certification of a secure communications processor to provide the functions of a front end processor for Multics. The secure communication processor shall be suitable for use in a network environment to support the following objectives:

a. Off-load from the host much of the network protocol processing.

b. Maintain continuity of operations in the event of failure of the host by providing connection of terminal users to alternate hosts.

c. Provide an interface with networks including the Autodin II and ARPA networks.

d. Insure the security of the individual hosts who are part of the network environment, i.e., insure that the network connection does not add security problems.

2.2 Approach

In examining the operations of the kernel, the contractor shall be guided by the following principles:

a. Complete Mediation -- A secure system must provide complete security mediation of information references. All references must be validated by those portions of the system hardware and software responsible for security.

b. Isolation -- The validation operators, a "security kernel", must be an isolated, tamper-proof component of the system. This kernel must provide a unique, protected identity for each user who generates references, and must protect the reference-validating algorithms.

c. Simplicity -- The security kernel must be simple enough for effective certification. The demonstrably complete logical design should be implemented as a small set of simple primitive operations and system data base structures that can be shown to be correct.

To ensure correct and consistent operation and to provide a method of certification, the kernel functional primitives shall be defined from an abstract model of computer security which has been mathematically proven correct. Such models are being developed under

related contracts and the Government will furnish the products of this development to the contractor. Then if an actual instance of the security kernel can be shown to correctly implement the functions of the model, the kernel can be certified correct, since the model has already been proven correct.

The contractor has previously identified several programming and structural techniques (e.g., the accomplishment of dynamic linking outside the supervisor and the use of a structured programming language) which may help to simplify the Multics operating system. The contractor shall investigate these and other techniques for possible benefits in easing the certification problem or simplifying the kernel structure.

The Multics system architecture will be the target of the integration requirements study. The study will also consider the prototype secure computer system as the context for future system engineering or the design of a secure front-end processor, the requirements and characteristics of surveillance and audit in a secure system, and the economic/technical potential of a secure low cost office terminal.

To implement and provide a technical certification of the Multics kernel, the contractor shall develop a set of formal specifications derived from specifications provided by the government. These specifications shall be amenable to technical certification using the techniques developed by Bell & Burke, (1) Robinson, et. al. (2) and used by Schiller. (3) The contractor shall also develop an algorithmic representation of the kernel in a suitable high level language (like PL/I) and a machine language version of the kernel, both technically certified by the previously mentioned techniques.

The contractor, utilizing the early results of the contract in restructuring the Multics operating system, shall complete the restructuring so that the operating system interfaces with the kernel while providing an efficient, compatible user interface.

---

(1) D. E. Bell and D. L. Burke "A Software Validation Technique for Certification: The Methodology", ESD-TR-75-54, Vol 1., The MITRE Corporation, Bedford, MA. November, 1974.

(2) L. Robinson, P. G. Neumann, K. N. Levitt & A. Saxena, "On Attaining Reliable Software for a Secure Operating System". Proceedings of 1975 International Conference on Reliable Software, Los Angeles, CA., April, 1975.

(3) W. L. Schiller, "The Design and Specification of a Security Kernel for the PDP-11/45", ESD-TR-75-69, The MITRE Corporation, Bedford, MA., March, 1975.

The contractor shall design, develop, and demonstrate a secure front end processor for Multics based on the results of the Secure Communications Processor Architecture Study (Contract No. F19628-74-C-0205). The development of the front end processor shall assure compatibility with the current I/O capabilities of the Multics system. In the course of the secure front end processor development, the contractor shall

-choose a minicomputer with appropriate architecture.

-design, build, and test a memory mapping device to augment the minicomputer hardware so that it adequately supports a security kernel.

-develop the detailed logic and mechanical design for a prototype version which meets militarization requirements.

-design, build and technically certify a security kernel for the secure communications processor.

-develop an operating system to run on the secure communications processor.

-develop the applications software to support the use of this processor as a secure front end processor for Multics.

3.0 General Background

3.1 Deficiencies of Present Systems

Most current computer systems exhibit a complex, ad hoc security design with a diffuse implementation that violates the third principle cited above, simplicity. Large portions of complex operating system execute in an all-powerful supervisor state, so that the entire operating system has potential security implications. Whatever nominal security controls exist in such bug-prone monoliths are not effectively isolated (in violation of the second principle) and so can be tampered with by exploiting errors or trap doors in other parts of the operating system.

The significance of these inherent security weaknesses has been amply and repeatedly demonstrated by the ease with which contemporary systems have been penetrated. Unfortunately, this lack of an underlying design methodology cannot be effectively overcome by ad hoc "fixes" and "security features". Although the current Multics system has basic security design features, it has no precise security criteria which provides a basis for certification.

