PROCEEDINGS OF THE FIFTH SEMINAR ON THE DOD COMPUTER SECURITY INITIATIVE

NATIONAL BUREAU OF STANDARDS
GAITHERSBURG, MARYLAND

MAY 24-26, 1982

TABLE OF CONTENTS

FIFTH SEMINAR

DoD Computer Security Initiative

May 24–26, 1982

About the DoD Computer Security Initiative

The Department of Defense Computer Security Initiative was established in 1978 by the Assistant Secretary of Defense for Communications, Command and Control and Intelligence to achieve the widespread availability of trusted ADP systems for use within the DoD.

In January 1981, a Computer Security Center was established in the Department of Defense. The Center is assuming responsibility for the activities of the Initiative. In the spirit of the Initiative, the Center is attempting to foster the development of trusted ADP systems through technology transfer efforts and to define reasonable ADP system evaluation procedures to be applied to government-developed and commerically developed trusted ADP systems. This seminar constitutes an essential element in the Center's Technology Transfer Program.

In conjunction with the DoD Center, the National Bureau of Standards Institute for Computer Sciences and Technology, through its Computer Security and Risk Management Standards program, seeks new technology to satisfy Federal ADP security requirements. The institute then promulgates acceptable and cost-effective technology in Federal Information Processing Standards and Guidelines. The Institute assists the Department of Defense by transferring the interim results of its research being conducted under the Computer Security Initiative.

About the Seminar

This seminar continues the efforts initiated in previous seminars to promote awareness of the DoD Computer Security Program and to present the developments and successes in the trusted computer operating systems area. The fifth in a continuing series, this seminar is the first to be co-sponsored by the DoD Computer Security Center. The program is directed toward the user of Computer Security products and includes presentations on the tangible results of efforts pursued by the Computer Security Center. Previous meetings have emphasized the theoretical aspects of computer security. This seminar emphasizes the practical aspects and the verification system/software analysis tools being used.

The theme is Evaluating Computer Security—the process of studying and analyzing the potential security products of manufacturers and government to determine their security attributes and their suitability as trusted systems.

TECHNICAL PROGRAM

General Theme: Evaluating Computer Security

Monday, May 24

A.M. Theme — DoD Evaluation Criteria and Process. Session Chairman—Melville H. Klein, Director, DoD Computer Security Center.

Welcoming Address
James Burrows
Director
Institute for Computer Science and Technology
National Bureau of Standards

Keynote Address
LtGen Lincoln D. Faurer
Director
National Security Agency

Trusted Computer System Technical Evaluation Criteria
Col Roger R. Schell
Deputy Director
DoD Computer Security Center

Commercial Product Evaluation—Part 1
Mario Tinto
Chief, Commercial Product Evaluations
DoD Computer Security Center

Commercial Product Evaluation—Part 2
Anne-Marie Claybrook
Group Leader
MITRE Corporation

Computer Application Evaluation—Science Out of Art
Stephen F. Barnett
Chief, Applications Systems Evaluations
DoD Computer Security Center

P.M. Theme — Technical Support for the Evaluation Process. Session Chairman—Daniel J. Edwards, Chief, Standards and Products, DoD Computer Security Center.

Research and Development in Support of Trusted Systems Evaluations
Hilda Faust Mathieu
Chief, Research and Development
DoD Computer Security Center

Software Tools
James Tippett
Chief, Technical Support
DoD Computer Security Center

Panel Session — Computer Industry Reaction to the Trusted System Evaluation Criteria. Daniel J. Edwards, Moderator.

Panel Members
Steven Lipner - Digital Equipment Corporation
Terry Cureton - Control Data Corporation
Theodore M.P. Lee - UNIVAC
Lester Fraim - Honeywell Federal Systems Division

Leslie DeLashmutt – Data General

Tuesday, May 25

A.M. Theme — DoD Perspective and Policy Implications. Session Chairman—Marvin Schaefer, Chief Scientist, DoD Computer Security Center.

Information Protection in an Information-Intensive Society
Melville H. Klein
Director
DoD Computer Security Center

DoD Perspective on Computer Security
Stephen T. Walker
Director, Information Systems
ODUSD (C^3I)
The Pentagon

Computer Security Policies—Challenges and Prospects
Eugene Epperly
Computer Security Specialist
Security Plans and Programs Directorate
ODUSD (Policy)

Computer Security Considerations in the Computer Life Cycle
LtCol Lawrence A. Noble
Computer Security Policy Analyst
Directorate of Computer Resources
Headquarters, U.S. Air Force

Non-Descretionary Controls for Commercial Applications
Steven B. Lipner
Engineering Manager
Computer Security Advanced Development
Digital Equipment Corporation

Computer Security Considerations for the Financial Community
Morgan Morrison
Vice President
Manager, ADP Security Division
Security Pacific National Bank

(TEXT NOT AVAILABLE)

P.M. Theme — NBS/ICST Computer Security and Risk Management. Session Chairman—Dr. Dennis K. Branstad.

Computer Security and Integrity Technology
Dr. Dennis K. Branstad
Manager
Integrity, Security and Data I/O Group
Institute for Computer Science and Technology
National Bureau of Standards

The Computer Security and Risk Management Program
Dr. Stuart Katzke
Leader
Computer Security Management Group
Institute for Computer Science and Technology
National Bureau of Standards

Financial (Banking) View of Computer Security

M. Blake Greenlee
Vice President
CITIBANK, NY

Cost-Benefit Impact Analysis of Computer Security Standards/Guidelines: A Base Case Framework.
Dr. Marco Fiorello
President
Fiorello, Shaw and Associates

Computer Security in the Retail Industry
John Pricz
Manager
Security Center
Carter, Hawley & Hale

Computer Security Evaluation and Certification
Zella Ruthberg
Staff
Computer Security Management Group
Institute for Computer Science and Technology
National Bureau of Standards

Wednesday, May 26

A.M. Theme — Computer Security Evaluation—Now and Future. Session Chairman—Col Roger R. Schell

Bizarre Bazaar: An Approach to Security Technology Transfer
Clark Weissman
Chief Technologist
System Development Corporation

Meeting Policy Requirements Using Object-Oriented Systems
Susan A. Rajunas
Technical Staff
MITRE Corporation

Initial Performance Measurement Results for KVM
Paul Cudney
KVM Program Manager
System Development Corporation

(TEXT NOT AVAILABLE)

Multi-Level Security Today
Lester J. Fraim
Manager, TCB Development
Honeywell Federal Systems Division

An Update on Computer Security Activities at Digital
Andrew C. Goldstein
Consulting Software Engineer
Digital Equipment Corporation

A New Verification System
Gary Grossman
Chief Computer Scientist
Digital Technology Inc.

(TEXT NOT AVAILABLE)

Panel Session — Future Directions for Computer Security Center/Industry Cooperation. (DoD Computer Security Center Chiefs.) James P. Anderson, President, James P. Anderson, Inc., Moderator.
Panel Members:

Panel Members:

Marvin Schaefer – Chief Scientist
Daniel Edwards – Chief, Standards and Products
Stephen Barnett – Chief, Applications Systems Evaluations
Hilda Faust Mathieu – Chief, Research and Development
James Tippett – Chief, Technical Support

(TEXT NOT AVAILABLE)

WELCOMING ADDRESS

James H. Burrows
Director, Institute for Computer Sciences and Technology
National Bureau of Standards

Jim received a B.S. in Engineering from M.I.T. and an M.S. in Mathematics from the University of Chicago. He directed the development of large information systems and data management projects for the MITRE Corporation and the Lincoln Laboratory until 1972, when he became Associate Director, Office of Computer Resources, U.S. Air Force. As the Air Force's senior civilian manager for data automation, he was responsible for developing and implementing policies for ADP management, operations, procurement, and standards utilization. Jim joined NBS as the Director of the Institute for Computer Sciences and Technology in 1979.

I am pleased to welcome you to this Fifth Conference on the DoD Computer Security Initiative Program and to add a special note of welcome to General Faurer, Director of NSA and the Conference's keynote speaker. The Institute for Computer Sciences and Technology has been happy to work with the Department of Defense and its new Computer Security Center in arranging this conference. The Computer Security Center is a co-sponsor for the first time and has been the primary force in planning the technical program.

The theme of this conference, Evaluating Computer Security, is an important concern for managers throughout government and industry, not just the Defense and Intelligence communities. Incidents of computer crime and computer tampering that appear regularly in the newspapers highlight that concern.

The General Accounting Office in a recent review concluded that federal government information systems are highly vulnerable to fraud and abuse. GAO detailed losses of thousands of dollars in several government departments from direct thefts of funds, as well as other abuses involving unauthorized use of computers and data. While the drive to reduce fraud, waste, and abuse gives special emphasis to the need to protect information systems from abusive and unauthorized use, it is also essential to maintain the availability of processing. Managers, users, and clients will not trust systems that are not available when they are needed—whether because of input errors, hardware or software errors, or deliberate attack.

