Massachusetts Institute of Technology

Project MAC

Computer Systems Research Division

Proposal for

Engineering of a Computer System for which Security can be Certified by Auditing

Submitted to

The Advanced Research Projects Agency

and

The Air Force Electronics Systems Division

October, 1973

Edward Fredkin
Director, Project MAC

Jerome H. Saltzer
Head, Computer Systems Research Division

October 26, 1973

SUMMARY OF PROPOSAL

The Computer Systems Research Division of M.I.T. Project MAC proposes a research program to make possible the certification, by auditing, of the centrally protected core of a general-purpose, remote-accessed, multi-user computer system. The key to the possibility of certification is to develop a central core structure which is simple, yet which does not sacrifice any of the features of a modern computer utility system. Because security and privacy -- protection of information from unauthorized release -- cannot be assured without certification, it is argued that the proposed research is in an area of an important national need. The proposed research program involves evolution of an existing commercial computer system, Multics, which is easily modifiable, which has advanced security features, for which expertise is available, and for which cooperation from the manufacturer can be expected. Experience in several areas of this system suggests that one major review of the central core of the system will reveal important simplifications which were overlooked during the intense pressure of initial implementation. This evolution is intended to result in a prototype operating system with all the essential features of the present Multics system, but with a small and simple central core, susceptible to line-by-line auditing by an expert. The work will be done as part of an ongoing program of research in computer systems and of related graduate educational programs. The work is estimated to require three and one half years, but will be done in a series of small steps with frequent benchmarks so that progress can be measured.

Massachusetts Institute of Technology

Project MAC

Proposal for

Engineering of a Computer System for which Security can be Certified by Auditing

Submitted to

The Advanced Research Projects Agency
and
The Air Force Electronics Systems Division

October 26, 1973

CONTENTS

APPENDIXES

ENGINEERING OF A COMPUTER SYSTEM FOR WHICH SECURITY CAN BE CERTIFIED BY AUDITING

Introduction

The Computer Systems Research Division of Project MAC is engaged in experimental studies on the problem of making the engineering of large-scale computer systems into a more methodical discipline. Its recent activities include the development, measurement, and documentation of the Multics system, and the attachment of that system to the ARPA network.

The Computer Systems Research Division proposes to undertake a major new research project. The goal of this new project is to make possible the certification that a large-scale computer system has been correctly implemented. Such certification has never before been achieved; in fact, most conventional computer operating systems are so complex that the concept of certification almost seems inapplicable. However, as will be explained, the central core of the Multics system appears to be susceptible to certification, if one wave of organization simplifications can be applied to it. The heart of the proposal is to make this wave of simplifications, reduce the size and complexity of the system core by an order of magnitude, and thereby render it certifable by auditing.

It is well-known that large-scale computer operating systems have a tendency to be extraordinarily complex, large in size, difficult to maintain, and awkwardly organized[1]. There seem to be several reasons for these tendencies, such as:

- Attempts to stretch the functional capabilities of the system as far as possible.
- Working in a hardware environment which was determined before software requirements were fully understood.
- Attempts to squeeze the hardware and software system to its absolute limit of performance.

- Attempts, because of the high cost of system development, to get the system "on the air" in the absolutely shortest time possible.

Of these four, probably the last two are the strongest contributors to unmaintainable and incomprehensible design, since they both encourage shortcuts to be taken and modularity to be violated against the better judgement of the system designer.

The net effect of these pressures is that most modern large-scale operating systems contain implementation errors which cannot be found except by running the system, and waiting for the errors to be exposed accidently. It is bad enough that timing-dependent errors which cause a system failure may occur just often enough to be disruptive, but not frequently enough to be easily reproduced by a benchmark test (e.g., every two hours.) In addition, there are at least two classes of implementation errors which may not routinely be noticed by running the system against a normal load:

1. Errors which result only in performance degradation of the operating system.
2. Errors in the implementation of the system's information protection mechanisms.

For both of these classes of errors, the system usually appears to correctly process its normal workload; only specialized tests (e.g., reproducible loads with known performance consequences, or test programs which attempt to violate the security mechanisms), which are not part of the usual workload, have any chance of revealing the difficulty, and even these tests cannot guarantee the absence of errors.

Need for this work

The current inability to certify the correctness of implementation of multi-user remote access computer systems is an important problem, with many immediate and practical consequences. Almost weekly one reads in trade journals about a new expression of public concern over computerized

health records systems[2]. or law enforcement systems[3]. or credit information systems, or federal dossier systems[5]. In addition to concerns expressed about the wisdom of information release policies, in most cases an issue is raised as to whether or not the computer system is capable of implementing the policy chosen. A second area, not yet subject to such widespread publicity, is industrial use of multiaccess computer systems. As such uses increase, corporate needs for privacy are being felt in the form of reluctance to automate certain functions. Overcoming such resistance to automation is probably one of the key motivations for IBM's announced $40 million effort to provide information protection facilities in future systems[6]. Finally, the defense establishment initially represents the largest "customer" for computer systems which provide information protection. At least two defense department committees have surveyed the state of the art, and concluded that one of the most important unsolved problems is the certification that system protection mechanisms work as intended[7,8]. The defense department probably has the largest immediate need for solutions: Air Force, Navy, and Army operations information is being rapidly upgraded into on-line, multi-user, remote access systems[9]. However, as the cost of computing, and especially on-line storage, declines, the economic advantages of on-line information processing are forcing information protection to the forefront of problems in the industrial and public sectors also.

As will be seen from Appendix A of this proposal, the need for information protection is sufficiently great that there are nineteen major research and advanced development projects in progress. The proposed work differs with all of the other activities, however, in its method of attack and its anticipated result, as described below.

Proposed Method of Attack

There appear to be three distinct strategies available for attacking the problem of reducing the probability of implementation errors in a computer operating system:

1. Use of a top-down constructive programming approach to the initial design of the operating system, following the principles of Dijkstra (structured programming[10]) and Mills (team programming[11]).
2. Subject all of the modules of the operating system, and its inter-module reference pattern, to the emerging techniques of program correctness proofs[4,12].
3. Modify an existing operating system so as to simplify its organization sufficiently that it may be subjected to line-by-line auditing by human examiners with high probability that they will detect any inconsistencies, errors, or subversions of the system.

The first of these three methods, while probably most appealing, has two interacting defects. First, there seems to be no way to release the top-down designer from the pressures toward complexity which were mentioned earlier, especially the pressure to get a system operational as soon as possible because of the development expense. Second, unless one can permit the top-down designer the freedom to discard functional requirements which are hard to fit into the design (that is, to simplify the problem being solved) then for a large system, a very long and expensive design period seems inevitable, during which time much thought and experimentation goes into devising schemes which methodically do everything called for in the system specifications.

The second method, correctness proofs, is in its infancy -- at the level of theoretical development -- and seems to be far from practical application to a set of programs as large as a typical computer operating system. Work is proceeding at many laboratories on expanding the capabilities of correctness proofs, but it will probably be some time before this technique is applicable.

An intermediate scheme, of constraining the system implementation to use only constructs which are demonstrably provable, may become useful sooner, but for the moment still apparently provides too many constraints.

For these reasons, the third technique, evolution of an existing operating system to simplify it to the point of auditability, is proposed as the method of this research. This proposal is based on the existence of a fully-equipped, operating, and easily evolvable system: the Multics system, previously developed by the CSR division of Project MAC[13]. Because of the special difficulty in discovering errors in the security and protection area, and because of the previously mentioned growing interest in that area both in private industry and in government and defense circles, the vehicle proposed for initial study is the "protection perimeter" of the system.

There are several reasons for choosing this particular system and this particular area. First, Multics has been developed from the ground up to be a protectable system, and it already contains protection mechanisms as advanced as any available, including special hardware features such as "protection rings"[14].* Second, the Multics system is better organized than most for evolution and modification, being relatively modular, being largely written in the PL/I language, and allowing initial checkout of most supervisor programs in a user environment[15]. Third, a pool of expertise in the system organization is available at Project MAC, and at Honeywell, whose continued cooperation is expected. Thus, productive work can begin immediately. Also, documentation of the Multics system is now sufficiently well-organized that new graduate students and staff members can rapidly learn about the environment. Finally, the result, if successful, can be

---

* Reference [20] provides an informal survey of the system's information protection mechanisms, and the Multics Programmers' Manual[13] describes most of the mechanisms in detail. One of the first tasks in the proposed project is to define more carefully these information protection mechanisms.

exportable. The present Multics system is a commercially available product[16]. and new ideas developed in the course of this research should be relatively easy to retrofit to the standard Multics system. Since it is intended that a working prototype be constructed, it would also be straightforward for, say, a government agency to request that a manufacturer turn the prototype directly into a production model. The relatively clean organization of the basic system will make the result suitable for study and imitation by designers of other operating systems in the same way that the present Multics system is already a subject for study and imitation in many locations in the U.S. and abroad. The specific area chosen, namely the centrally protected area of the system, is one for which there is much interest in understanding and exportation.