3.2 Certification

A naive (but occasionally attempted) approach to insuring the security of a complex operating system is to have a penetration team of "experts" test the system. It is supposed that repeated unsuccessful penetration attempts demonstrate the absence of security "holes". A security evaluation through such attempts may reveal weaknesses of a system but provides no indication of the presence or absence of trap doors or errors in areas unnoticed by the attack team. The failure of an attack team to notice a particular penetration route does not provide a basis for proving or certifying that any future penetration attempt will overlook it. The underlying concern is that an active penetrator is not particularly thwarted by the various flaws found and fixed through testing so long as there remains just one vulnerability that he can find and effectively exploit.

On the other hand, the three principles identified above (viz., complete mediation, isolation, and simplicity) should lead to a simple, well-defined subset of the system totally responsible for information protection. The goal is that the primitive functions of this small, simple kernel can be tested by enumeration, and other parts of the system are not relevant to security. As a result of the small portion of the operating system included in the kernel, most system changes will not affect the kernel, so routine system maintenance will not require repeated recertification.

3.3 Operating System Organization

Developers of current computer systems have paid very little attention to issues involving security. Where attention has been paid, designs tend to be ad hoc. In most cases, the operating system executes in an all-powerful supervisor state, is not effectively isolated and can be tampered with easily. The Multics kernel design effort will provide a security kernel that will effectively overcome these deficiencies.

The introduction of a security kernel in Multics will require restructuring and a revised organization of the Multics operating system. The revised operating system shall comprise those non-kernel functions that are necessary to maintain the same user interface that currently exists. It is essential that alterations to the user interface be minimal so that user compatibility is maximized.

4.0 Contractor Tasks

The contractor shall develop, maintain, and submit a detailed set of technical working notes relating to each of the following specific tasks throughout the course of the project.

4.1 Analysis Activities

In pursuit of the objectives of this contract the contractor shall conduct the following general activities throughout the course of the project.

4.1.1 Review Security Kernel

The contractor shall perform a detailed review of a set of security kernel functions and primitives to be developed under related efforts to be identified by the Government during the performance of work under this contract. The contractor shall attempt to insure, based on his experience with Multics, that the primitives form a complete set for security, but do not contain unnecessary functions (from the standpoint of security). The contractor shall prepare the results of this review as a technical report identifying unnecessary functions in the kernel design supplied by the Government, functions that should be added to the kernel, and proposed restructuring to improve interfaces between the kernel and the remainder of the operating system.

4.1.2 Define Operating System Interfaces

The contractor shall define a revised organization for the Multics operating system to interface with the security kernel. He shall identify design and structural techniques by which the operating system may simplify the kernel's organization and functions. These techniques include but are not limited to restructuring for parallel processing, replacing bulk memory with large primary memory and using ARPANET techniques for input/output. If the ARPANET interface is used as the front-end I/O processor, then the evaluation of the security kernel for Multics must consider the security implications of the ARPANET.

4.1.3 Definition of Non-Kernel Security Functions

The contractor shall define those non-kernel security related functions (e.g., user password authentication) which are required to be part of the Multics operating system. The definitions of these functions shall include considerations of ease of certification, simple user interface, and efficient interface with security kernel.

4.2 Reduction of Hardcore to Permit Certification

The contractor shall accomplish the following specific tasks during the course of this contract. The following paragraphs describe several specific tasks which so far have been identified as plausible candidates for the restructuring of Multics. Several of the tasks suggested here involve modifications to the current Multics system. For each of these, two observations are in order: 1) a method of measurement of progress is needed, to establish "how much" each

modification carries the project toward the goal of an auditable central core; and 2) discussions and negotiations with the Multics development team are required to establish whether or not each suggested modification should be targeted toward installation in some current or future standard version of Multics. It seems inevitable that at least some of the changes which will be needed to achieve an auditable system will violate either compatibility or performance constraints of the standard system, and thereby force development of a parallel version.

Most of the initial tasks are directed toward identifying more exactly which functions of the operating system must be privileged, and which, by careful design, can be left to the user (in Multics, on a per ring basis.) This work may be described as better defining where the security perimeter of the system should be located. It is expected that there will be many more such tasks in this class. Two remaining major areas of work, both more suitably tackled later, are the rewriting of otherwise untouched protected programs in a standard auditable style, and installation of at least one internal firewall or protection ring within the protected supervisor to separate those procedures which actually implement the protection mechanism itself--a so-called "security kernel". The tasks so far identified are the following:

4.2.1 Dynamic Linker

Removal of the dynamic linker and library search modules from ring 0. This modification would remove two large and hard-to-audit modules from the protected area. The dynamic linker is especially hard to audit because its correct operation depends on its interpreting a highly structured but unprotected data base (an object segment linkage and definition area) without accidentally getting mixed up. Neither of the modules has need for supervisor privileges or protection from the invoking user; both are currently in ring 0 because of their intimate interface with the storage system. The task includes better definition of the interface to the storage system, and taking advantage of the lower cost of changing protection rings with the 6180 hardware.