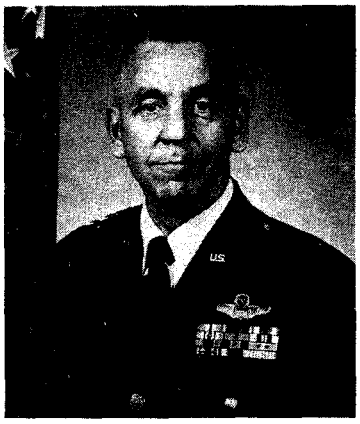
It is clear that the health and well-being of government and industry are dependent on correct, reliable information processing. Managers, users and clients of data processing systems must be assured of data processing integrity and reliability. Only then will systems be trusted by all parties to perform needed and critical functions. Development of trusted systems for the military has been the goal of the DoD Computer Security Initiative since its inception. Trusted systems for all those associated with information processing is a goal that we should work toward.

It is clear to us in ICST that developing trusted systems involves consideration of both technical and management issues, which I am pleased to see will be addressed by this conference. Security is an essential consideration in the design and implementation of a system, but planning for security does not end with the system itself. Audit trails, risk analyses, evaluation techniques, and complete system documentation, including the transaction flows outside the automated processes, all contribute to security. Perhaps the most critical components in a secure system are human integrity and reliability—a subject that clearly requires more study and understanding. ICST's current program of work is addressing user needs for standards and guidelines to improve the management and use of information processing. In the area of computer security we are focusing on developing techniques for evaluation and certification of systems, controlling user access,

and protecting data in the network environment. In many of these efforts, as well as in other parts of our program, we work closely with technical and management staff members of the Department of Defense.

Well-publicized incidents of computer tampering, or alleged computer tampering, will help to call attention to weaknesses and vulnerabilities of computer systems. We must respond by working together to design and develop computer products which include cost-effective security features. Only then will all of us—managers, users, and clients—receive the full benefits from the many uses of information processing. □

KEYNOTE ADDRESS



Lieutenant General Lincoln D. Faurer
Director, National Security Agency
Chief, Central Security Service

General Faurer received a B.S. from the U.S. Military Academy, a Master of Engineering Management degree from Rensselaer Polytechnic Institute (New York), and a Master of International Affairs degree from George Washington University (D.C.). He also attended the National War College at Fort McNair (D.C.). He received his pilot wings in 1951. Among his many assignments were Director, J-2, U.S. Southern Command, Canal Zone (1971-1973); Deputy Assistant Chief of Staff, Intelligence, Headquarters, USAF (1973-1974); Deputy Director for Intelligence, then Vice Director of Productions, DIA (1974-1977); Director, J-2, U.S. European Comman, Vaihingen, Germany (1977-1979); Deputy Chairman, NATO Military Committee, Brussels (1979-1981). He became Director of NSA in 1981.

This Conference is the initial opportunity for the DoD Computer Security Center as an entity to meet publicly with many of those from government, industry and academe with whom we have been dealing over the past nine months. Last September when I accepted an invitation to address the annual Washington Computer Conference of the Institute of Electrical and Electronic Engineers, I outlined the *goals* and *aspirations* of the newly formed DoD Center. Today, eight months later, I can report on its *performance* as the Center is off and running under the direction of Mr. Mel Klein. I am pleased with its progress, and trust that you will be too as you hear from its members during these proceedings.

By way of background, it is worth noting that the Center is a product of the DoD Computer Security Initiative established in 1978 under the aegis of OSD(C^3I). The fact that this is the Fifth Conference in a continuing security initiative series attests to both the vision and dedication of those responsible and the growing importance of *computer* security to the *national* security. Steve Walker, the mainstay of this initiative since its inception, has been instrumental in the creation of the Center. Because of these efforts, the DoD is better positioned to provide the direction and priority that computer security warrants. This conference selected what I find to be a most appropriate theme—"Evaluating Computer Security." It is clear that protection measures, at the heart of many information systems, achieve acceptability and are put into practice only to the extent that results of independent evaluations can unequivocally satisfy certification and accreditation authorities. The Center has taken the lead to provide criteria and tools to enable such evaluations of commerical products and selected DoD C^3 and intelligence applications.

Before focusing on the evaluation process, let me review the strides we have made over the past year in forming the Center itself. It has grown from an initial cadre of 34 people to it's present strength of 61. Since February, the Center has been housed in its own offices and laboratory facilities at our Baltimore Airport Annex which they are rapidly outgrowing. I appointed Mel Klein as its Director, taking over from George Cotter who had done a yeoman job during the Center's formulative stages. Now the Center has a staff which is growing to critical mass, contiguous space, a budget and the only thing it awaits is a formal charter. The DoD Directive formalizing the Center is in the final stages of coordination. My 1984 program objectives memorandum contains a major new thrust for a cooperative computer security research and development effort with the other DoD components.

Now, let me turn to the progress of the Center in "Evaluating Computer Security," focusing on three major areas:

— first, on the development of evaluation criteria for candidate products and systems submitted by industry,

— secondly, the on-going evaluations of computer security products where the Center acts as a catalyst and participant, and

— lastly, on the application of the criteria and evaluation results to the acquisition process.

A set of evaluation criteria is a primordial element in trusted system technology. As you in this audience are well aware, technical measures for achieving computer security present an exceedingly difficult challenge; complete solutions have often been elusive. Past pursuits along this line have frequently resulted in despair and grudging accommodation of the severe limitations imposed on operational capabilities. I believe we have now turned the corner. Given a set of carefully specified evaluation criteria, all can, and we in DoD will, consistently specify and evaluate the security capabilities needed most, such as reliable labeling and controlled, audited access while maintaining acceptable capability and peformance.

The Center has recently finished drafting such evaluation criteria. They are intended to provide an objective basis for judging the inherent integrity and trustworthiness of hardware and software security controls within a computer system. These draft criteria define distinct levels of increasing confidence, based on the substantial body of trusted computer technology that has been developed over the past several years. The mere existence of such security criteria for evaluation of products represents a significant step forward. What I find equally significant is the accompanying progress by industry in trusted systems development. For each and every level considered to be within the current state of the art, industry has produced at least one serious candidate. Furthermore, several of these are being pursued as standard products.

The development of these criteria has been a participative venture begun several years ago under the DoD Computer Security Initiative Program. The Center has been handed the baton to complete this work. Over the next several months, the draft evaluation criteria presented today will be reviewed and critiqued both within and between the government and industry. Your feedback will be used to refine the criteria to a set of suitable levels. These levels must be meaningful and realistic with respect to the range of policy distinctions that exist within the DoD, against the gamut of threats we face and the sensitivity levels of the information. They must respond to a wide variety of DoD needs from personal privacy and financial information to various levels of classification. The criteria must make clear to a vendor what he must do to attain a given level and to a user what security benefits derive from that level.

Even though the criteria are not final, we are making progress in applying them to product evaluations. The informal evaluations begun by C^3I as part of the DoD initiative have now transitioned to the Center. I am encouraged that not only has the impetus of this on-going effort continued undaunted, but also new interest has been generated.

Computer vendors are currently the only practical sources for hardware and software for the trusted systems we need. Industry participation and cooperation is essential. The products currently under active evaluation range from large-scale, general purpose computers to micro-computers. Our interest is, of course, heavily centered in standard product lines that are *readily available* and *fully supported* by the *vendor.*

The Center's evaluation teams are working closely with several vendors to apply the evaluation criteria throughout the design and implementation process; the security evaluation is *not* a one-shot activity applied to an already completed product. These teams are composed of experts from both government and industry. They not only critique the products but also provide technical consultation for alternatives and security improvements. The Center has on occasion highlighted other resources to further assist vendors, such as independent experts, seminars, and academic courses in the computer security field.

The Center will see that the other government users are made aware of formal evaluations through bulletins which provide the general status of our evaluations. Completed evaluations will be reflected in a published evaluated product list. The Center will publicize the level achieved and a summary of the principal security-related findings.

Furthermore, the Center is available to the national security establishment as a clearinghouse for information on potential security vulnerabilities of evaluated products. The Center is developing a mechanism for the reporting, monitoring and discreet, limited dissemination of this information. In these endeavors, we carefully respect the vendors' proprietary data.

Finally, I want to identify the context in which the results of our progress will be applied. Within the national security establishment, I expect the evaluation criteria and related support from the Center to provide a basis for much-needed self-discipline in the procurement of computer security capabilities. I want to be candid about the nature of our role: The Center's efforts are advisory, not directive, on DoD components.

Ergo, the evaluation criteria we are presenting today provide minimum standards that, to be effective, must be user-imposed in their requirement specifications and user-enforced in the acquisition process. As a practical matter, many installed systems will need to be "grandfathered." Over time we would expect the minimum security level required for new systems to increase to a level that provides greater and greater confidence in the security controls. Thus, though we begin at a minimal acceptable level, we will on an evolutionary basis advance toward significantly improved security. I want you to know that as Director of NSA, I intend to have the Center keep NSA and our substantial computer resources in the forefront of security demand. Furthermore, I expect the Center itself to be a showplace for computer security.