Although Multics is, if anything, already less complex in organization than most contemporary computer operating systems with similar functional goals, its original design was very methodical, and potentially supportable with mechanisms which are much simpler yet. The intense pressure of initial implementation did not permit time for contemplation and development of simpler supporting structures. The basic premise of this research is that one wave of simplification applied to the central core of the system will produce a badly-needed example of a significantly easier to comprehend structure.

The method which will be followed will be to build, by evolution from the standard Multics system, a new version. This new version is to have a "protection perimeter" whose contents are simpler in organization and much smaller in size than the current system. The present "protection perimeter" contains about 300 modules, or about 45,000 lines of PL/I and machine language* code. An initial target is to reduce both module count and the number of source lines by approximately a factor of ten. Although a change of this magnitude may seem at first to be unrealistic, there are several reasons why we expect that it is possible:

---

* Machine language code currently constitutes about 33% of the source lines, and about 10% of the executable object code in the protected area.

- Experience in working with Multics over the last several years has proven that its PL/I code and general structure make the system malleable. Recent drastic revisions such as new formats for directories, replacement of software by hardware rings, and even introduction of a new subroutine calling sequence have gone smoothly and quickly.

- Even before a comprehensive review has been undertaken, several specific possible simplifications have already been recognized which can strongly reduce the size of the most protected area. Again, past experience in reviewing and revising such areas as system initialization, interprocess communication, and drum and core management strategy, have proven that the opportunity for a more leisurely review of a design, after the initial rush to get a first draft design completely operational, can yield real insight into the necessary structure, and a resultant simplification and shrinkage of the mechanisms required for implementation.* In the present system, user terminal management, dynamic linking, and library search are examples of functions which do not need to be protected from the user to the same degree as, say, protection descriptor manufacture. Yet, they are so protected, and for the wrong reason; to maintain high performance, since they must communicate intimately with certain modules which must be protected, such as shared I/O buffer managers and directory format managers. These are examples in which the full effect of the newly-available ring protection hardware can be easily exploited to segregate into the more protected areas only the functions honestly requiring maximum protection. The ring hardware permits calls to procedures in a protected area to be performed with the same mechanism and at the same cost as a call to a procedure which is not protected.

---

* This effect has often been compared to the shrinkage in size and complexity of a typical mathematical proof between the time of the first discovery of the theorem and its later appearance in second and third generation textbooks.

- In the rush to make Multics fully operational, several well-understood shortcuts were taken. These shortcuts can now be reviewed, and redone, generally producing a more methodical structure. The most significant of this class of change is to permit multiple, parallel, processes in a single address space. Such a change would eliminate a vast number of special tricks used to provide rapid interrupt-time response, to provide a "quit" button on the user terminal, and to coordinate activities which are more simply viewed as parallel ones, such as management of all inactive telephone lines. The primary effect of this change, to reduce all hardware interrupt handlers to a half-dozen lines of code plus one subroutine call, is especially far-reaching in terms of simplicity and auditability of the resulting structure.
- Previously an exercise was undertaken to develop a paper model of a computer utility[23]. The resulting system description, based on experience with Multics, was at least an order of magnitude smaller than the corresponding description of Multics itself. This model has since been useful in training of undergraduate and graduate students in the issues of system organization; many of the ideas of that model are directly applicable to Multics also.

In light of these four reasons, it is anticipated that substantial progress toward a full order of magnitude reduction in the size of the most protected area can actually be accomplished. As a target, the following objective is proposed: the actual prototype operating system should consist of a small set of structured programs with annotation and commentary to make it easy for a reader to understand, and suitable for publication in a technical report or possibly (except perhaps for bulk) even in the "Algorithms" section of the Communications of the ACM. On the other hand, this publishable system

kernel should not restrict the present functional capabilities of the full Multics system in any important way. From past experience with such activities, it would appear that about three years from the time work begins, a prototype system may be available.

Of course, if auditing is to be successful, an auditor must have a precise definition of what functions are performed by the system he is auditing. This definition is, for Multics today, informal and imprecise, and one of the proposed tasks is to make this definition more exact.

Since it is proposed that the project be done by evolution of a current relatively easily modifiable operating system, the work can be broken up into small tasks, and progress will be assessable at all times during the course of the research. An initial set of plans, some fifteen specific tasks, is described in detail (and in Multics local jargon) in Appendix B.

Working Environment

An important difference between this kind of activity and the traditional activity of building or modifying an operating system is that when an unexpected problem is encountered, it must be mastered by completely consistent redesign rather than by shortcuts and patches, if simplicity of organization is to be a result. For this reason, we propose to work in an environment free from expectations by users of products to be delivered on a firm schedule. To accomplish this change of environment, a method of operation distinctly different from that used in the past to develop Multics itself is proposed. In the past, as rapidly as innovations were checked out, they were delivered and installed in the standard system, to be used immediately at M.I.T., and after some delay at the other sites operating the Multics system. This pattern will probably continue to be the method of Honeywell with respect to the Multics system. However, for purposes of this project, we propose to split off and develop a distinct version of the operating system, unhampered by compatibility requirements and day-to-day operational needs of the various Multics sites.

Close coordination has already been arranged with two Honeywell groups: the Cambridge Information Systems Laboratory (CISL) and the System Development Center of the Data Systems Organization (DSO). CISL has responsibility for development of the Multics product, and intends, wherever performance and compatibility constraints are not violated, to adapt ideas, techniques, or even changed modules resulting from this research project for installation in the standard system, since the payoff in improved reliability and maintainability should be immediate and quite high.

The System Development Center of Honeywell is interested explicitly in the development of a new version of Multics which adequately satisfies the protection needs of government customers, and it is also prepared to cooperate in the project. As indicated in appendix C, a separate proposal to this group in Honeywell requests funds for purchase of computer time to augment this proposal. SDC also plans simultaneously to engage in or support work in closely related areas, such as implementing a version of the government multilevel security structure within Multics, and interfacing encrypting equipment for communication lines to remote users. These activities are complementary to our proposal, and are significant because they indicate that Honeywell is clearly a potential path for making the results of this research quickly available to users.

Computation resources will be obtained by purchasing computer time and storage space on the M.I.T. Information Processing Center Multics site, to allow editing, compiling, generating, and storage of test systems. In addition, time on a distinct, small configuration of hardware will be required for checkout and performance benchmarking of test systems. Currently, such a configuration is available at the M.I.T. site, under control of CISL.

In addition to cooperation with Honeywell, close coordination with several other activities will be maintained. Most important of these are the "software assurance project" at the University of Southern California

Information Sciences Institute, the computer security activities of the Air Force Electronics Systems Division and those of the MITRE Corporation. Discussions with the people in charge of these projects have produced a general agreement to remain in close contact. At the present time, it would appear that the several organizations can supply distinct, unique and interlocking capabilities.

1. ISI can suggest that constraints on an operating system design which will make auditing possible, and can help develop standards for program organization and structure which will make auditing easier.
2. AF/ESD and MITRE have been sponsoring and performing work on mathematical and structural modeling of the security functions of operating systems, with a view toward interpreting real systems, such as Multics, in the terms of the models. Since those groups are located near to M.I.T., it should be possible to arrange frequent direct contact, so as to assure that the modeling efforts influence the course of the proposed research where appropriate. In addition, AS/ESD can provide the constraints on an operating system design needed by a user who has both operational and security requirements.
3. MAC/CSR can provide the experience in the real world of operating system construction to make sure that the prototype which results has a suitable functional capability, reasonable performance, and is maintainable.

In addition to these three organizations, communication is being developed with many of the other organizations working on projects related to information protection. The first step in this communication has already been completed, with the compilation and distribution of the survey of related work attached as Appendix A of this proposal.