4.2.2 Reference Name

Removal of the "reference name" concept from ring 0. The notion of a remembered reference name is currently maintained on a per-ring basis, in the per-process known segment table in ring 0. There is no apparent reason why reference names cannot be remembered in the ring of interest; such an arrangement will also permit a subsystem writer to disable reference names if he desires. This change would simplify both the implementation and the description of several supervisor interfaces.

4.2.3 Working Directory

Removal of the "working directory" concept from ring 0. The comments regarding reference names apply to the working directory also.

4.2.4 Storage System

Develop a uniform storage system status-returning entry. This minor cleanup would replace about half a dozen distinct supervisor interfaces with a single, more easily audited interface for returning to the user any status information about his segments. (This task is actually the iceberg tip of a larger task to develop a simple, consistent set of supervisor entries.)

4.2.5 Traffic Controller

Modify the traffic controller to provide cheap, rapidly scheduled, wired-down processes which can operate using any descriptor segment which happens to be available in primary memory. This change would allow the present interrupt handlers for the printer, teletype interface, network interface, and tape handlers to be replaced with scheduled processes. The actual interrupts would do nothing but notify the appropriate process. The virtue of this strategy is that scheduled processes can coordinate their activities with standard coordination primitives (block, wakeup, wait and notify); the present interrupt handlers cannot, for example, wait on an interlock, and are therefore filled with tricky code which uses read-alter-rewrite instructions to avoid encountering interlock situations.

4.2.6 Multiple Processor

Modify the traffic controller (and other per-process data base managers) to permit multiple processes per address space. This modification is the key to untangling several very complex paths through the present supervisor. Typewriter management, network interface management, dialup handling, and quit handling can all be done as simple coordination of parallel processes rather than with the present ad hoc "multiplexing" of a single process among many conceptually parallel activities. The propagation of this change through the network control is part of the task, to test its effectiveness.

4.2.7 Message Strategy

Develop a uniform process coordination/message passing strategy. The current Multics has several different coordination and message passing schemes in it, each with slightly different properties as to the scope of naming and details of interface:

-- Wait and notify, used for storage system signalling
-- Block and wakeup, used for I/O coordination
-- Interprocess communication, used for multiplexing processes among event call channels
-- Signals, used to generate interrupts in a process
-- Message segments, used to queue messages in a catalogued place
-- Mail facility, used for inter-user mail
-- Lock and Unlock, used for coordinating data base use
-- The I/O system, used for message passing and queuing

The task here is to develop one or two moderately flexible process coordination and message passing facilities which can be used to support all of the various users of those facilities. The payoff in simplification of the central supervisor should be quite high.

4.2.8 Network Interface

Merge the network interface with the typewriter communications interface. These two interface programs are two of the largest protected subsystems; they largely duplicate each other. The typewriter control system should use the network code conversion strategy which does not require protection; the network interface should use a buffering strategy more similar to the typewriter modules. With moderate effort, the interface between the 6180 and the DataNet 355 communications computer can be made essentially identical to the network host-to-IMP interface, allowing further control program sharing. By taking the best design from each of the two systems, a compact and effective communication interface module should result, with minimum privileged code.

4.2.9 System Census

This task consists of conducting a census of the number of programs, number of lines of source code, and number of lines of generated text (machine instructions) in the protected supervisor. This census will be useful for two purposes: identifying subsystems which are unreasonably large or complex for further study, and to keep track of progress in simplifying and reducing the size of the protected supervisor.

4.2.10 ALM Program Catalogue

A list of all protected programs currently written in ALM (the Multics Assembly Language) should be developed, with the goal of identifying all reasons why assembly language has been used. This task includes the development of proposals to eliminate the need for assembly language completely. Such elimination is an important step in simplifying the description of the system and of simplifying the job of an auditor.

4.2.11 Coding Standards

Development of coding style standards. A standard programming style will need to be developed, one which emphasizes clarity in program structure to an auditor. Undoubtedly, the programming style will borrow much from the emerging area of structured programming. The task includes the experimental rewriting of some parts of the storage/directory system to the new standards to test their viability.