Occasionally we sell ourselves short and have a need which is matched by capability and yet that capability is not insisted upon in systems being purchased. For example, we are quite careful about classification markings for papers and documents. However, once the information is computerized, it frequently does *not* have either a *reliable* or *identifiable* classification label internal to the machine with anything like the confidence associated with the classification stamp at the top and bottom of a page in a document. Yet providing reliable classification labels integral to the computer and controlling the internal flow of sensitive information is clearly do-able: the Air Force's Multics system in the Pentagon has provided this capability for a number of years. We must push technology, not fail to use what exists.

In order to facilitate the application of our criteria, the Center will make the unique software tools available to the vendors along with data processing support for the formal verification aids that can significantly increase our, and the vendor's, confidence in the resultant products. In the near-term we are working to identify and control the configuration of a specific version of selected tools. We are also improving the user interface and documentation. To make these tools accessible to government, academic, and industry participants, each tool will be hosted on a computer that can be used from remote locations via government and commerical networks. Over time, this support will be improved.

Though the ADP market share for DoD is relatively small, the indirect leverage of our consolidated evaluation program extends beyond DoD. This conference is evidence of the interest of other communities in finance, government and industry. The similarity is obvious between the support processing needs for personnel, finance, resource control, etc., of DoD and those of other large organizations. Our growing DoD need for trusted computers in this support segment certainly suggests that the commercial sector will similarly find them useful for the needs of their commercially sensitive processing.

We will continue to pursue cooperative interaction with those who share common interests. We have noted a growing perception that the private sector, due to its dramatic dependence on automation, is exposed to the same kind of deliberate attack that has led us to emphasize the need for trusted computers. I would expect that our evaluated products list will in fact be used by the commercial sector as well.

In summary, the accomplishments of the Center and the rest of the community to date are encouraging. I believe we are on the threshold of an exciting era for computer security. For a number of years, computer security (or the lack of it) has had a growing negative impact on our ability to effectively use our computers. With the continued resolve of the OSD and the growing expertise of the computer security community, including the Center, we should be more articulate in our expression of needs, better able to evaluate security products and more confident in our mutual pursuit, with industry, of trusted systems. We look forward to building on the relationships developed under the initiatives program now being carried on by the Center. Thank you. □

TRUSTED COMPUTER SYSTEM TECHNICAL EVALUATION CRITERIA

COL Roger R. Schell
Deputy Director
DoD Computer Security Center

COL Schell received his BSEE from Montana State, MSEE from Washington State and PhD in Computer Science from MIT. He has served as software project manager, system engineer, and program manager for Air Force weapon system acquisitions. His technical experience includes design and implementation of dynamic reconfiguration for a commercial operating system and introduction of the security kernel technology that is the basis for current DoD research. He completed a three year assignment at the Naval Postgraduate School as an Associate Professor of Computer Science with research and teaching interests in the areas of computer security and operating systems.

Most of us are familiar with the fact that the problem of evaluating the security of computer systems has been with us for a long time. The Ware panel, headed by Dr. Ware of Rand in 1969 and 1970, looked at the area of computer security and published its report, which primarily served to identify that indeed there was a problem of computer security and that it was difficult to evaluate a system to determine whether it provided adequate security. Later on, the Air Force conducted a panel with Jim Anderson as the secretary. He prepared a report that identified some research directions to provide solutions to some of the problems identified by the Ware panel. In about that same time frame, the Department of Defense issued its directive and provided some policy regarding computer security. During the next few years, there was a substantial amount of research and development activity, as well as attempts to understand better the policy implications. Then in 1977, the DoD Computer Security Initiative efforts were kicked off to try and consolidate the gains that had been made. As was mentioned by General Faurer, the initiative program was largely the foundation for the current activities. In 1981, the Center was formed, and we have provided to you a draft copy of the technical evaluation criteria.

The foundations for the criteria are the notion that you know what it is you want to protect, that you have some mechanism for determining every time there's an attempt by some user, or surrogate for that user, to reference the information, and that the mechanism validates each reference. In order for that to be useful we have to have some model of what we mean by secure. A system which has a policy that says you can only access information on Thursdays may be secure with regard to that policy, but not secure with regard to one that says you have to have a secret clearance. So we need to have some model that defines what it is we want the system to provide by way of protection.

In addition, the notion of a reference monitor has with it some design requirements that are reflected in the criteria. One of these is that if I am going to have a mechanism that's providing protection, that mechanism must be tamper-proof, and must be able to protect itself so that it cannot be invalidated. Certainly, one technique that has been demonstrated in some of the penetration efforts is if you can once penetrate the system you may well be able to put in some form of trap door; in other words, you tamper with the mechanism itself so that the next time around you have your own bug and you don't have to depend on one that somebody else left for you. So this criterion of having the mechanism itself be tamper-proof is clearly necessary. It's also necessary, of course, to have the mechanism always invoked. Some of the early efforts at providing security retrofits have the notion that I could put an applique on a system and sort of tell the user that every time he is going to process sensitive information, he will always use this applique. Well, the problem is that there are a lot of ways of getting at the information without invoking that mechanism, and it really only serves to get in the way of those that were playing by the rules; unfortunately penetrators are not constrained by the rules.

And finally, a criterion in terms of a design requirement was that the mechanism should be subject to analysis and tests. This has turned out to be a rather difficult requirement and is one that has led to the evolution of the criteria that we'll be detailing today. In implementation, this reference monitor mechanism is going to be some piece of hardware and software that is what we refer to as a trusted computing base, and that subset of the system is going to be the principal part that will be the subject of the examination in applying the evaluation criteria.

One of the questions which comes up in looking at a set of evaluation criteria is, why do you have this set of criteria? Why not some other set? A group of people working on a particular approach, if they are allowed to provide the criteria, have a natural tendency to try to match the criteria against their particular thrust and activities at the moment. Now even within the government we face a similar sort of difficulty: why do we want a particular set of criteria that might be written down? In order to answer that question both for ourselves and for the reader, we've looked at the basic requirements that we're trying to address with a trusted computer system, and then from that derive the criteria that will allow us to judge how well the system meets those requirements.

One of the requirements is the marking of information. And in the Department of Defense it is clear that we depend very heavily on various markings of information. In the manual world, as General Faurer mentioned, we mark documents top and bottom of page (at least we're suppose to) to reflect the sensitivity of the information that is contained. In computer systems, we have the problem in many cases that the information is homogeneous with regard to markings. You can not pick up a magnetic tape and tell what "color" the bits are. There have been some conceptual schemes for coloring blue and red bits but I haven't seen any implementation approaches. That lacking, we need some sort of sensitivity-labeling system internal to the computer itself. This must be consistent with the same kinds of rules we have for protection outside the computer. Secondly, we have the notion of mandatory security restrictions that limit information access to authorized users. Classified processing uses the term "clearance" for the user and "classification" for the information. The policy is mandatory in the sense that you as an individual who might have access to secret information are (if you are going to follow the policy) mandatorily required to ascertain that somebody has a secret clearance before he can have access to that information. Even if he may need that information to do his job in the routine processing, you cannot grant him access to that information until he has a clearance. This mandatory restriction on the security must similarly be reflected in the computer system.

Now, a problem which occurs, and in some sense is really unqiue to computers as opposed to the paper and pencil system, is that of uncontrolled downgrading. This perhaps is most easily illustrated with the problem of trojan horses. If I have an editor on my system which edits a secret file, and in the course of editing that secret file it makes a copy for me, then that copy may be in an unclassified file that I can retrieve at a later time. In fact, in one of the early penetration efforts in an Air Force system, there was a nominal security solution which said that every time a classified job ran, its printout was very carefully routed to just a classified printer. The control program was "enhanced" by the penetration team so that whenever something went to the printer spool it sent a copy to the printer for you and also sent a copy for me. I could then, at a later point in time, submit a batch job and pick up all the copies that were saved for me. That problem of uncontrolled downgrading really is something that comes about because of the nature of computers. In the manual world if we have an individual controlling the information, we trust him to know better than to make an extra copy for his mother-in-law every time he looks at a classified document.

In addition to this mandatory requirement, we have a class of requirements which have been called discretionary security. I would hasten to add that it is not discretionary in the sense that you have the choice to either apply it or not, but it is discretionary in the sense that you can, as an individual who has control of classified information, exercise your discretion in determining if someone else has the need-to-know for that information (if he has a clearance). Then and only then do you release it to him. This need-to-know control is at an individual level, although individuals may be grouped into some organizational entity. This requirement to enforce need-to-know is perhaps closer to the usual and historical notions of controls in a computer system where you identify the users (the individuals) and you identify the access that they have to the information. Now, in order to implement these previous requirements you need to know who the individuals are that have access to the system; both in the manual world and the automated world there is a notion of individual accountability. For an individual who processes or otherwise accesses sensitive information, there should be a record of what action has been taken with regard to that access. For

example, if an individual exercises his authority to downgrade information after determining it is no longer classified, there should be a record of that, so that if this judgement is called into question it could be determined who in fact made that decision.

And finally, in order for the security controls to be effective, manually or otherwise, there must be some notion of continuous protection. If in the manual world you transport, on 3-by-5 cards, the files of the people who have security clearances and you don't protect that information, your friendly KGB agent may take the opportunity to put some additional cards in your file. What you want is continuous protection of those things which are responsible for the security. In other words, you want to control unauthorized changes. This is also a requirement in the computer world since if you're depending on the hardware and software in the trusted system to assure the protection, you have to have some way to satisfy yourself that what you've evaluated against your criteria is in fact the same hardware and software that is actually executing at any point in time. These requirements form the framework for the individual criteria.