Although the proposed work bears on mechanisms which can provide security of computer stored information, and is to be sponsored by organizations of the U.S. Department of Defense, the research project and its results are to be completely unclassified and publishable in the open literature. In addition, security clearance will not be required of project personnel, since there is no classified work going on in the area for which contact would be necessary.

In addition, it is assumed that relations with Honeywell will be carried on under confidential disclosure and patent rights agreements similar to those which have been successfully used for several years between M.I.T. and Honeywell. In essence, these provide for M.I.T. protection of Honeywell confidential product information, and for M.I.T. title to M.I.T. developed patents and copyrights. The products of the research will be unconstrained by confidentiality agreements and publishable in the open literature.

Collaborative Research and Education

The proposed project is part of a larger program of Computer Systems Research and education being carried on by the CSR division of Project MAC. These other activities are mentioned here to show how they reinforce and support this proposal. Slightly more than half of the energy and resource of the CSR division will be applied specifically to this proposal; the remainder is currently funded by the Advanced Research Projects Agency of the U.S. Department of Defense. Three projects are noteworthy:

1. The attachment of the ARPA network to Multics and participation in continued development of the protocols for use of the ARPA network has been a subject of increasing interest in the CSR division. This work has led to development within the group of a high level of expertise in data communication protocols, which will be useful in proposing simplifications to the telecommunication and network control system interface, both of which are currently felt to be unnecessarily complex, both in Multics and also in most other commercial systems. The possibility of directing all source-sink input and output activities through the network is especially intriguing, in light of the potential reduction in the number of different central supervisor mechanisms.

2. Measurement and modelling of the performance of computer operating systems occupies several graduate research assistants. This research has a bearing on the present proposal in that measurements often suggest simplifications in controlling algorithms for use of memory and processor scheduling. The simplifications result from measurements which tell exactly what properties of controlling algorithms are essential, and what properties are not significant. This activity also helps assure that new system versions developed as part of the work proposed here have no unexpected performance degradations.

3. Development of a large primary memory subsystem for Multics, for use in advanced research projects requiring intensive access to a wide variety of data is underway. This development would replace the present primary memory and paging store with a directly addressable high speed primary memory containing 4 or 8 million 36-bit memory words. This work is intended to pave the way for use of the predicted impressively low prices of integrated circuit memory in the next few years. This work bears on the current proposal in two ways: the resulting prototype system will be up-to-date when it is completed, ready to take advantage of

whatever memory technology and price is available then, and the current highly sensitive demand paging algorithms can be relaxed, and simplified, as part of the overall plan to reduce complexity.

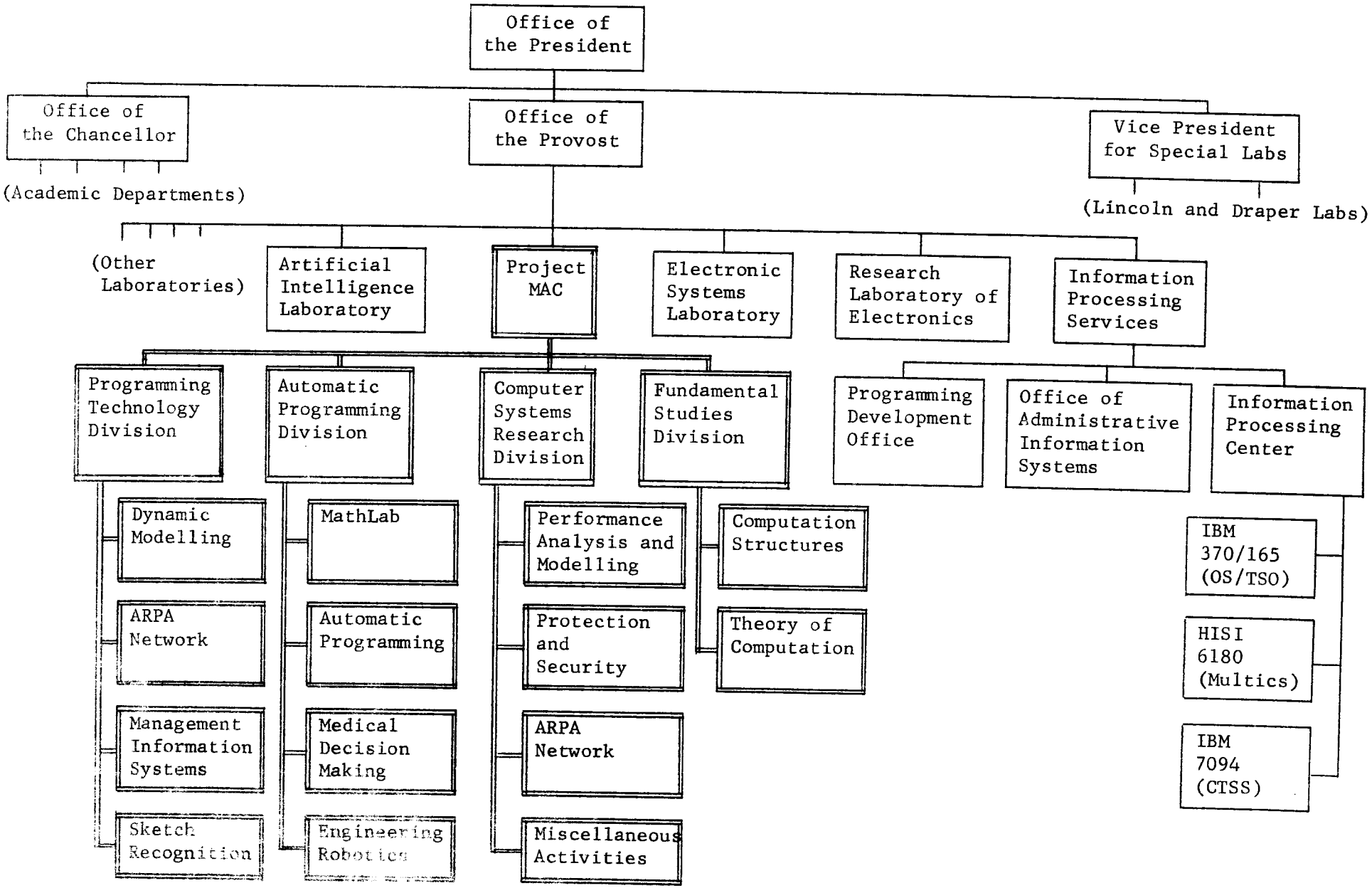
In addition to these three related research projects, the Computer Systems Research Division maintains both a graduate and an undergraduate "on-the-job training" educational program. Approximately ten graduate students and ten undergraduate students normally maintain contact with the division under this program. These students work together with professional staff members, providing a helping hand and also many new ideas, in return for the educational opportunity. These educational programs maintain a steady flow of students into government and industry who are knowledgeable about the problems of large-scale computer systems; they represent a major method of exporting knowledge gained in the course of research.

Background

Project MAC was founded in 1964, with support from the Advanced Research Projects Agency, to do research on man/computer systems. One of its first major projects was to complete development of the Compatible Time-Sharing System, for the IBM 7090/4 computers, into a polished tool, suitable as a vehicle for man/computer research. The Computer Systems Research group of Project MAC then designed a completely new time-sharing system, named Multics; from 1964 until 1972 the development, implementation, and improvement of that design was its major activity. This work was done in close cooperation with groups within Honeywell Information Systems (formerly part of the computer department of the General Electric Co.). Starting in 1971, attachment of the Multics system to the ARPA network was also undertaken by the group. In 1972 Project MAC was reorganized into four divisions, of roughly similar size, as indicated in the organization chart at the end of this section.

The Computer Systems Research division has had a longstanding interest in the protection of information stored in on-line, remote accessed computer systems. The CTSS system had some simple but significant provisions for information protection[17] and the Multics system was designed from the beginning with this object in mind[18,19,20]. The most recent work in this area consists of two doctoral theses, one on a method of implementing protection domains in hardware[21], the second on controlling authorization and restricting the use of information even after it is released[22]. An intensive study of the feasibility of developing a certifiably secure version of the central core of Multics, leading to this proposal, has been supported by ARPA: that support has permitted CSR to begin implementation of several of the tasks listed in Appendix B.

M. I. T. PROJECT MAC AND RELATED ORGANIZATIONS AT M. I. T.



5/8/73

(over)

Notes

(1) The faculty of Project MAC (and the other laboratories) hold appointments in an academic department. Most Computer Science faculty are in Electrical Engineering, but some are in Mathematics, the Sloan School of Management, or Architecture.