4.2.12 Unique Segment Numbers

The implication, in terms of simplifying system structure, of using unique identifiers for segment numbers will be explored. An immediate implication of such a strategy would be that pointers containing segment numbers could be left in permanently catalogued, shared storage; many programmed tricks to accomplish the equivalent effect could be eliminated from the system. There are many other implications for system creation, interprocess communication, dynamic linking, and hardware addressing architecture which should be examined; many simplifications seem to follow. An intermediate strategy, of using unique identifiers to replace the absolute addresses in a segment descriptor word, and developing a microprogrammed memory controller architecture which responds to such unique identifiers and contains in a separate box all virtual memory implementation seems worthy of exploration as part of this task.

4.2.13 Hardware Reconfiguration

A fair amount of very intricate machine language code in the protected core of Multics is devoted to the dynamic reconfiguration of processors and memory, a valuable feature. Much of the intricacy can be attributed to performing reconfiguration with hardware not designed for it. A general design developed by P. Schell in his 1971 Ph.D. Thesis should be reviewed and a specific hardware proposal for the 6180 system should be constructed along the lines suggested by Schell. Such a design would probably influence future rather than current versions of the Multics hardware but the result is of interest now to establish how large is the effect in reducing complexity of the protected supervisor. In addition, operation of a secure system probably requires padlocking many of the control panels currently used by the operator to accomplish dynamic reconfiguration.

4.2.14 System Description

If an auditor is to review a supervisor program for correctness, he must have a complete, concise statement of what the program is intended to do. Today's description consists of English language supervisor interface descriptions, with PL/I calling sequences. There is no simple description of the "state" of the

supervisor and the things a user may do to legally alter its state. The first step in this task is simply to collect in one place all the present documentation of the protected supervisor interface, and evaluate it. The next step is to try to develop a more precise state description of the supervisor, and the ways in which a user can change or observe its state. This task seems to include becoming expert in description languages, such as the Vienna Definition Language, so as to develop equivalently powerful methods of describing an operating system.

4.3 I/O and Modeling Tasks

In addition to the direct efforts to reduce the Multics hardware, the contractor shall conduct the following examinations and investigations.

4.3.1 Secure Input-Output Processing.

This effort will consist of the examination of current Multics practices in the areas of input-output and communications processing with regard to their conformance or nonconformance to the principles of a certifiably secure system. Technology reports and functional specifications shall be produced for secure front-end processing and secure input-output processing. A key result of the thesis just completed by D. Clark at MIT is that with correct design, essentially no I/O strategy or device management code, except that dealing with multiplexed channels needs to be protected. Since I/O software is a significant part of the present protected supervisor, a detailed design proposal for a new hardware I/O architecture along the line of Clark's thesis is in order. Thanks to the modular organization of the 6180, it is relatively easy to envision actually building and trying out this design at some point in the future.

4.3.2 Consistency with Modeling Activities.

Each effort shall be monitored by personnel familar with and qualified in the mathematical investigation of certification now underway under Government sponsorship, in order to ensure consistency of the proposed technology development activities with the results of the mathematical investigation.

4.4 Integration Requirements Analysis

The contractor shall determine and evaluate the future work required to design and implement security software on the Multics system. He shall address the overall work to plan and integrate the complete elements of ADP systems security. He shall consider not only the central computer but also the user interface elements needed to make secure computing practical and available. He shall define the planning and integrating tasks on the central computer and its

operating system software, a front-end processor with multiplex cryptographic capability, a secure office terminal, and applications engineering such as a secure data management system and security audit and surveillance.

4.5 Kernel Development

The contractor shall begin the design and certification of a security kernel for the Multics computer system which involves the development of software for both the Multics central processor and the front end processor.

The contractor shall begin to develop a set of formal (Parnas) specifications (1) for the security kernel. These specifications shall be based on a mathematical model for security and shall be derived from the preliminary kernel specifications furnished by the government. Special attention shall be given the form of the specifications to allow both the ultimate certification of the design and its faithful implementation in a high-level language and in machine language executable on the Multics (central processor and front end) hardware. The contractor shall be specifically concerned with the efficiency and compatibility of the specifications in supporting the present Multics user and external interface. The security kernel shall be capable of supporting an operating system which is compatible at the user interface with the existing Multics operating system to the maximum extent possible. Improving the functional capabilities of Multics is not a goal of this effort.

The kernel specification shall be hierarchical in form. The kernel shall be decomposed into levels of abstract machines each specified as a Parnas module. The top level specification documents the security kernel interface with the Multics (central processor and front end) operating system.

4.6 Operating System Restructuring

The introduction of a security kernel in Multics will require the Multics operating system to be restructured and revised. The revised operating system shall comprise those non-kernel functions that are necessary to maintain the same user interface that currently exists. It is essential that alterations to the user interface be minimal so that user compatibility is maintained.