Now the criteria are structured into a series of several different classes. These represent a progressive increase or increment in the integrity, and each one of these increments is intended to reduce the risk that one would be taking if he used that class of system to protect sensitive information. These increments are intended to be cumulative in the sense that as you look at each one, it includes all the requirements of the previous increment. These are organized into four major divisions that I will outline. The intention is that between each distinct class there is a significant jump in the capabilities provided and in the difficulties in providing these capabilities. Furthermore, it is expected that they will be part of a natural evolution path so that by choosing one of these classes you are not precluded from going further. And these criteria are based on the policy requirements that I just outlined.

These criteria are applied on a system basis. We're talking about trusted systems and we've divided these systems into four different divisions that are independent of any specific applications. These divisions apply to operating systems, or to operating systems and application combinations. For our current focus, it is not really meaningful to talk about applying the criteria to an individual component. If you have a badge-reader and you ask what division this falls into, that's probably not going to be a meaningful question to us with the current evaluation criteria.

These divisions begin with a place-holder as a starting point that says if you don't meet any other division, there is a division D that provides minimal protection not defined by any real criteria other than perhaps that the question was asked.

The next division, which we call C, provides discretionary security. It provides for individual authentication so that you know who the individual is, and provides some set of nominal access controls for the discretionary security. You recall that discretionary security associates access to an information object with a particular individual or class of individuals. In that environment, there is implicity a uniform information sensitivity. At this level, nothing has been said about maintaining mandatory security. This matches our use of systems in what we call "dedicated" mode, or "system high" mode. In this mode we assume that all the information in the system is at the same sensitivity level; e.g., it is all secret, and all the users have a secret clearance. This discretionary level is quite useful in maintaining the individual accountability for that kind of application.

This next division (B) is when the mandatory controls are added so that we have the internal labels. It is based on some precise statement of a mandatory policy that we are trying to implement and, of course, has the hardware and software that is an identifiable implementation of the reference monitor notion.

The final division (A) is what we call verified security, in which the formal verification methods are applied in determining the confidence that one would have in the controls. As you look at this set of divisions, you see that for many of them the same requirements apply. Everything from level C and above has discretionary controls, and both levels B and A have mandatory controls, so to a user these systems might look very much the same. The principal distinction being made between the divisions is the amount of confidence that one can have in the controls and the protection that they would provide.

Next, I will go over the individual evaluation classes that exist within the divisions I've just outlined. As you see there are a number of these that range from class D to class A2, and I'll address them individually.

Evaluation class D is easy to address. It says that the system has been looked at and in the judgement of the evaluators it doesn't meet any higher class. This is different from just saying a system has not been evaluated, in which case it could fall into one of the higher classes.

In the division that is called the discretionary division, the first class is class C1 and this is the first point at which we're really providing some degree of confidence in the hardware and software controls. From our first principles we note that it has to be self-protective. The authentication of the user must be provided so that you know who was using the system. You don't allow the system to be used by just anybody if you have sensitive information, and of course the whole notion of a trusted computer system is one in which there is sensitive information, either classified or unclassified, to be protected. And there is an explicit access control mechanism in which you have named users or groups of users, and you have named objects in the system so you can identify them. You have some understandable definition of the controls that are provided so that at any point in time, if you have information contained in the system, you can say here are the people who will be allowed to access this information and no others. The establishment of confidence that you've met this level is primarily based on functional testing. In other words, there's some nominal set of controls that have been advertised; you rather carefully go through and you test those controls to make sure that they in fact function as advertised. The system also has user and operational documentation. An example of a candidate for this class could be nearly any of the systems that are offered today. We mentioned the UNIX system as a minicomputer operating system that is not specific to any particular hardware base and provides these sorts of facilities. It's one which you could not evaluate with any formal verification methods or against other more detailed criteria. However, you would require a defined set of semantics for the protection, and you'd say yes I met those and they do allow me to distinguish the individual users and the accesses which they have.

The second class, C2, is one in which the resources are more heavily encapsulated, at least for a subset of the resources. This may not be applied to all the objects in the system; it may be principally the ones that you're trying to protect and have a direct interest in. They would be things like individual files or in some cases particular information, and in addition at this level you introduce explicit auditing requiremens so that not only have you said to the individual, identify yourself when you're going to use this system, but you can have a record of what that individual has done. I will emphasize that this auditing requirement is a selective recording of accesses. In the past, the Department of Defense has used auditing in various of its systems, and there are some installations which would lead you to believe that the principal measure of their security is the size of the stack of audit outputs in the corner. That, we think, is not a very meaningful criterion, and also there is probably an inverse relationship between the amount of audit information you have and the amount that it's telling you about the system in terms of what people really understand about what's going on. So there needs to be a selective way of recording the accesses that occur.

In addition to having that selective recording, of course, there will be a need to have some sort of tools for examining the audit record. Indeed, you may want rather vuluminous recordings to answer the questions at some later point in time, when you suspect a problem and say, "Whoever accessed this file?" But routinely you would not want to have all of that provided to a human user. So you need to have some facilities to reduce the data from your audit collection.

An example of candidates for this class of system are things like the RACF package that is added to the IBM MVS operating system. These kinds of packages allow you to provide a tighter control over the list of users that access the system, as well as provide some of the auditing faciliies. This provides the need-to-know controls that you would look for in a classified environment in a single-level mode of operation. Remember that in division C we're assuming that there's an implicit single sensitivity level for all the information. Fortunately this class of system matches fairly well to the commerical offerings for many systems of control on a basis of an individual's access to files.

Next, we move into the class where the mandatory controls are introduced as well. The first of this class, B1, is one that we call the label security protection. For this class we have an explicit model of the protection that is to be provided. In the case of the usual classified processing in the Department of Defense, the model is quite straightforward. It says that if you're going to access information with a given sensitivity level, say secret, you have to have a secret clearance, and that if you have access to secret information you are not allowed to downgrade that information unless you're one of those explicitly authorized to engage in downgrading.

There are identifiable security components in the system. The parts of the system outside what's called a "security perimeter" are shown not to be harmful. This represents a fairly distinct jump in capability from the previous class, in that it is now possible to focus one's look at the system more closely to a subset of the hardware and software. The task for the evaluators and the producers is to identify those parts of the system which could affect the security, and to show that the remainder is in fact not harmful. In some specialized systems where you have very simple applications, it could in fact be that the entire system is within this perimeter. In most cases we would expect to have applications in which we do not want to evaluate each and every program that can run. The importance of this division B of the criteria, I think, is particularly evident when you consider the problem of the maintenance and support of the system. If your evaluation is dependent on all the programs that are ever run on that processor for that system, that means any time there's a change in any program, you have to concern yourself with whether your evaluation is still valid. By limiting the portion of the system which is responsible for the security controls, you substantially ease that problem of evaluation.

At this level, the requirement is that the mandatory access controls be applied on all the storage objects in the system (the things which actually store the information), and that there be internal parametric labels for that information. By parametric we mean that you should be able, at a given installation, to identify what outside label you want for a given bit pattern. We have not restricted ourselves to just classified processing. In unclassified processing, it may be that you would have, for example, sensitivity levels for things that system programmers are working on, as distinct from those objects which are the actual personnel files. Those labels should be reflected in the system in a parametric way, so that in one installation my printout might come out saying "personnel sensitive," whereas in another installation, identically the same system with a different set of parameters, might print at the top and bottom of the page "secret" or "top secret." We think that this parametric nature of the labels is quite important in order for the system to have a broad application.

One of the questions to which it is difficult to get the right answer is how many labels we should provide for. As we look at our classified processing and we look at some of the past examples, it appears that at least 8 levels in a hierarchical sense and something like 29 different categories are an adequate set. I think it is important for us to come to a common understanding as to what is an adequate size, so that we don't have the problem of one DoD customer asking for a label size that is just a little bit bigger than that posessed by some system that has been evaluated to fall into a class B1 or above.

The evaluation of this system, in addition to the functional testing of the previous levels, will also include some penetration tests — deliberate efforts to look for ways to circumvent the controls that are provided. In addition, since at this level there has to be a notion of a security officer (somebody who is responsible for the facility), there must be a manual that indicates how you would use this system in a given installation. I think we've all been aware of the kinds of problems that have occured, even with the nominal controls that we have today, where in a given installation people by bad practice don't take advantage of what's there. The example that is frequently cited is when the manufacturers distribute an initial tape or disk (or whatever the form may be), there are frequently passwords provided for the administrators and assistants. The expectation is that the passwords will be changed so that only the administrator will have access. In practice, there have been several reported incidents in which people who were able to dial in or otherwise access the system have tried that administrator's password and found that it still had the default password that existed when the tape was distributed, and therefore they could access any of the information that they wanted in the system. In order to help alleviate that kind of problem, there has to be good documentation that tells the facility manager how to use the system. You will notice that in each of these classes, one of the emphases is on having suitable user documentation and operations documentation, so that we can interface with requirements of the external controls as well as the controls internal to the hardware and software.