(2) There is currently a proposal to combine the three divisions of the Artificial Intelligence Laboratory with the four divisions of Project MAC, to produce a single Computer Science and Engineering Laboratory.

(3) Computer Science research is also carried on in sections of the Electronic Systems Laboratory, the Research Laboratory of Electronics, the Cambridge Project, and the Lincoln Laboratories, among others.

(4) The M.I.T. Information Processing Center operates three centralized computer systems as a service bureau for the M.I.T. community. In addition, many laboratories have their own private computers ranging in size from a DEC PDP-8 to an IBM 360/65. The Lincoln and Draper laboratories have their own centralized computer facilities. Estimates of the total number of computers at M.I.T. exceed 100.

References

[1] Naur, P. and B. Randell, Ed. "Software Engineering: Report on a conference sponsored by the NATO Science Committee, Garmisch, Germany, October 7-11, 1968," Scientific Affairs Division, NATO, Brussels 39, Belgium, 1969.

[2] "States Balk at U.S. Drug Plan, Say Privacy Rights Jeopardized," ComputerWorld VII, 33 (August 13, 1973), p. 1.

[3] "Governor Doubts Privacy For Records Tied to NCIC," ComputerWorld VII, 28 (July 11, 1973), p. 1.

[4] London, R.L., "The Current State of Proving Programs Correct," Proc. 25 ACM Nat. Conf. (August, 1972), pp. 39-45.

[5] Committee on Government Operations, "Privacy and the National Data Bank Concept," 90th Congress, 2nd Session, House Report No. 1842, pp. 1-34.

[6] "The SJCC," Datamation 18, 7 (July, 1972), pp. 59-62.

[7] Ware, W., et al., Security Controls for Computer Systems, Rand Corp. Technical Report R-609, 1970.

[8] Anderson, J.P., "Computer Security Technology Planning Study," Vol 1, ESD-TR-73-51, H.Q. Electronic Systems Division (AFSC), L. G. Hanscom Field, Bedford, Mass., 1973.

[9] "Management Procedures for the WWMCCS Standard Systems," Joint Chiefs of Staff Publication 17, September 1, 1971.

[10] Dijkstra, E.W., "Notes on Structured Programming," Technische Hogeschool, Eindhoven, Netherlands, 1969.

[11] Baker, F.T., "Chief programmer team management of production programming," IBM Systems Journal 11, 1 (1972), pp. 56-73.

[12] Elspas, B., et al., "An Assessment of Techniques for Proving Program Correctness," Computing Surveys 4, 2 (June, 1972), pp. 97-147.

[13] The Multiplexed Information and Computing Service: Programmers' Manual, M.I.T. Project MAC, (Available from the M.I.T. Information Processing Center.)

[14] Schroeder, M.D., and J. H. Saltzer, "A Hardware Architecture for Implementing Protection Rings," Comm. ACM 15, 3 (March, 1972), pp. 157-170.

[15] Corbató, F.J., J. H. Saltzer and C.T. Clingen, "Multics: The First Seven Years," AFIPS Conf. Proc. 40 (1972, SJCC), pp. 571-583.

[16] "Honeywell Introduces Commercial Version of Its Large Multics Computer System," The Wall Street Journal, Jan. 18, 1973, p. 10.

[17] Crisman, P.A., ed., The Compatible Time-Sharing System: a programmer's guide, 2nd edition, M.I.T. Press, Cambridge, 1965.

[18] Glaser, E.L., "A Brief Description of the Privacy Measures in the Multics Operating System," AFIPS Conf. Proc. 31 (1967, FJCC), pp. 303-304.

[19] Graham, R.H., "Protection in an Information Processing Utility," Comm. ACM 11, 5 (May, 1968), pp. 365-369.

[20] Saltzer, J.H., "Protection and Control of Information Sharing in Multics," submitted to Fourth ACM Symposium on Operating Systems, October, 1973, and the Communications of the ACM.

[21] Schroeder, M.D., "Cooperation of Mutually Suspicious Subsystems in a Computer Utility," Ph.D. Thesis, Department of Electrical Engineering, M.I.T., September, 1972; also Project MAC Technical Report TR-104.

[22] Rotenberg, L., "Making Computers Keep Secrets," Ph.D. Thesis, Department of Electrical Engineering, M.I.T., June, 1973; also Project MAC Technical Report TR-115.

[23] Clark, D.D., R.M. Graham, J.H. Saltzer and M.D. Schroeder, "The Classroom Information and Computing Service," Project MAC Technical Report MAC-TR-80, January, 1971.

APPENDIX A: ONGOING RESEARCH AND DEVELOPMENT ON INFORMATION PROTECTION

Introduction

Topics related to protecting information stored in computer systems have recently become a very popular subject for research; there appear to be at least nineteen major ongoing research and advanced development projects and perhaps ten smaller efforts in the general area. In addition to these research projects several different manufacturers are developing protection features for their commercial products, and many different government agencies have begun actively worrying about how to secure sensitive information already being (or about to be) processed by existing computer installations. In the following paragraphs are described all of the current projects so far encountered that are exploring new areas. Much of the work described here is government sponsored, and that work is largely under the Department of Defense. The topics are categorized by the organization performing the work; the organizations appear in no particular order. As an aid to fitting together the information, the report begins with a brief summary of the different kinds of activities being pursued, with forward pointers to the more explicit project descriptions.

Summary of the kinds of Activities being pursued:

The different kinds of activities related to information protection cover a wide spectrum and defy complete categorization, but it is helpful to summarize the various projects at different locations in terms of their intent. Following each summary is a parenthetical list of organizations doing something in that category. Nine categories cover most of the work reported here:

1. System penetration exercises. This kind of activity is directed at current systems, usually with the intent of pointing out the fallaciousness of premature claims that security has been achieved. Successful exercises have also the beneficial side effect of suggesting areas where more work is needed. (Lawrence Livermore Laboratory, Information Sciences Institute, IBM Corporation Research Division, Air Force Electronic Systems Division, System Development Corporation, and the Central Intelligence Agency.)

2. User interface studies. One of the chief problems of present protection systems is that they provide the user with an unacceptably complex environment in which to describe his information protection desires. Research in this area consists in trying out particular protection description schemes on real user communities to learn what responses occur. (IBM R.S.S. Project, Cornell University, Information Sciences Institute, M.I.T. Project MAC).

3. Proofs of correctness. Another major problem of present protection systems is the uncertainty as to whether they are correctly implemented. There are currently no acceptable methods of certifying the correctness of a typical operating system. Work in this area consists primarily in attempts to extend current, rather primitive techniques for proving assertions about programs to cover the constructs typically encountered in operating systems. (Information Sciences Institute, Stanford Research Institute, National Security Agency, University of Toronto.) N.B.: Many other organizations are working on proofs of correctness. The organizations mentioned here have particular interest in the correctness of protection mechanisms.

4. Mathematical models of protection kernels. Complementing the work on proofs of correctness is work on developing a mathematically consistent model of what it means to protect information. Such models might well suggest assertions which would then be input to proofs of correctness. (Air Force Electronic Systems Division, Mitre Corp., Stanford Research Institute, Case Western Reserve, University of Washington.)

5. Protection mechanisms. This activity consists of invention of appropriate architectural structures for protection of information inside the computer system. Problems being studied include structures for enforcing access to change protection specifications; structures which permit user-defined protected objects or protected subsystems; and capability systems. (Carnegie-Mellon University, University of Cambridge, University of California at Berkeley, Digital Equipment Corporation, University of Toronto, and M.I.T. Project MAC.)
6. Security in data communication networks. This activity is centered outside the computer system, but it involves such techniques as end-to-end encryption, which may require help from the computers using the network. (National Security Agency, IBM Corporation Research Division, and Air Force Electronic Systems Division.)
7. Data base facilities. Included here is work on integrating information protection facilities into data management systems. (Cornell University, Rand Corporation, TRW Systems, Rutgers University.)
8. Authentication mechanisms. Several groups are interested in developing more reliable methods of identifying and authenticating the user of a remote terminal. (IBM Research Division, National Security Agency, Air Force Electronic Systems Division.)
9. Department of Defense operational problems. A variety of activities are underway in support of current operational problems of various Defense Department computer-using organizations. (System Development Corporation, TRW Systems, Air Force Electronic Systems Division, and Central Intelligence Agency.)