The Air Force will provide the contractor with the preliminary kernel specifications for the security kernel interface that will describe the functions available at the kernel interface. Using the

---

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

interface specification, the contractor shall proceed to define the set of operating system functions that will describe the user interface for the security kernel based Multics system.

The contractor shall prepare a system specification documenting the necessary operating system revisions. In this specification the contractor shall specifically analyze the compatibility of the proposed restructured operating system with the current operating system and shall provide a quantitative estimate of the effect on performance.

4.7 Security Kernel Certification

The contractor shall begin to certify that the formal specification of the security kernel for both the Multics operating system and the front end processor corresponds to an abstract model of security.

4.7.1 Certification Procedures and Approach

The contractor shall develop procedures and approaches to be followed during the certification effort.

Since the security kernel is fundamental to the protection of highly classified data, it is important that the kernel be protected from sabotage during the development process. One example of sabotage is the insertion of trapdoors in the kernel code. Security procedures shall therefore be developed for the general protection of the kernel during the development process. The contractor and the Air Force shall work together to identify a set of procedures describing the clearance requirements for the personnel and physical environment involved in the protection, development, and certification of the security kernel. The contractor shall document these requirements and submit them for approval. The contractor shall then implement the approved procedures.

In advance of the actual certification the contractor shall identify and document the technical approach for certifying the Multics security kernel and the front end processor kernel. The contractor shall select a certification technique and show that this technique is sufficient to provide a convincing and complete deductive demonstration that the security kernel software corresponds to the

---

(1) K. G. Walter, W. F. Ogden, W. C. Rounds, et al., Primitive Models for Computer Security, ESD-TR-74-117, Case Western Reserve University, January, 1974.

abstract models of security. (1) (1) The contractor shall provide a plan for the certification of the Multics security kernel and the front end processor security kernel.

4.7.2 Demonstrating Correspondences

The certification methodology involves the transition from the mathematical model to the machine language code in a series of kernel representations. For each of these transitions the correspondence between the initial and final representation shall be demonstrated. Specifically, the contractor shall initiate the technical certification of the correspondence of the top level of the kernel specification to an abstract model of security. (2) (3)

4.7.3 Certifying Representations

Each stage in the representation of the system shall in itself be proven correct and the general security rules shown inviolable. From the formal kernel specification, the contractor shall begin to technically certify that the levels of decomposition of the specification faithfully represent the top level specification.

4.7.4 Maintaining Correctness of Representations

Techniques for maintaining the correctness of representations shall be advanced in case changes are made to the formal specification and/or the program code. During the lifetime of any system it is inevitable that changes will be made to the capabilities of the system. These changes will involve corresponding changes to some and/or all of the representations of the system. Procedures shall be developed that would describe the format for changing each system representation, taking into consideration ease of recertification.

4.8 Secure Front End Processor (SFEP)

The contractor shall select and begin to obtain for use (with prior written Government approval) during the period of performance a minicomputer that is capable of being augmented by a Security Protection Unit (SPU) to form a Secure Communications Processor (SCOMP). The contractor shall begin the design of a certified

---

(1) D. E. Bell and L. J. LaPadula, Secure Computer Systems: A Mathematical Model, ESD-TR-73-278, Volume I-III, The MITRE Corporation, 1973.

(2) Walter, Ogden, Rounds, et al.

(3) Bell and LaPadula

security kernel and an operating system for the SCOMP. The SCOMP design shall be modular so the SCOMP could serve as a communications/front end processor for a variety of terminals and/or host computers.

A specific application area has been identified to demonstrate the functionality of this machine. The contractor shall begin to design, for later development, a hardware interface and the applications software to support its use as a Secure Front End Processor (SFEP) for a large machine of the Honeywell 6000/Series 60 family. The SFEP-host interface shall be designed, and a SFEP-communications network interface shall be planned. In parallel with this effort, the contractor shall initiate the design of a militarized version. The contractor shall plan to construct a small number of these prototypes as necessary for the purposes of this effort. (The exact number of prototypes will require prior written Government approval.) The construction and form of these prototypes shall be based on standard commercial practices.

For each of the following subtasks, the contractor shall, as appropriate, prepare Technical Notes identifying the alternative approaches considered, the primary technical factors and the basis for the particular alternatives selected.

4.8.1 Minicomputer Selection

The contractor shall select a minicomputer to serve as the hardware base for the Secure Communications Processor (SCOMP). This minicomputer shall be suitable for use as a SCOMP with a minimum capacity of 0.1 million bits of directly addressable, random access storage (e.g., core). The selection shall be based upon the Air Force approved technical report issued under the Secure Communications Processor Architecture Study that will have considered such architectural hardware features as isolation mechanisms, virtual addressing mechanisms, isolation of hardware controls, and input/output controls. Various communication processor applications are to be considered in the hardware selection. The application of the SCOMP as a front end for Multics or a network front end for the Honeywell 6000/Series 60 machines are especially pertinent to this program, and the chosen base must be shown to have sufficient capacity to perform adequately in such environments.