An example of this kind of product could be built from any of the third-generation systems. One case in the past where that kind of effort was made was during the WWMCCS procurement, a number of years ago, in which I think at least three different vendors provided versions of their system which had this sort of security protection. Of course, the current Honeywell WWMCCS GCOS operated today in that environment would be an example of a candidate for a class B1 system.

The next class of systems, which we call structured protection, adds some additional requirements in the criteria, and the emphasis shifts more to the problem of actually doing the evaluation. This says that we focus on the evaluatability of the product by building it in a way which allows us to assess what level it really meets: in particular that it meets level B2. To do this, the internal structure of the system must be evident. There should be distinct storage objects that you can identify. There should be, within the control mechanism itself, functionally distinct modules and you should be able to identify which module is providing which part of the protection. Sometimes this is referred to as identifying the security-sensitive or security-related modules of the system. You may well still have an operating system which has, within the hard core of the system, components that are not security-related, but you should be able to identify which ones you were depending on. In addition, at this level we begin to see an emphasis on the problem of the unauthorized downgrading (that I talked about) from the fact that there can be channels through the mechanism itself for leaking the information from one level to a lower level. In class B2, the requirement is that these channels should be identified. You should be able to distinguish how a trojan horse might abuse the individual features to pass information. The most common example of the channels for passing information are codes. I will pick one for illustration which is a little bit far out but serves the purpose. If the trojan horse is trying to pass a bit of information from a secret-level job to an unclassified job, that secret-level job may bring the system to the point in which there's only one record of disk storage left of a given type. It will then use the fact that there's a record left to represent a 0 and the fact that there is no record left to represent a 1. Then if it wants to signal a 1 it asks for and gets that last record. If it wants to signal a 0 it leaves that record there. The unclassified job can ask for the last record and he's gotten the one bit of information depending on the response which is yes, you've got that record, or there's no more storage left. That particular example may be a rather low band-width path, but the channels should be identified so that you know what those paths are and if you decide to leave some, such as the one I've identified, you might want to audit that. In other words, any time you reach the point in which you have consumed the last record and you provide a message that says there is no more storage left, you make a record of that in the audit trail, and if you see a lot of those you may suspect that it is not being used for its usual purposes. So at this level the focus is on identifying the problems that might be there, and having the ability after the fact to determine that they have been used illegally.

The labels of the mandatory control should be enforced for all the visible resources. These resources are not just the explicit storage objects, but also you should have labels for communications lines, and you may even have labels for the packets that occur in packet switch interface. Whatever the resoures are, if you have information provided, there should be an explicit label for that information.

In terms of the objects outside of the system, such as the files and devices, the flow control should be enforced so that the editor which was trying to make a copy for you and a copy for me would not be allowed to make the copy for me. This kind of control of the unauthorized downgrading will provide a substantially increased assurance that the system is in fact protecting the information. In addition, at this level you want to have an explicit control over the storage residue. The problem of what to do when you release the storage is illustrated by some of the older systems. When you wanted to release the storage, a block on disk or series of blocks, you merely removed the indication that it was used and you left on the disk all the information that was previously there. The next user who would ask for a copy of that, or ask for additional disk storage, would get the copy of the potentially secret file that was previously contained on that disk. That kind of residue can not be permitted and has to be controlled at the B2 level.

I mentioned the concern for the continuous protection of the system. At this level there needs to be some explicit tools provided for monitoring the configuration changes. Recognize that we are very much interested in commercial products applying these criteria. We are not going to develop and maintain and support these items within the Department of Defense. But what we can do is look at the mechanisms which the vendors have for providing the configuration control, and we can also provide tools for doing things like comparing this version of the system to a previous version of the system.

An example of a system which might be a candidate for class B2 is the Multics system with its Access Isolation Mechanism, which was initially developed with Air Force sponsorship and has been run by the Air Force for several years in the Pentagon.

We move now to the next class, still within division B. This class, B3, is the final class in division B and is primarily addressing the remaining difficulty of evaluation at this division, which says, I still had a lot of

things that I had to look at in my, say, operating system or special application, that really didn't have anything to do with security. For example, in some systems I have mechanisms for linking programs in order to make an executable unit. Sometimes that linking mechanism can be quite complex and for convenience it may well have been put along with the basic security controls in the heart of the system. At the level of class B3, the intention is to separate out those various mechanisms, to have a simple, central encapsulation mechanism that allows us to separate those portions of the system that are security-sensitive from those that may provide some necessary and common services to the users but really don't relate to the security as such. This will tend to lead to a layered set of abstract machines in the same sense that the term is used in software engineering literature today. This will provide a protection at each layer for the sensitive modules and an ability to separate those layers. There may be additional protection besides just the layers. There may be partitionings within those layers, but it should be possible to remove from consideration most of those parts of the system that are not really protection-sensitive. In addition, at this level, there will be a good deal more attention paid not only to identifying the storage channels but also to closing the storage channels. This may, in fact, actually have an impact on the kinds of services that one can nominally provide across levels. In other words, in the case of the out-of-disk kind of information channel that I used as an illustration, it may be that one would provide separate allocations of disk for the secret and the top secret disk, so that you would not be able to pass the information between the levels.

Also at this level you begin to give more explicit consideration during the evaluation to the problem of denial of service. We have to be aware of the limitations of the technology today in that we do not have precise definitive models for denial of service, but we do have some increased understanding, I think, out of the software engineering work over the years, that structures such as we talked about here do tend to provide more reliable software. Reliability in software, of course, does not have the same meaning as it does in hardware. One is not used to finding software instructions failing. This sort of attention, however faltering, to denial of service reflects that in many applications we're concerned not only with unauthorized access and destruction of the information, but also with the fact that the services need to be available. This is particularly evident in some of the command and control environments we have in the Department of Defense.

We also need to have an increased trust in the authentication process. We have a trusted path that when a user is providing his authentication (say a password) to the system, he knows that he is really talking to the system at that level. This is related to the problem that has been illustrated by a well-known penetration approach: I will, perhaps, leave a terminal and put in a little program which asks for your password, but it is really my program. You come up to the terminal and say, oh well here it is asking me to log in; you will go ahead and log in. My program will ask for your password; you'll provide me the password and thereupon I will simulate a crash of the system and save for myself your password. This is a problem where there has not been a trusted path for providing the authentication. You really provided your authentication to a program I left behind and not to a system. You would like to have, at level B3, a trusted way of making sure that when you are authenticating yourself, you are authenticating to the system and not just giving your password free to some hopeful penetrator.

With these sorts of requirements, at this B3 class we will have a highly structured implementation of the design, and we recognize this again will represent a significant increment in the difficulty. There is not just a set of functional capabilities one can add, but at this point one has to build the system in a way which is subject to the evaluation for class B3. This evaluation process is somewhat more akin, I think, to the notion of quality assurance and looking at how the system is being developed than it is to a notion of testing a black box after the fact. As we get to this level, there really has to be an ongoing effort during development to achieve a meaningful evaluation.

In terms of candidates for this level, there was in the mid-70's an Air Force effort, jointly with Honeywell and MIT, that identified a version of the Multics systems that used this sort of a structured design approach.

That represents about what we believe is meaningful to do in the division B systems where we have the mandatory controls, and if you want to have an increased confidence beyond that, we believe you have to move to using some of the more structured verification support tools.

For verified design, at the current state-of-the-art, we apply some mathematical tools, which use formal models with explicit security theorems. You have top level specifications that you can subject to a lot of formal analysis: I will in quotes say "prove." You can look at the specification in a very systematic way, and gain a high degree of confidence by this analysis that if that specification is correctly implemented, you will in fact have provided the security perimeter that you wanted to enforce and the controls that you expected to have. In addition to this, you need to assure yourself that the implementation corresponds to the actual specification. I note that the method for doing this is going to be some sort of verification evidence. At the current state-of-the-art, I believe that mostly what we can expect is evidence. We are not going to have a wrapped-up proof that the system is secure, but we do have emerging tools that can provide us this evidence in various degrees of formality, both at the specification level and at the level of the implementation correspondence.

In addition, the configuration management continues to be of interest, and becomes of increasing interest as you move to these forml methods, because now you want to control not only the hardware/software mechanism, but also the specifications. You want to make sure that at any point in time you are always building to the same requirements, particularly since these specifications are probably going to be subjected to automated validation. This control must be throughout the life-cycle: during its design, development, production and distribution. That says we must conern ourselves (within the Department of Defense at least) with the trusted distribution of the system. We would have a version of the trusted computing base which we had established as meeting the requirements of class A1. This would be distributed in a trusted sort of way, which would typically be more than the commercial distribution mechanism of just sending it through completely uncontrolled channels; aside from whether or not it ever got there, this raises other questions of whether what you received was what was sent. This trusted distribution requirement reflects that as we move to this level we have increasing dependence on the system. We expect a decrease in the risk from using it, and therefore are more concerned about its vulnerability to subversion during the distribution process.