As will be seen, not all of the work of the various organizations fits perfectly into these categories. Also, some categories such as "Data base facilities" are almost certainly incomplete, since almost every organization developing a data management system is also looking at protection specification mechanisms.

Carnegie Mellon University: HYDRA

As part of the ARPA-sponsored project to develop a multiprocessor computer from a mini-computer base, Professor William Wulf is leading the development of an operating system kernel named HYDRA. This system kernel defines a capability-like architecture which can identify and enforce usage constraints on a large variety of data types (extendable by the user of the kernel). Such a view provides a simple and consistent description of what is sometimes called a "protected subsystem" in other architectures. There are, of course, formidable design and implementation problems that are being explored by Wulf and his colleagues: the protection and stacking of the addressing environment has been worked out in detail; the integration to a user-level interface has just begun.

Although the work was begun in support of the multimini-processor project, the objective of exploring this protection architecture is a full-fledged goal in itself, and has already been the topic of a Ph.D. thesis, by Anita Jones. The current intent is to carry through a complete inplementation of this protection scheme in both hardware and software.

University of Cambridge: The Cambridge Capability System

Dr. Roger Needham of the Computer Laboratory of the University of Cambridge (England) is leading a hardware/software development project to create a new computer system based on a mildly restricted implementation of the "capability" view of protection. In the Cambridge Capability System there are two kinds of segments: those which hold data and those which hold capabilities. (A completely unrestricted architecture would allow capabilities to be placed anywhere.) Further, as many as four capability segments may be active at once in the sense of being available for indirect addressing. This architecture is quite experimental and raises a host of questions about how to develop a complete system around it; understanding and proposing answers to those questions seems to be the main research objective of the group. Complete implementation of both hardware and software are intended.

Lawrence Livermore Laboratory: RISOS Project

Under the leadership of Robert Abbott and sponsorship of ARPA, the RISOS Project (Research In Secured Operating Systems), at Lawrence Livermore Laboratory, in Livermore, California, is trying to develop systematic methodologies to be applied in testing the security of computer operating systems. A key element in the work is the application of computers in both the execution of a test and in the analysis of source code while preparing for a test. A PDP-11 computer is being used during the performance of security tests, while a CDC 7600 is used to assist in analyzing source codes of different operating systems prior to the start of tests.

The PDP-11 will contain a catalog of techniques which have permitted successful systems penetration in a variety of systems. This catalog is hoped to be useful in: developing a metric of a system's security features; helping to spotlight common weaknesses so that researchers and developers of secure systems will learn which areas need more work; and in providing for a while at least, a counter-measure for those present day systems which are retrofitted with security features.

The overall effort is intended to form the basis of procedures which will be used in the certification of the security of an operating system.

In an unrelated project, the Laboratory has developed a computer terminal access system for its CDC 6600-7600 complex which is claimed to be secure.

Information Sciences Institute: Software Assurance Project

Another ARPA-sponsored project is the University of Southern California Information Sciences Institute software assurance project. So far, only a small team is at work there, concentrating primarily on the extension of Ralph London's work on automatically proving assertions about a computer program. London hopes to be able to expand the range of program constructs he can handle to include those of parallel processing coordination, and to handle larger programs. Overall, the objective is to extend these techniques to the point that the central "protection" core of an operating system could be subjected to proof of correctness, as a method of certification.

Information Sciences Institute (Continued)

A second, independent effort at ISI is a system-penetration expertise developed by Richard Bisbey. This expertise was developed before the Lawrence Livermore Laboratory began its project; Bisbey is still performing some of the same functions that LLL is expected to assume when it gets underway. This expertise is being used to develop policies, tools, and techniques for testing and evaluating secure systems.

Also at ISI, James Carlstedt is attempting to formalize the expression of user protection requirements and the translation of such expressions into a general protection description and assertion language.

Case Western Reserve:

There are two projects at Case that are related to information protection. The first of them, under the leadership of Professors E.L. Glaser and C.W. Rose, the LOGOS project (sponsored partly by ARPA) is only indirectly related; its objective is to structure the design and make the implementation of hardware and software systems more or less automatic, thereby reducing the number of logical and implementation errors which make certification of correctness so hard to achieve. The second project, directed by Professor Kenneth G. Walter, is more directly on the subject of information protection: to develop a model of the central protection kernel of an operating system. This work is being done under a newly developing Air Force program on computer security.

Stanford Research Institute: A Demonstrably Secure Kernel

A small team at SRI, led by Peter Neumann, is designing the security-dependent kernel of an operating system. The initial aim is to verify the correctness of the design with respect to security, by as formal an approach as possible. This involves heavy use of structured design, a formal specification language, and structured verification techniques. Considerations of implementation are relevant, but the emphasis is initially on the design.

With respect to verifying the correctness of (implemented) programs, related work is being done by Bernard Elspas, Karl Levitt and Richard Walinger.

Stanford Research Institute (Continued)

At present a prototype system exists, whose performance is now being improved by at least an order of magnitude. Although not specific to security, this type of system could be helpful -- in addition to the above techniques for proving design correctness -- in eventually verifying an implemented system. Each of these projects is government sponsored.

Also at SRI, under a subcontract of the LLL RISOS Project, Donn Parker is developing a file of case studies of computer-related criminal and anti-social acts. This historical record is intended to be of value to workers in the protection field by providing information both on typical system vulnerabilities, and also about typical situations and personalities involved in misuse of computers.

National Security Agency: Computer Security Research Division

The National Security Agency, which already has responsibility for developing methods of securing government communication facilities, is beginning to develop an interest in the problem of securing information in computer systems. To this end, a research division on computer security, under the leadership of Hilda Faust, has been formed. Activities currently receiving attention include: design of a certifiably secure operating system, investigation of methods for proving program correctness, and development of communications security techniques applicable to computer networks. The division is also interested in methods of identifying and authenticating remote terminal users.

University of California at Berkeley: PRIME Project

An ARPA-funded project to develop an ultra-reliable protection system is underway at the University of California at Berkeley. As part of this project, Professor Robert Fabry is developing a software operating system with the property that all protection decisions are performed twice with different algorithms on independent hardware, and cross-checked. Such a strategy will presumably greatly increase confidence that the operating system is correctly implemented, and will also catch most of its own errors, thus making certification easier. It would also maintain protection integrity in the face of hardware failure.

University of California at Berkeley (Continued)

Fabry also is continuing to develop a better understanding of the organization of capability architecture, in which both he and the University of California at Berkeley have had a long-standing interest.

Cornell University:

At Cornell University, Professors R.W. Conway and W.L. Maxwell have been concerned with the appiication of privacy controls to data base systems. They have developed a model of the problem which separately identifies controls applicable at compile-time from controls applicable at run-time, and which includes the idea of controls which are data-dependent. (E.g., access is granted to examine all salaries whose value is less than $10,000.) This model has been implemented in a data base system which is now in production use, and used as a case study for instruction.

University of Toronto:

At the University of Toronto work on the design of secure systems was carried on under the auspices of project SUE. Currently there is an attempt to prove properties of programs including the effectiveness of protection mechanisms. In addition, an investigation is underway of existing systems with respect to security requirements and provisions. The work is under the leadership of Professors K. Sevcik and D. Tsichritzis.

IBM Corporation: Research Division

The IBM Research Division at Yorktown Heights has recently created a group under the leadership of Dr. Joel Birnbaum to explore protection problems. Three projects are currently underway. The first is a long-standing effort to develop a simple cryptographic system suitable for securing the communication line between a terminal user and his computer. A prototype system, called LUCIFER, has been developed, based on a hardware enciphering box at the terminal which accepts a 128-bit key on a magnetically striped card. At the computer end, the main processor performs enciphering and deciphering. The second project is exploration of the possibilities of using the dynamics of a hand-written signature as an authentication mechanism. A simple scheme for observing

IBM Corporation (Continued)

the dynamics of a handwritten signature has been invented, and the problem being explored is the reproducibility of the dynamics of the original signer as compared with those of a potential forger. The third project, under Dr. Laszlo Belady, is an analysis of the CP-67 system from a security point of view to learn what kinds of security holes may still turn up in a system which is basically designed to keep users totally separated from each other. This third project is intended to form the nucleus of a larger, more general attack on information protection problems.