The result of this subtask shall be a subsystem design analysis report that will justify the choice of the SCOMP hardware base. Subject to prior written Government approval of the selection, the contractor shall take action to obtain the chosen minicomputer for use during the period of performance.

4.8.2 Security Protection Unit Design

The contractor shall begin to design and plan for the fabrication of a brassboard Security Protection Unit (SPU) based on the Air Force approved design specifications produced under the Secure Communications Processor Architecture Study. The contractor shall update the development specifications produced under the Communications Processor Architecture Study as necessary.

The SPU will augment the hardware protection already available in the chosen minicomputer. The contractor shall consider including a cache or content addressable memory in the SPU. The combination of SPU and minicomputer will form the SCOMP which provides the hardware necessary for a reference monitor.

4.3.2 SCOMP Design

The selected minicomputer shall be augmented by the SPU to form a SCOMP. The architecture shall be appropriately designed to support a security kernel and to provide for the efficient execution of applications programs.

4.3.3.1 Configuration and Interface Considerations. The following system configuration is representative of the requirements for the use of the SCOMP. There should be the capacity of simultaneously supporting a number of terminals and various terminal types (included TTY model 40 and all those supported by the current Multics system). The number of terminals should be modularly expandable up to a maximum of 256 terminals. However, this maximum capacity may be provided by a small number of processors working together rather than a single processor. The SCOMP shall be able to support terminals operating in both half-duplex and full-duplex mode and at a variety of speeds (including 110, 134.5, 150, 300, 1200, 2400, 4800 and 9600 bps). A SCOMP shall be capable of supporting various external I/O devices (1) such as a card reader, card punch, magnetic tapes and disks. The SCOMP shall in addition be capable of supporting and operating devices such as disks and drums for the purpose of performing internal I/O. (2) The capacity of and throughput to secondary storage media will depend on the nature of the Honeywell 6000/Series 60 interface. The SCOMP shall also be capable of providing a connection to networks, such as IDX II (Autodin II) or the ARPANET.

To augment the SCOMP to include a communications network interface capability, the contractor shall plan for the design of a

---

(1) Burke, E. L., "Concepts of Operations for Input/Output in a Secure Computer System at the AFDSC", The MITRE Corporation, ESD-TR-74-113, November 1974.

(2) Burke

baseband communications network IU for use in connecting the GFEP to the AUTODIN II system as defined in the "Draft Specification (TYPE A) for AUTODIN II Phase 1 (Defense Communications Agency, February 75)". This IU shall implement the American National Standards Institute (ANSI), Advanced Data Communications Control Procedures (ADCCP) - Independent Numbering, X3S3/SG3, Draft 3, 12 Dec 74. Specifically, the operational class defined as Primary-to-Primary, Full Duplex shall be used. The IU shall be capable of operating at the standard communications circuit speeds of 110 bps to 56 Kbs, and shall include all responses and commands defined for the Primary-to-Primary selective Reject Exception Recovery Class of Procedures. However, the capability to intermix the Set Asynchronous Response Mode (SARM) and Set Asynchronous Response Mode Extended Format (SARME) commands on any one circuit shall not be implemented. Any given circuit shall optionally be capable of operating SARM or SARME but not both. The command for the mode selected shall only serve as an reset of the protocol.

The contractor shall consider and plan for interfaces to the DCIN IV network. DCIN IV is a command and control communications network which processes messages of various classifications. The contractor shall consider for the interfaces necessary to allow the SCOMP to serve as network processor in the DCIN IV network.

The contractor shall also consider and plan for the interface to the secure communications controller being developed under the sponsorship of the Communications Security Engineering Office (CSEO/TCA). This controller will integrate controller and communications security (cryptographic) functions. The controller will be capable of simultaneously servicing several remote terminals and eliminate the need for one communications security device at the local site for each remote site. The confidential "Statement of Work for Project T520/01/01 Secure Communications Controller (17 Apr 75)(U)" describes the effort involved in the communications controller development effort. The government will provide information on the progress of the controller effort as the information becomes available.

4.4.2.2 Analysis Considerations. The contractor shall conduct a performance analysis for the SCOMP. The contractor shall identify SCOMP configurations relevant to its potential applications as a Multics secure front end processor and a network front end processor. The contractor shall provide quantitative estimates in critical areas of throughput and capacities for these configurations. As a basis for comparison the contractor shall provide similar estimates for a minicomputer without an SPU operating in similar environments.