An example of a system which might be a candidate for this is a SCOMP processor that has been offered by Honeywell and has resulted from government sponsorship, particularly by the Navy.

If we want to move beyond the A1 Class, we move into a class A2, which we would like to define but really don't know how to get to: a fully verified implementation. We could view this as a goal for the class of systems in which we verify at a source code level. We have to have some supportive high order language to make this possible. We would also like to provide a good deal of trust all the way down to the hardware implementation. It is clear that a lot of the details have not been worked out, and I will not discuss it very much since I think we have some time to refine those criteria a bit further.

At this class, as we move into greater dependence on the trusted system for security, we also have increased concern for the development environment. This has implicity been there in some fashion before, but certainly as we have more trust we have to ask questions in terms of the vendor's environment. What kind of controls are you providing so that the local KGB agent can't come in overnight and substitute his version for your version, or take away your version, look at it, and bring back a new one a week later? In addition, if we are going to look at the implementation, we're probably talking about some automated generation of the test cases. This represents a goal that we would like to head toward: the most difficult and demanding class of the various classes.

I will just comment briefly on the kinds of applications that we would see for the criteria. You will be hearing more about this as we talk about the process of doing the evaluation using the criteria and the evaluated products list that we are currently working toward. This will be the result of our cooperative evaluation with the vendors. The published results include the bulletins as evaluations are ongoing, and the actual entry on the list itself.

We are also providing a set of generic application guidelines within the Department of Defense that would reflect the modes of operation. For example, the dedicated mode will need fewer controls, viz, a lower class than the multi-level mode in which you are much more dependent on the hardware/software controls. Part of the distinction between the various applications is whether there are unevaluated programs running. The issue of storage channels, for example, becomes more severe if you don't know anything about the source of the programs you are running than in the case where trusted people provide all of the programs. The sensitivity of the data will be a factor.

In addition to these applications of the criteria to the commercial products per se, the Center provides certification recommendations on the selected systems within the Department of Defense for a wide variety of applications, from computer centers to communications systems. We will consistently use the criteria in developing these recommendations, to say, for example, as part of the evaluation we assess this system to be a class B2 system. This allows us to have a common basis for communicating with our users with regard to whether it ought to be certified for use in a given mode.

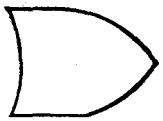
Finally, but not less importantly, we will be using these criteria in procurements. We expect that certainly in the procurements for which we are responsible, we will identify the level of trust that we expect in the hardware/software by identifying the class of system that would meet that requirement. In other words, in order to be responsive to a procurement, we would say that your system must be evaluated to, say, a class A1 system. That kind of use of the criteria, I think, will make it easier for us to be consistent in our specification from procurement to procurement, so that you don't have each procurement asking for off-the-wall kinds of standards. By requiring a specific class, we are confident that we can expect to receive a system with adequate security to meet our needs for that procurement. □

EVALUATION BACKGROUND

1970	**WARE REPORT — PROBLEM IDENTIFICATION**
1972	**ANDERSON REPORT — SOLUTION RECOMMENDATIONS**
	DOD DIRECTIVE 5200.28 — POLICY
1977	**DOD COMPUTER SECURITY INITIATIVE**
1981	**DOD COMPUTER SECURITY CENTER**
1982	**DRAFT CRITERIA**

CONCEPTUAL FOUNDATIONS

- REFERENCE MONITOR CONCEPT
  - VALIDATES EACH REFERENCE TO DATA
  - PRECISE MODEL OF AUTHORIZED REFERENCES
- DESIGN REQUIREMENTS
  - TAMPER-PROOF
  - ALWAYS INVOKED
  - SUBJECT TO ANALYSIS AND TEST
- IMPLEMENTATION — ''SECURITY KERNEL''
  - HARDWARE
  - SOFTWARE



SUBSET OF SYSTEM

BASIC REQUIREMENTS

- MARKING
  - INTERNAL SENSITIVITY LABEL
- MANDATORY SECURITY
  - ACCESS LIMITED TO AUTHORIZED USERS
  - CONTROLLED DOWNGRADING
- DISCRETIONARY SECURITY
  - NEED-TO-KNOW — INDIVIDUAL OR GROUP
- INDIVIDUAL ACCOUNTABILITY
- CONTINUOUS PROTECTION
  - CONTROL UNAUTHORIZED CHANGES

STRUCTURE OF CRITERIA

- PROGRESSIVE CONFIDENCE INCREMENTS
  - REDUCTION OF RISK
- CUMULATIVE CAPABILITIES
  - FOUR MAJOR DIVISIONS
  - SIGNIFICANT ''JUMPS''
  - NATURAL EVOLUTION PATHS
- BASED ON POLICY REQUIREMENTS

EVALUATION DIVISIONS

- DIVISION D: MINIMAL PROTECTION
- DIVISION C: DISCRETIONARY
  - INDIVIDUAL AUTHENTICATION
  - NOMINAL ACCESS CONTROL
  - IMPLICIT UNIFORM INFORMATION SENSITIVITY
- DIVISION B: MANDATORY
  - INTERNAL SENSITIVITY LABELS
  - POLICY MODEL
  - REFERENCE MONITOR IMPLEMENTATION
- DIVISION A: VERIFIED
  - FORMAL VERIFICATION METHODS

OVERVIEW OF EVALUATION CLASSES

- MINIMAL
  - CLASS D: MINIMAL PROTECTION
- DISCRETIONARY
  - CLASS C1: DISCRETIONARY SECURITY
  - CLASS C2: CONTROLLED ACCESS
- MANDATORY
  - CLASS B1: LABELED SECURITY
  - CLASS B2: STRUCTURED
  - CLASS B3: SECURITY DOMAINS
- VERIFIED
  - CLASS A1: VERIFIED DESIGN
  - CLASS A2: VERIFIED IMPLEMENTATION

EVALUATION CRITERIA - CLASS D

''MINIMAL PROTECTION''

- EVALUATED
- MEETS NO HIGHER CLASS

EVALUATION CRITERIA — CLASS C1

''DISCRETIONARY SECURITY PROTECTION''

- SELF-PROTECTING
- AUTHENTICATION OF USERS
- ACCESS CONTROL
  - NAMED USERS OR GROUPS
  - NAMED OBJECTS
- FUNCTIONAL TESTING
- USER AND OPERATIONS DOCUMENTATION

EXAMPLE CANDIDATE: UNIX

EVALUATION CRITERIA — CLASS C2

''CONTROLLED ACCESS PROTECTION''

- RESOURCE ENCAPSULATION
  - SELECTED OBJECTS, e.g., FILES
  - MAY BE ADD-ON PACKAGE
- SHARING BY EXPLICIT AUTHORIZATION
- AUDIT
  - SELECTIVE RECORDING OF ACCESSES
  - EXAMINATION FACILITIES

EXAMPLE CANDIDATE: RACF FOR IBM MVS/370

EVALUATION CRITERIA — CLASS B1

''LABELED SECURITY PROTECTION''

- EXPLICIT SECURITY POLICY MODEL
- IDENTIFIABLE SECURITY PERIMETER
  - EXCLUDED ELEMENTS SHOWN HARMLESS
- MANDATORY ACCESS CONTROL ON STORAGE
  - INTERNAL PARAMETRIC LABELS
  - 8 LEVELS, 29 CATEGORIES
- PENETRATION TESTING
- TRUSTED FACILITY MANUAL

EXAMPLE CANDIDATE: RETROFITTED THIRD GENERATION

EVALUATION CRITERIA — CLASS B2

''STRUCTURED PROTECTION''

- EVIDENT INTERNAL STRUCTURE
  - DISTINCT STORAGE OBJECTS
  - FUNCTIONALLY DISTINCT MODULES
  - CONFINEMENT CHANNELS IDENTIFIED/AUDITED
- LABELS ENFORCED FOR ALL VISIBLE RESOURCES
- FLOW CONTROL ENFORCED
- RESIDUE CONTROLLED
- CONFIGURATION CHANGE CONTROL TOOLS

EXAMPLE CANDIDATE: HONEYWELL MULTICS AIM

EVALUATION CRITERIA — CLASS B3

''SECURITY DOMAINS''

- SIMPLE, CENTRAL ENCAPSULATION MECHANISM
  - LAYERED, ABSTRACT MACHINES
  - SEPARATE PROTECTION - SENSITIVE MODULES
  - STORAGE CHANNELS CLOSED
- DENIAL OF SERVICE PROTECTION
- TRUSTED AUTHENTICATION
- HIGHLY STRUCTURED IMPLEMENTATION

EXAMPLE CANDIDATE: MULTICS ''GUARDIAN'' DESIGN

EVALUATION CRITERIA — CLASS A1

''VERIFIED DESIGN''

- FORMAL MODEL WITH SECURITY THEOREMS
- TOP LEVEL SPECIFICATIONS
  - IMPLEMENTATION CORRESPONDENCE
  - CONSTRUCTIVE FORMAL ANALYSIS
- LIFE CYCLE CONFIGURATION MANAGEMENT
- TRUSTED DISTRIBUTION
- FORMAL VERIFICATION EVIDENCE