IBM Corporation: R.S.S. Project

The IBM System Development Division has installed a special version of OS/360 MVT, known as the Resource Security System, at four sites. The system has two major modifications when compared with the standard OS/360: all known ways of "crashing" the system have been repaired, and a security compartment/category/classification system resembling the U.S. government defense security system has been added. The purpose of the study is to test the modifications in a live user environment to see what reactions, if any, are invoked. Specific questions under study are: What performance cost is encountered? How is the user not concerned with security affected? Is the security control system provided in R.S.S. adequately convenient and flexible? The test sites are the M.I.T. Information Processing Center, TRW Systems, The State of Illinois, and an IBM internal site in Gaithersburg, Maryland.

System Development Corporation: System Security Department

The System Development Corporation is engaged in a broad program of research and development in computer security, headed by Clark Weissman. Three separate efforts are exploring security problems in current, soon-to-be-available, and future computer systems.

The first effort, under Bruce Peters, is directed at the securing of existing, especially defense department, systems. This effort includes both penetration studies, and identifying security requirements of particular installations. One of the key systems under study is the World-Wide Military Command and Control System (WWMCCS) and its data communication network.

System Development Corporation (Continued)

The second effort, directed by William Shorberger, is intended to make possible specification of security requirements when procuring a new computer system. It includes developing operational policies for computer systems which correspond to already existing security policies of military and commercial organizations, and translation of these policies into system specifications.

The third direction, under Gerald Cole, is development of appropriate system architectures for future systems to use, and for practical retrofits to improve security on current systems. The group is also exploring the use of structured programming as an aid to producing provably correct operating system software.

Air Force Electronic Systems Division: Security project

The United States Air Force Electronic Systems Division at Hanscom Field, Massachusetts, has begun a program, under the leadership of Major Roger Schell, to provide security in Air Force computing systems. This program is planned to be quite wide-ranging, although it is just now gathering momentum. Among the projects now underway are: development of mathematical models of the security kernel of an operating system; development of simple cryptographic facilities for securing computer input/output lines; constructing a demonstration prototype of a magnetic strip "credit card" authentication system; and establishment of security requirements and system changes needed by a Honeywell Multics system to be installed at an Air Force Pentagon site. A technology study, performed by a panel of outside consultants, chaired by Professor E. L. Glaser, has recommended a seven-year, multi-million dollar activity to provide security for Air Force computing systems; the Electronic Systems Division is proposing to carry out the activities recommended in the technology study.

MITRE Corporation:

The MITRE Corporation, under Dr. Steven Lipner, is working closely with the Air Force Electronic Systems Division, and in addition is pursuing several related projects. These are: design and implementation of a provably secure kernel for a DEC PDP 11/45 computer, formal modeling of security aspects of computer systems (Multics in particular), two micro-programmed emulations of

MITRE Corporation (Continued)

the Honeywell 6180 as part of security architecure studies, application of virtual machine techniques within a security kernel, application of a certifiably secure kernel to Multics, and development of a prototype secure data management system for the PDP 11/45.

RAND Corporation: Privacy in Data Banks

The RAND Corporation is performing a study, funded by NSF and led by Dr. Rein Turn and Dr. Mario Juncosa, on aspects of maintaining privacy of personal information stored in data banks. Two directions are being pursued. The first direction is the application of mathematical analysis to privacy problems. Methods from information theory are being used to analyze irreversible privacy transformations, and entropy theory is being used to measure the amount of protection afforded by statistical aggregation in a data bank. Methods of game theory are being applied to models of the interaction between a data bank protector and an intruder. Finally analysis of protection in centralized versus decentralized data bank systems is being performed, along with preliminary studies of the psychology of privacy problems.

The second direction is less theoretical, and includes developing a model which displays the privacy aspects of a data bank, and the threats it must face, developing models of protection systems, and developing measures of effectiveness of protective mechanisms, including privacy transformations which encrypt sensitive information.

In a related project, Dennis Hollingworth has been exploring methods of adding entrapment strategies to computer systems.

Rutgers University:

Rutgers University has an ARPA contract to do research on secure operating systems, concerned with the conceptual and technical problems involved in the specification, design, and validation of protection mechanisms in operating systems. The long-range goal is to automate the design and verification of protection mechanisms in operating systems.

Rutgers University (Continued)

This project, under Professors William Easton and Chino Srivasan is just getting started. One particular area to be explored is protection of individual records stored in highly indexed information retrieval systems, in which the indexes may be sufficient to reconstruct individual records.

TRW Systems:

TRW Systems, in Redondo Beach, California, has developed several activities related to information protection, primarily with respect to government classified information. Dr. Eldred C. Nelson is associated with these activities. In addition to participating as a study site in the IBM R.S.S. project, three other kinds of activities are underway:

1. A relatively secure operating system for batch jobs on the CDC 6600 has been developed
2. Under a contract with Rome Air Development Center, TRW is developing a computer security handbook for Air Force systems. It would help administrative personnel establish their real security requirements, and provide detailed specifications, for example, of threat monitoring techniques.
3. TRW has developed data management systems with separate control of access to files, records, fields, and even specific field values, and depending on the operation requested. Study is proceeding on the interface between the data management system and the underlying operating system: the object is to develop more carefully a model of the data management system's support needs.

M.I.T. Project MAC

The Computer Systems Research Division of M.I.T. Project MAC, under Professor Jerome Saltzer, has a variety of projects underway related to information protection. These fall into three categories. The oldest area is research on design and implementation of new protection mechanisms. The techniques being explored include protection implications of an addressing architecture with an address space so large that addresses need never be reused; methods of permitting mutually suspicious programs to be used in a

M.I.T. Project MAC (Continued)

single computation; and methods of imitating, in the computer, the restricted hierarchical control of information access found in most social organizations. The second area is development of better, simpler user interfaces to information protection facilities. Particular problems here include simplifying the interface encountered by the user with a routine information protection need; establishing reliable default settings for access controls on newly created objects, so that only exceptions require explicit consideration; and arranging so that a user can easily determine whether or not his access control specifications match his protection intents. The third, and largest, activity is developing a version of the central core of the Multics supervisor which is sufficiently simple and small in size that it could reasonably be audited for security troubles. The heart of this third activity consists of sorting presently protected supervisor functions into three categories: those not requiring any protection, those requiring protection but not implementing information protection (e.g., processor scheduling), and those which implement information protection. A key constraint on this work is that the functional capabilities of the present Multics not be compromised in any important way. This project is just beginning, is expected to require about three years, and is intended to produce a working prototype.

Miscellaneous other Activities:

A variety of other activities, on related topics but generally smaller in extent, are underway at other places. These are listed here for completeness:

- The Oberpfaffenhofen Computer Center in Wessling (near Munich) is devising an access control language to be used in a "national data bank" for the German Government.
- Within the ARPA network, interest has developed in one-way encrypting transformations to protect passwords to be passed through an otherwise insecure environment. Suggestions for such transformations have come from G. Purdy, of the University of Illinois, Dr. Arthur Evans of the M.I.T Lincoln Laboratory, and Major Roger Schell of AF/ESD.

- The Association for Computing Machinery convened a workshop in January, 1973, to discuss protection of information in computer systems and to write papers summarizing the state of the art. (The workshop proceedings have not yet been published.)
- The Central Intelligence Agency reported at the June, 1973, National Computer Conference that it has developed a team of specialists to test the security of its own computer installations.
- The Canadian software house of Sharp Associates, Ltd., has developed an APL-based file system which it claims to be secure. The system goes under the name of APL plus.
- The Dartmouth Time-Sharing System has recently been described as providing substantial protection from user-induced system failures.
- Gregory Andrews of the University of Washington, is attempting to develop a model protection system about which provable statements may be made. The system seems to have properties of both domain and capability models, and is to be demonstrated by implementation on an X.D.S. Sigma 5 computer system.
- James Anderson, a private consultant to several government agencies on computer system security, has been examining defenses against attempts to render a system unusable. He has also acted as a leader in helping define the government's requirements for secure systems.
- National Cash Register, San Diego, recently developed a secure transaction processing system based on a capability organization. For the moment, however, this work is not being pursued.
- At the Digital Equipment Corporation, Michael Spier has developed a domain model of protection and has implemented a working prototype on a PDP 11/45 computer system.