To allow the kernel and operating system design subtasks to begin, the contractor shall deliver a processor manual for the combined minicomputer and SPU. This processor manual shall be issued

prior to the SCOMP Preliminary Design Review (PDR).

The contractor shall plan for the eventual test and evaluation of the SCOMP hardware using software designed and developed for the purpose of test and evaluation. The SCOMP hardware, especially the CPU, shall be fully exercised through this testing process.

4.8.2.3 Militarization Considerations. The contractor shall begin the design of a prototype/militarized version of the SCOMP and the Honeywell 6000/Series 60 IU in parallel with the effort described above. A design objective shall be to meet the RED/BLACK separation and TEMPEST requirements. Specifications for RED/BLACK separation are described in MIL-HDBK-232, RED/BLACK Engineering Installation Guidelines. NACSEM 5200 (dated 8 June 1972), which describes techniques which may be applied to the design of equipment to reduce or eliminate compromising emanations shall be used as a design guide. The specifications for TEMPEST are provided in DCA Circular 370-B195-2 and DOD Directive 5200.19.

The contractor shall begin to produce initial design specifications of the militarized prototype unit to be discussed at the SCOMP PDR. These initial specifications shall be issued as a technical note.

A future goal of this task is the fabrication, test, and evaluation of a militarized prototype version of the SCOMP. Testing will have to address RED/BLACK separation and TEMPEST requirements. When produced, the test plan and test procedure shall be subject to Air Force approval.

The ultimate goal is the delivery to the Air Force of a fully integrated, tested, and militarized SCOMP to serve as a SFEP to Honeywell 6000/Series 60 computers.

4.8.4 Design 6000/Series 60 Interface Unit

To support the use of an SCOMP as a SFEP, the contractor shall design and plan for the fabrication of a hardware interface unit for connecting the SCOMP to a Honeywell 6000/Series 60 large scale computers. The Honeywell 6000/Series 60 Interface Unit (IU) shall be based on the Air Force approved design specifications delivered under the Secure Communications Processor Architecture Study. The IU shall be modular in design so as to facilitate similar interfaces to other mainframes. The contractor shall produce interface unit description documentation in the form of functional specifications, to be discussed at the SCOMP PDR.

4.3.5 Initial Design of the SFEP Security Kernel

The contractor shall begin the design of a security kernel for the SCOMP. The particular application for which this security kernel shall be designed is the SFEP application. The SFEP security kernel shall be a suitable base for additional software to perform SFEP functions. Although the primary objective is a SFEP kernel, the contractor shall generalize the kernel design to permit application of the kernel in environments other than that of a SFEP. The security kernel shall be designed to meet the requirements of certifiability, efficiency, and effectiveness. The effort required to design the SCOMP security kernel is described in paragraph 4.5. The contractor shall produce top-level design specifications derived from initial kernel specifications provided by the Air Force.

The eventual goal of this effort is a fully implemented security kernel for the SCOMP based on approved final design specifications.

4.3.6 SCOMP Hardware Verification and SFEP Security Kernel Certification

In order to verify the SCOMP hardware, the contractor shall investigate the probability of compromise due to hardware failure. A technical report identifying the hardware verification techniques to be used shall be issued prior to the SCOMP PDR.

The contractor shall consider future security kernel technical certification. Based on established techniques and proven methodologies, this technical certification shows that all stages of the representation of the security kernel correspond to an abstract model of security. The certification effort required for the SFEP security kernel is described in paragraph 4.7.

4.3.7 SFEP Operating System and Applications Software

Based on the Air Force approved report of the task performed under Paragraph 4.3.1, the contractor shall begin the design of an operating system and applications software which will permit the SCOMP to function as a Multics SFEP. This software shall provide a user interface that is compatible with the Multics user I/O interface whenever possible.

4.9 Program Review/Management

The contractor is responsible for integrating all efforts of the various organizations and subcontractors toward the overall goal of a certifiably secure prototype general purpose computer system. The nature of the program is such that intermediate technical milestones are required which are to be decision points for the Air Force Program Manager.

4.9.1 Program Schedule

The contractor shall prepare and update as necessary a program schedule which reflects initiation of contractor efforts based on milestones visible to the Air Force. These schedules, initial and updated, shall be reviewed and approved by the Air Force Program Manager. Initiation of specific efforts will be based on agreement by the Air Force and contractor program managers that the prerequisite milestones have been achieved. The progress towards meeting the milestones of the program schedule shall be reported in both written status reports and program review meetings.