EXAMPLE CANDIDATE: HONEYWELL SCOMP

EVALUATION CRITERIA — CLASS A2

''VERIFIED IMPLEMENTATION''

- VERIFIED TO SOURCE CODE
  - SUPPORTIVE HIGHER ORDER LANGUAGE
- TRUSTED DEVELOPMENT ENVIRONMENT
- AUTOMATIC TEST CASE GENERATION

BEYOND CURRENT STATE OF THE ART

APPLICATION OF CRITERIA

- EVALUATED PRODUCTS LIST
  - COOPERATIVE EVALUATION
  - PUBLISHED RESULTS
- GENERIC APPLICATIONS GUIDELINES
  - ''MODES'' OF OPERATION
  - UNEVALUATED PROGRAMS
  - SENSITIVITY OF DATA
- CERTIFICATION RECOMMENDATION BY CENTER
  - SELECTED DOD SYSTEMS
- PROCUREMENT STANDARDS

COMMERCIAL PRODUCT EVALUATION—PART I

Mario Tinto
Chief, Commerical Product Evaluation
DoD Computer Security Center

Mario received a B.S. in Physics from Fordham University, NY, and an M.S. in Physics from Catholic University, D.C. He joined NSA in 1960, and was initially assigned to the COMSEC organization as a TEMPEST engineer. Mario first became involved with computer security in 1972 with an assignment in the landmark DIAOLS test, and later participated in the follow-on WWMCCS testing of the Honeywell 6000. He provided computer security/systems security analysis support to programs such as AUTODIN II, AMPE, BLACKER/end-to-end encryption efforts, and WWMCCS ADP. At the Center he is responsible for encouraging commercial ADP vendors to introduce into the marketplace products which enforce DoD security policy, and for evaluation of those products against the evaluation criteria.

The next speaker and I will be discussing the commercial product evaluation process: what it is, what it is not, its purpose, and its elements. We will also be describing some of the details of how we see it being carried out. I will be addressing the purpose, context, and overall structure and approach of the process, and the next speaker will present the details of the process.

The product evaluation effort is interested in commercial products as examples of both technology and basic principles which have been incorporated into a "stand-alone" device and provide enforcement mechanisms for DoD security policy. This evaluation is aimed at off-the-shelf products, and is completely divorced from any consideration of potential applications or specific environments. Therefore, it is not a "certification" of a system composed of a computer with an application package, user mix, data mix, environmental factors, etc. This is an important distinction; the certification/accreditation process takes into consideration a good deal more than just a raw hardware/software device, and is concerned with a number of application-specific parameters. It is noted, however, that product evaluation and system evaluation are closely related processes. The results of the product evaluation are expected to be a major determinant in a myriad of systems engineering decisions, and certainly in the certification/accreditation process. Additionally, the utility of the evaluation criteria to the procurement cycle is obvious; we would expect to see system specifications call for the minimally acceptable level of computer product, or define the set of capabilities and basic principles to be supported. Clearly this calls for a "mapping function" which relates levels in the evaluation criteria to environments. Such a document is currently being developed, and will probably map levels from the evaluation criteria into selected generic environments.

COMMERCIAL PRODUCT EVALUATION

As the DoD developed an improved understanding of computer security issues and some of the R&D efforts had developed what then seemed to be achievable technologies, it naturally followed that procurement packages began to reflect these developments by incorporating requirements for security-related ADP functionality, as well as design, implementation, and assurance techniques. Whereas this may seem like a reasonable thing to have happen, it came to the attention of some (notably Steve Walker of OSD) that, in effect, we were attempting to design and build a secure operating system for each application. Also, this burden was falling not on the commercial ADP vendor, but rather on the system integration contractor. Clearly, DoD should not be in the operating system business, and certainly does not want to be in the position of designing and building a customized, "secure" operating system for each and every system acquisition. Rather, DoD wants to be able to purchase the requisite capabilities off-the-shelf.
That is, we should be able to satisfy computer security requirements for any specific system by engineering applications layers on top of a commercially available product which incorporates sound security principles, and which enforces DoD security policy at the most basic levels of the device. The goal, then, is to

encourage U.S. computer vendors to offer, as part of their standard product lines, devices which have been designed to enforce basic security principles.

Clearly, we will not be able to accommplish such a goal without the willing participation of industry. As a minimum, this implies that the DoD be able to convey, in reasonably well-defined terms, what we need. That is, we must be able to specify our technical requirements in terms which have more meaning than, "it needs to be secure." I believe that we are now at a point at which we can do this considerably better than we have in the past; we are able to state principles that need to be embodied (e.g., "least principle"), functionality that needs to be incorporated (e.g., internal labelling), and even specific technology and features which can be applied.

It will be noted that the process, although devoutly desired, and encouraged by us, is initiated by the vendor. (This is not to say that we will not actively elicit vendor participation in the program.) The process is technical; from the government's side, we expect to bring to the table a small, highly technical team of computer scientists and system security experts, capable of discussing design and implementation issues. Most importantly, the evaluation will be based on a known and widely publicized set of evaluation criteria. In fact, we expect that the vendor should be able to evaluate his own product and reach essentially the same conclusion as the government evaluation team.

I now want to discuss our present view of the evaluation process itself. We expect it to encompass two distinguishable stages: preliminary (or informal) evaluation, and formal evaluation.

PRELIMINARY EVALUATION

Again, note that the process is initiated by the vendor, although we expect to aggressively elicit participation by industry. The main purposes of the preliminary evaluation phase are:

—address design and implementation issues, and

—develop an initial assessment of how the product will measure against the criteria.

This stage can be characterized as follows:

(a) The product is in design or planning stages; the major technical activities are requirements definition, top level design, etc. Thus, we would normally expect to be addressing a new product, the next generation of a product, or the redesign or enhancement of an existing product. However, a vendor might desire to enter into a preliminary evaluation of an existing product solely in order to get a preliminary assessment of that product.

(b) The manufacturer is under no commitment/obligation to produce and market the product; we have no desire to interfere with corporate marketing decisions, but only wish to have the opportunity to influence a potential product during its early stages.

(c) There is no firm schedule during this phase; the effort will be driven by the vendor's need for interaction and by how the design and development of the product progresses.

(d) On the government's part, there is no commitment to conclude the effort with an official, formal rating of the product. Presumably, in many instances, there will be no commercially available product to evaluate. Thus any judgement relating to how the product fares against the evaluation criteria would normally be, at best, a projection of what the product might achieve when completed.

(e) The product will not be placed on the evaluated products list (EPL). The EPL is intended for products which can be purchased off-the-shelf. There may be a wrap-up report, but it will neither constitute a formal rating (i.e., against the criteria), nor will it place the product on the evaluated products list. Additionally, we expect that the level of effort that has gone into the evaluation per se, as well as the amount of technical detail available about the product during this phase, will be insufficient to warrant a formal statement about it.

Reports generated during the preliminary evaluation phase will be "Product Bulletins," or intermediate status reports, and their release will be fully coordinated with the vendor, not only for technical accuracy, but also to preclude the release of any proprietary or company sensitive data relating either to the product itself or to corporate plans and market strategy.

We see several benefits accruing from the preliminary evaluation exercise, primarily:

(a) Technology transfer: DoD over the years has made a sizeable investment in computer security technologies, mainly in the areas of formal specifications, design strategies/methods, verification, and implementation issues. We are anxious to share detailed information in these areas with the vendors.

(b) It allows the vendor an opportunity to gain insight into DoD security policy and requirements, and especially the implications of those in the ADP environment. This, in conjunction with the criteria, should allow us to discuss potential applications for the product.

(c) It focuses attention on security design issues at the proper stage in the product development, namely the requirements definition and design phases.

(d) It should allow the vendor to develop a target rating (against the criteria) and obtain a preliminary assessment from the evaluation team, thereby identifying shortfalls and making it possible to scope the effort necessary to achieve the desired goal.

We would stress the importance of beginning the preliminary evaluation as early as possible, since it identifies security design and implementation issues early in the product's development cycle, and allows reasonable goals to be set which will be based upon both marketing and technical factors.

FORMAL EVALUATION

The purpose of the formal evaluation phase is to evaluate a specific product against the criteria, culminating in a detailed, formal report, and placement of the product on the evaluated products list. For this phase of the effort we expect to make a heavy investment in resources and time. Therefore, we would expect to see:

—Product/market plans; there is every reason to expect to see the product available in the marketplace in the near future.

—Some pre-release, testable product; can be accessed or otherwise made available for testing, code and other documentation is available, and there are technically qualified individuals who can be queried about the product and be made available to support the evaluation team in learning about the product.

For the government's part:

—There is a committment to producing a format report and establishing a rating for the product. This means that once we agree to undertake the formal evaluation, it is our intent to finish it and publish the results.

—There will be a firm schedule established for the evaluation; we expect that the details of the schedule will be worked out jointly between the vendor and the evaluation team. Accordingly, we will commit the quality and number of people to the effort that is required to perform a technically demanding evaluation.

—The product will be placed on the evaluated products list, and the final report will receive wide distribution within the government.

The final report will be coordinated with the vendor for technical accuracy and to assure that it contains no proprietary data; otherwise the content of the report, and the determination of the final rating, will be at the discretion of the government.