Summary of Locations and Level of Effort

Summarized below are the locations working on information protection, and a rough estimate of the current level of effort at each location, measured in professional man-years of work per year of real time. The estimates are mostly based on personal observation or hearsay, and are

therefore subject to considerable noise. They are nevertheless useful in assessing the breadth and depth of the work to be done at each site.

	Site of work	estimated man-years/year
1.	Carnegie Mellon University	3
2.	University of Cambridge	7
3.	Lawrence Radiation Laboratory	7
4.	Information Sciences Institute	4
5.	Stanford Research Institute	4
6.	Case Western Reserve	3
7.	National Security Agency	5
8.	University of California at Berkeley	3
9.	Cornell University	3
10.	University of Toronto	3
11.	IBM/R.S.S. project	?
12.	IBM/Research Division	6
13.	System Development Corporation	~10(?)
14.	MITRE Corporation	13
15.	Air Force Electronic Systems Division	5
16.	RAND Corporation	6
17.	Rutgers University	3
18.	TRW Systems	?
19.	M.I.T. Project MAC	8

The following sites are estimated to be operating at a rate of about one man-year per year or less:

20. Oberpfaffenhofen Computer Center
21. University of Washington
22. ARPAnet password work
23. ACM Workshop
24. Central Intelligence Agency
25. Sharp Associates/APL
26. Dartmouth Time-Sharing System
27. James Anderson
28. National Cash Register
29. Digital Equipment Corporation

Bibliography

It is difficult to construct a useful bibliography of work in progress, since written material is often limited to project proposals and progress reports, neither of which are widely available. For the reader interested in pursuing the literature on information protection, the following two references contain extensive annotated bibliographies of previously published work:

1. Bergart, J.G., M. Denicoff, and D. K. Hsiao, "An Annotated and Cross-Referenced Bibliography on Computer Security and Access Control in Computer Systems," Ohio State University, Computer and Information Science Research Center Report OSU-CISRC-TR-72-12.

2. Hoffman, L.J., "Computers and Privacy: A Survey," Computing Surveys 1, 2, June, 1969, pp. 97-103.

In addition, the Lawrence Livermore Laboratory RISOS Project has constructed a KWIC-index style bibliography of the literature in this area.

APPENDIX B: SUMMARY OF INITIAL TASKS

One of the goals of our initial study has been to lay out in more detail some specific tasks which will allow us to get started on the overall project of developing a certifiably ("auditably") correct protected core for Multics. This note describes several specific tasks which so far have been identified as plausible candidates.

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 us toward the goal of an auditable central core; and 2) discussions and negotiations with Honeywell 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 compatability or performance constraints of the standard system, and thereby force a development of a parallel version, which we might call Multics/A (for Multics/Auditable).

On the whole, it will be seen that 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. This general class of activity, when completed, will represent perhaps one-half to two-thirds of the intellectual and programming effort of the overall project. 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 "protection kernel".

The tasks so far identified are the following:

1. 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.
2. 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.
3. Removal of the "working directory" concept from ring 0. The comments regarding reference names apply to the working directory also.
4. 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.)
5. 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.

6. 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 planned as part of the task, to test its effectiveness.

7. 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 queueing

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 these facilities. The payoff in simplification of the central supervisor should be quite high.

8. 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.
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.
10. ALM program catalogue. A list of all protected programs currently written in ALM (the Multics Assembly Language) is being 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.
11. 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.
12. Use of 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.

13. I/O hardware architecture proposal. A key result of the thesis just completed by D. Clark 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 Clarks' 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.

14. Reconfiguration hardware proposal. 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 R. 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.

15. System description improvement. 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.

Some of the fifteen tasks described above are already being worked on; others may not be started for some time, depending on availability of personnel to do them. A detailed schedule of target completion times has not yet been developed.

APPENDIX C: BUDGET CONSIDERATIONS

The enclosed budget estimates are made on the basis of yearly costs in 1974. The level of effort planned is a compromise between two factors: getting enough people together so that a critical mass of expertise can be maintained, while not developing so large a staff that management control is hard to maintain. By selecting carefully a few very competent people, the size of the team has been kept quite low. To help maintain a steady flow of new ideas a relatively large participation by graduate students is planned. The proposed balance of staff, students, and work rate is based on seven years of experience in managing the development of Multics itself.

The budget for computer time is designed to permit use of the M.I.T. Multics service as the development, organizing, and staging tool. The amounts requested are bare minimums based on the experience of system programmers using Multics for similar projects. A separate proposal to the Honeywell Data Systems Organization requests additional support for computer time purchase in the amount of $150 K/yr. If that proposal is accepted, the total computer time budget should be adequate to provide an effective working environment. The budget estimate takes into account a 50% discount over regular I.P.C. computer rates, provided to Project MAC in return for its guarantee of bulk purchase.

Budget Estimate 1/1/74 - 6/30/77

Period 1/1/74 - 12/31/74

Salaries and Wages	Number	Full time Equivalent	Cost
Faculty	5	2.0	$ 37,850
Staff	6	4.4	59,350
Graduate students	7	6.0	49,710
Undergraduate students	5	1.0	6,240
Support staff (secretarial)	3	1.0	7,385
Total			$160,535
Overhead Fiscal Year 1974 - 28% (1/1/74 - 6/30/74)			20,460
Fiscal Year 1975 - 29% (7/1/74 - 12/31/74)			25,360
Employee Benefits - 17.3%*			18,095
Total Salaries and Wages			$224,450

Computer time, to be purchased from M.I.T. Information Processing Center

	Cost per month	
On-line computing - 1000 hr/mo @ $10/hr	$10,000	
On-line storage - 10K records @ .50/rec/mo	5,000	
Stand-alone configuration - 20 hr/mo @ $200/hr	4,000	
	$19,000 x 12 mo x 50%	$114,000

Other:	
Space	30,000
Terminals	16,000
Travel	5,000
Reproduction	3,000
Telephone	3,000
Miscellaneous	4,550
Total of "other" category	$ 61,550
Total estimated cost for 1974	$400,000

Total estimated cost for 1975	(assumes 5%/yr salary increases and increases in "other" category)	$415,000
Total estimated cost for 1976		430,000
Total estimated cost for 1977		220,000
Total project cost, 1/1/74 - 6/30/77		$1465,000

* Excluding students

APPENDIX D: PERSONNEL

Jerome H. Saltzer

Position:
Associate Professor of Electrical Engineering, M.I.T.
Head, Computer Systems Research Division, Project MAC

Degrees:
S.B. in Electrical Engineering, 1961, M.I.T.
S.M. in Electrical Engineering, 1963, M.I.T.
Sc.D. in Electrical Engineering, 1966, M.I.T.

Selected Publications:

"Traffic Control in a Multiplexed Computer System," M.I.T. Project MAC Technical Report MAC-TR-30, June, 1966.

"Some Considerations of Supervisor Program Design for Multiplexed Computer Systems," IFIP Congress '68 Invited Papers, North Holland Publishing Co., 1968, pp. 66-71 (with F. J. Corbató).

"The Instrumentation of Multics," Communication of the ACM 13, 8 (August, 1970), pp. 495-500 (with J.W. Gintell).

"Remote Terminal Character Stream Processing in Multics," AFIPS Conf. Proc. 36 (1970 SJCC), pp. 621-627 (with J.F. Ossanna).

"Technical and Human Engineering Problems in Connecting Terminals to a Time-Sharing System," AFIPS Conf. Proc. 37 (1970 FJCC), pp. 335-362 (with J.F. Ossanna).

"The Classroom Information and Computing Service," M.I.T. Project MAC Technical Report MAC-TR-80, January, 1971 (with D.D. Clark, R.M. Graham, and M.D. Schroeder).

"A Hardware Architecture for Implementing Protection Rings," Communications of the ACM 15, 3 (March, 1972), pp. 157-170 (with M.D. Schroeder).

"Multics -- The First Seven Years," AFIPS Conf. Proc. 40 (1972 SJCC), pp. 571-583 (with F.J. Corbató and C.T. Clingen).

"A Simple Linear Model of Demand Paging Performance," submitted to Communications of the ACM.

"Information Protection and the Control of Sharing in the Multics System," 4th ACM Symposium on Operating Systems Principles, October, 1973.

As an M.I.T. faculty member, Professor Saltzer has been active in the development of the undergraduate curriculum in Computer Science, including development of a subject titled "Information Systems". At Project MAC he was involved in the later development of the Compatible Time-Sharing System (CTSS) and all aspects of the design and implementation of the Multiplexed

Information and Computing Service (Multics). His research interests include processor multiplexing, privacy - achieving mechanisms, data communications, memory system organization, computer system performance measurement, and certification of operating systems. He is a consultant in the topic of computer systems to a variety of governmental and industrial organizations. The certification project is Professor Saltzer's primary research commitment.