4.9.2 Technical Reviews

The contractor shall conduct two Formal Design Review Meetings (20 days after government receipt of the necessary documents) and other Technical Review meetings as required by the Program Manager. Location of all meetings will be at the Contractor's facility. The Contractor shall be required to provide the necessary resources and material to effectively perform the Review.

A Preliminary Design Review shall be conducted by the contractor at which time the SCOMP functional specifications, the Honeywell 6000/Series 60 IOU functional specifications, the minicomputer subsystem design analysis report, and an approach plan which identifies the hardware verification techniques to be used will be discussed. The initial design specifications for the militarized prototype SCOMP will also be considered at this review.

A Preliminary Design Review shall be conducted in conjunction with submission of the operating system design analysis and kernel specifications. At this PDR, the Multics security kernel functional specifications submitted shall be reviewed for their ability to preserve security as defined by the mathematical model while providing an efficient base for the Multics operating system.

The contractor shall conduct monthly review meetings and other review meetings, as required by the project manager, at the contractor's facility or some other mutually agreeable location. The contractor shall present reports of progress and projected tasks for completion of the efforts. Program coordination and technical guidance will be provided by the program manager (ESD/MCI), through the Administrative Contracting Officer. The contractor shall be responsible for the conduct of the Reviews in accordance with the following requirements.

4.9.2.1 Subcontractors and Suppliers. The Contractor shall be responsible for insuring that his subcontractors, vendors, and suppliers participate in Technical Reviews, as appropriate.

4.3.2.1.1 Contractor Requirements. The Contractor shall be responsible for establishing the time and place for each review, subject to coordination with the Program Manager. This shall be accomplished sufficiently in advance of each Formal Review to allow adequate preparation for the meeting by both the Contractor and the Program Manager. In addition, the contractor shall:

4.3.2.1.2 Insure that each Review schedule is compatible with the availability of the necessary information and contract articles.

4.3.2.1.3 Prepare for each Review in sufficient detail consistent with the scope and magnitude of the Review.

4.3.2.1.4 Designate a co-chairman for each Review. Participating contractor and subcontractor personnel or those chosen to make presentations shall be prepared to discuss in technical detail any of the presented material within the scope of the review.

4.3.2.2 United States Air Force Participation and Responsibilities. The Program Manager participates in each review to the extent specified below:

4.3.2.2.1 Serves as co-chairman.

4.3.2.2.2 Invites personnel, as necessary, from affected organizations, e.g.,

a. Local Staff;

b. Command Staff;

c. Other Commands;

d. Other Government agencies;

e. General System Engineering/Technical Direction Contractor

Attendance should be limited to those who are knowledgeable and can significantly contribute to a particular review. Final selection of individuals is the perogative of the Program Manager.

4.3.2.2.3 Provides the name, organization and title of each participating individual to the Contractor prior to each review.

4.3.2.2.4 Provides formal acknowledgement to the Contractor of the accomplishment of each Formal Review within ten working days.

4.3.3 Data Management

All contractor data and reports are required as specified in the attached DD Form 1423. The contractor shall forward a copy of each letter of transmittal for data items directly to ESD(MC-1) Data Management Office, L. G. Hanscom Field, Bedford, Massachusetts 01730, at the time of distribution.

The contractor shall gather and maintain the necessary information to provide the meaningful program status reports. This information will include identification of the specific technical work units in progress by the contractor and his subcontractors. (These technical work units will generally be elements of work needed to accomplish a task or subtask of this Statement of Work.) Each report will include status of efforts toward the contract goals by tying the technical work units to these goals including how they contribute to accomplishing the goals and the near term contractor steps to accomplish these goals. It will also include assessments of how the various efforts are being integrated into a running capability. (It should be noted that test and evaluation of the prototype at one operational Air Force Multics site is the major thrust of this program). Information should be provided on all efforts underway under this contract.

The contractor shall report monthly on the impact of the development efforts on performance and compatibility. The baseline for performance will be that of Multics prior to the incorporation of the Access Isolation Mechanism. Initially, the performance impact of individual task activities will be quantitatively described. However, the goal is to develop an integrated quantitative system measure for incremental performance changes. User compatibility issues shall be related to the current Multics.

4.0 Documents Preparation

To facilitate the timely informal exchange of information the contractor shall use Multics text-editing facilities in the preparation of documents whenever possible.

4.2.3 Other

The Government will provide system engineering guidance and documentation. Technical review of this documentation will be performed by the contractor when specifically directed by the Air Force Program Manager.

5.0 Government Furnished Support

It is the responsibility of the contractor to provide the required remote terminals, communication facilities, and communication lines. The

Government shall provide limited permanent on-line storage space for contractor programs, data, and document text files.