. . .

Michael D. Schroeder

Position:
Assistant Professor of Electrical Engineering, M.I.T.
Member, Computer Systems Research Division, Project MAC.

Degrees:
A.B. in Mathematics, 1967, Washington State University
S.M. in Electrical Engineering, 1969, M.I.T.
Ph.D. in Computer Science, 1972, M.I.T.

Publications:

"The Classroom Information and Computing Service," Project MAC Technical Report MAC-TR-80, January, 1971 (with D.D. Clark, R.M. Graham, and J.H. Saltzer).

"Performance of the GE-645 Associative Memory while Multics is in Operation," ACM Workshop on System Performance Evaluation, April, 1971, pp. 227-245.

"A Hardware Architecture for Implementing Protection Rings," Communications of the ACM 15, 3 (March, 1972), pp. 157-170 (with J. H. Saltzer).

"Cooperation of Mutually Suspicious Subsystems in a Computer Utility," Project MAC Technical Report MAC-TR-104, September, 1972.

As an M.I.T. faculty member, Professor Schroeder has been involved in developing and teaching a subject on the structure of computer-based information systems and in developing and teaching a laboratory for undergraduates on computer systems performance measurement. His association with Project MAC started in 1967, where he has been closely involved in the Multics development effort. He played the major role in designing the hardware protection mechanisms in the Multics central processor, the HISI 6180. This work on protection mechanisms for computer systems continued in his recently published Ph.D. thesis "Cooperation of Mutually Suspicious Subsystems in a Computer Utility". Other research interests include techniques for organizing computer based information systems, and certification of operating systems. Professor Schroeder is a consultant to several industrial and governmental organizations on computer systems. The certification project is Professor Schroeder's primary research commitment.

David D. Clark

Position:
Research Associate, Department of Electrical Engineering, M.I.T.
Member, Computer Systems Research Division, Project MAC.

Degrees:
S.B. in Electrical Engineering, 1966, Swarthmore College.
S.M. in Electrical Engineering, 1968, M.I.T.
Ph.D. in Computer Science, 1973, M.I.T.

Publications:
"The Classroom Information and Computing Service," Project MAC Technical Report MAC-TR-80, January, 1971 (with R.M. Graham, J.H. Saltzer and M.D. Schroeder).

A Demonstration of the Multics System, Video tape, Project MAC, M.I.T., 1972.

"An Input/Output Architecture for Virtual Memory Computer Systems," Project MAC Technical Report MAC-TR-117, September, 1973.

Dr. Clark has been associated with Project MAC since 1966, where he has played an active role in the development of Multics. His research interests include structure of operating systems (especially in the areas of input/output design and the role of processes in the simplification of software structure), formal descriptions of computer operating systems, new uses of computer networks, and certification of operating systems. Dr. Clark has immediate responsibility for the direction of the professional staff of the Computer Systems Research Division, and is responsible for staff efforts on the certification project.

. . .

Fernando J. Corbató

Position:
Professor of Electrical Engineering, M.I.T.
Co-head, Automatic Programming Division, Project MAC
Member, Computer Systems Research Division, Project MAC

Degrees:
S.B. in Physics, 1950, California Institute of Technology
Ph.D. in Physics, 1956, M.I.T.

Selected Publications:
"An Experimental Time-Sharing System," AFIPS Conf. Proc. 21 (1962 SJCC), pp. 335-344 (with M.M. Daggett and R.C. Daley).

"The Linking Segment Subprogram Language and the Linking Loader," Communications of the ACM 6, 7 (July, 1963), pp. 391-395 (with J. McCarthy and M.M. Daggett).

The Compatible Time-Sharing System: A Programmers' Guide, M.I.T. Press, June, 1963 (with M. M. Daggett, et al.).

"Introduction to Time-Sharing," Datamation 10, 11 (November, 1964), pp. 24-27 (with E. L. Glaser).

"Introduction and Overview of the Multics System," AFIPS Conf. Proc. 26 (1965 FJCC), pp. 185-196 (with V.A. Vyssotsky).

"Structure of the Multics Supervisor," AFIPS Conf. Proc. 26 (1965 FJCC), pp. 203-212 (with V.A. Vyssotsky).

"Time-Sharing on Computers," Scientific American 215, 3 (September, 1966), (with R.M. Fano).

"PL/I as a Tool for System Programming," Datamation 15, 6 (May, 1969), pp. 68-70.

"Multics -- The First Seven Years," AFIPS Conf. Proc. 40 (1972 SJCC), pp. 571-583 (with J.H. Saltzer and C.T. Clingen).

Professor Corbató has achieved wide recognition for his pioneering work on the design and development of multiple-access computer systems. He was in charge of the design and implementation of the Compatible Time-Sharing System (CTSS) developed at M.I.T. and played the same role in the development and implementation of the Multiplexed Information and Computing Service (Multics) at M.I.T. He has been a member of Project MAC since its organization, where he was head of the Computer Systems Research Division for nine years. In 1966, the Computer Group of the IEEE presented him with the W. W. McDowell Award for his work in the development of time-sharing systems. Professor Corbató expects to serve as a consultant to the certification project.

. . .

Eiiti Wada

Position:
Associate Professor of Faculty of Engineering, University of Tokyo
Visiting Associate Professor of Electrical Engineering, M.I.T.
Visiting member, Computer Systems Research Division, Project MAC

Degrees:
S.B. in Physics, 1955, University of Tokyo
S.M. in Physics, 1957, University of Tokyo

Selected Publications:
"Monitor Systems," Journal of IPSJ 3, 5 (September, 1962), pp. 267-277.

"Implementation of a Time-Sharing Computer System," Journal of the Faculty of Engineering, University of Tokyo, Series A, Vol. 5 (1967), pp. 44-45.

"Some Features of Interaction in the Hitac 5020 Time-Sharing System," Hitachi Review 17, 11 (1968), pp. 428-431.

"TSS I/O Control Program," Journal of IPSJ 9, 6 (November, 1969), pp. 335-343.

"System Program Description Languages D32, D34," Proc. IPSJ 12th Programming Symposium (January, 1971), (with T. Simauti et al.).

"Algol N," Seorsum Impressum Ex Tom. XXI, Fasc. 1 Commentariorum Mathematicorum, Universitatis Sancti Pauli, Tokyo, 1972.

At the University of Tokyo Professor Wada is responsible for teaching an undergraduate subject on logic design, system design, compiler and monitors, as well as a subject on introductory programming and elementary numerical analysis. He also teaches graduate subjects on programming languages and software techniques. Professor Wada has extensive experience with software systems, dating from 1957. From 1964 to 1968 he was responsible for the production and maintenance of software for the Hitac 5020 computers at the University of Tokyo. During this period he cooperated with Hitachi, Ltd. to design the communication line controller, the address translator, and the modus operandi of the TSS operating system. His current research interest is the search for basic concepts of programming languages; he is participating in the development of Algol N. During his year's stay at M.I.T. Professor Wada expects to devote substantial research effort to the certification project, paying particular attention to the formal methods of describing the operating system.

. . .

The remainder of the personnel are professional staff, graduate students, and undergraduate students. The persons in the first two categories are all experienced in the design, implementation, and measurement of operating systems, most having spent several years of close association with the Multics development effort. While this research effort is not yet fully staffed, those presently involved are:

Professional Staff:

Rajendra A. Kanodia	M. Tech., 1968, Indian Institute of Technology Kampur; 5 years professional experience
Robert F. Mabee	4 years professional experience
Elaine L. Thomas	E.E., 1969, M.I.T.; 4 years professional experience
Douglas M. Wells	S.B., 1970, M.I.T.; 3 years professional experience

Graduate students:

Richard G. Bratt	S.B., 1973, M.I.T.
Richard J. Feiertag	E.E., 1969, M.I.T.
Bernard S. Greenberg	B.E., 1971, The Cooper Union
Douglas H. Hunt	S.M., 1968, University of Wisconsin
Phillipe A. Janson	E.E., 1972, U.L. Brussels
David P. Reed	S.B., 1973, M.I.T.
Victor L. Voydock	S.M., 1968, University of Illinois