In conclusion, I believe that the manner in which the product evaluation effort is being defined and organized can be characterized by some key words. The first is openness; we have no cards to play "close to the vest," the evaluation criteria are openly published, and they will be discussed and coordinated in great detail. Thus, the results of the evaluation, and how they are arrived at, are also open. The second keyword is cooperation; as General Faurer noted, "the commercial vendors are clearly the only practical source for the hardware/software products needed, and their active cooperation is essential." The DoD expects to bring to the table many man-years of experience in computer security and the development of related technologies, as well as insight into DoD security requirements for processing classified information in a variety of environments and applications. For his part, we are asking the vendor to share with us his development plans and designs in order to allow us a chance to influence his product line. We firmly believe that each of us has something to give and something to gain. □

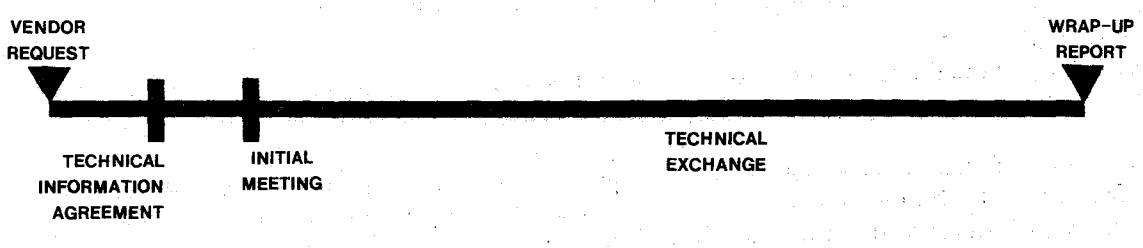
COMMERCIAL PRODUCT EVALUATION

MOTIVATION: MAKE IMPROVED COMPUTER SECURITY PRODUCTS AVAILABLE TO DOD ''OFF-THE-SHELF''; PROVIDE BETTER BASELINES WITH WHICH TO SATISFY DOD REQUIREMENTS FOR ADP SECURITY.

MECHANISM: INDUSTRIAL RELATIONS PROGRAM; ELICIT WILLING COOPERATION OF U.S. COMPUTER MANUFACTURERS.

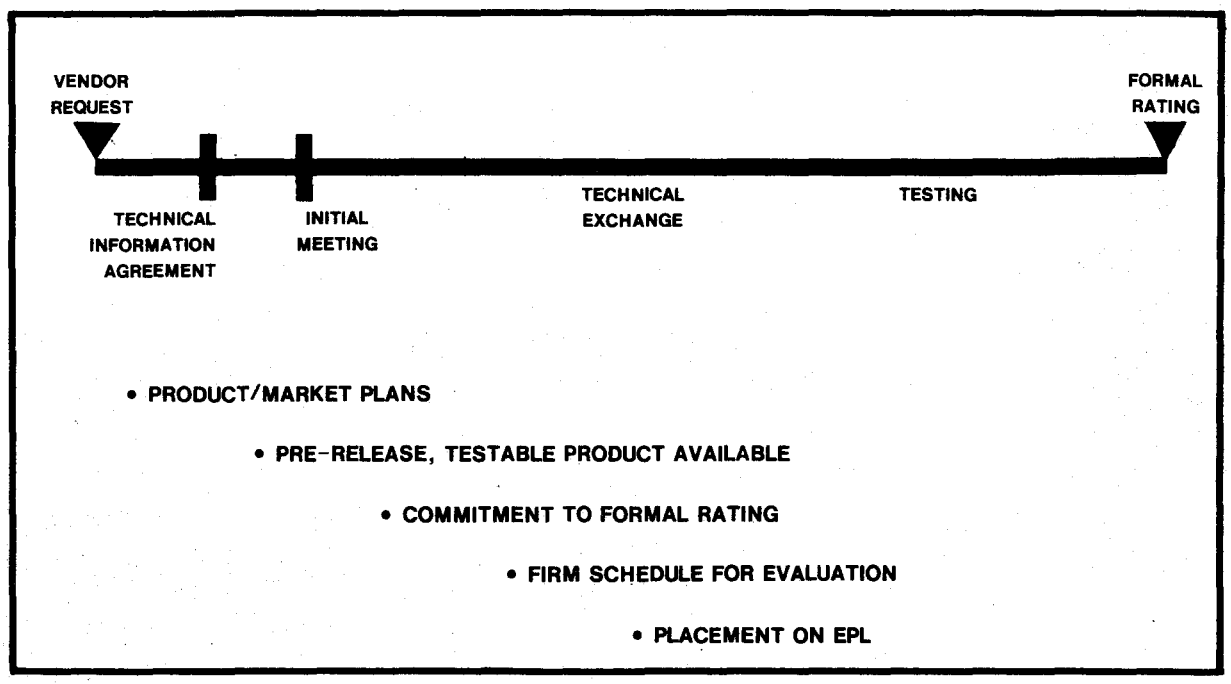
PROCESS: PRODUCT EVALUATION; TECHNICAL EXCHANGES INITIATED BY VENDOR REQUEST.

PRELIMINARY EVALUATION



- DESIGN/PLANNING STAGES
  - NO COMMITMENT/OBLIGATION TO PRODUCE/MARKET
    - NO FIRM TIME TABLE/SCHEDULE; LINKED TO PRODUCTION SCHEDULE
      - NO PLACEMENT ON EPL

FORMAL EVALUATION



COMMERCIAL PRODUCT EVALUATION—PART II

Anne-Marie Claybrook
Group Leader
MITRE Corporation

Anne-Marie received a B.A. in Psychology and an M.S. in Computer Science from the University of Connecticut. From 1973 to 1976 she was employed by United Technologies Research Center, Hartford, Connecticut as an applications programmer, and she was with NCR Communications Division in Columbia, South Carolina, from 1979 to 1980. She was also employed as an instructor in Computer Science at the University of South Carolina. In 1980, she joined the MITRE Corporation as a member of the technical staff. Anne-Marie is currently Group Leader of the Trusted Systems Evaluation Group. Her research interests include formal specification languages and transformation program verification.

The intent of this briefing is to pick up where the previous speaker left off and talk about the evaluation process in more detail.

I'd like to start off with a timeline for a preliminary and a formal evaluation and briefly go through each of the milestones or steps in the evaluation.

A vendor might initiate a preliminary evaluation in the design stage of a new product, in an enhancement stage for an existing product or to gain an initial assessment of a current product. By the time the formal evaluation stage has been reached, there is definitely a pre-release of the product available. For preliminary evaluations, some of the vendors we are currently dealing with include Control Data Corporation (CDC), Digital Technology Incorporated (DTI), IBM, Intel, Sperry-Univac and Tymshare, a time-sharing service bureau. We are just beginning a formal evaluation with Honeywell on their SCOMP system, now that there exists both a pre-release of the SCOMP software and criteria against which the SCOMP can be formally evaluated.

The first step in the preliminary evaluation is the vendor request. This takes the form of a letter sent to the Computer Security Center. Next, a non-disclosure or technical interchange agreement is signed. The purpose of this agreement is to protect any proprietary information of the vendor. Secondly, as we shall see, evaluation teams consist of members of government and various other communities. What is needed is one binding document for all evaluation team members. This one document covers the entire evaluation team.

Also at this time, there is some document exchange. The vendor needs to supply documents on the product or design under preliminary evaluation. What we are looking for here is some statement of protection philosophy for the system. Where such a document does not already exist, it may be possible to deduce the information from existing architecture and internals manuals. It is not necessary to see all existing manuals, for example, compiler and subsystem documentation. What is necessary is all security relevant information.

On our part, the Computer Security Center will furnish material on the evaluation criteria and DoD policy. For those systems that are aiming at the highest classification levels in the criteria, the vendor may request additional material or references on formal methodologies, specification languages or formal verification.

At about the same time as the signing of the non-disclosure agreement and the document exchange, evaluation teams for the preliminary evaluation are formed. For the Computer Security Center, team members are drawn from the Center, MITRE (a non-profit federal contract research center which works closely with the government), and government agencies such as the Department of Energy (DOE) and the Naval Research Lab (NRL). The reason for members from these and other government agencies is that

many government agencies possess a great deal of computer equipment and have employees who are experts on various vendors' equipment. Other team members come from academia and also from private research companies.

There are four kinds of expertise we are trying to bring together in an evaluation team. First, of course, is computer security expertise. Next is something I call experiential expertise, which is basically hands-on experience with the vendor and his products. Theoretical expertise, a high level or abstract understanding of the underlying principles of operating systems, is needed to judge if a system actually implements security principles at its lowest level of abstraction. The final type of expertise is formal expertise either in existing verification methodologies or in the mathematics necessary to do verification.

On the vendor's side, we also encourage formation of an evaluation team or group of people to be associated with the evaluation. First, there is the security person or evaluation contact within the organization. His role is particularly important in preliminary evaluations where there are many questions and requests for information that need to be routed to the correct technical people. Representatives from marketing management also need to be involved. Their input frequently affects the fate of new products and proposed enhancements. We find it very useful for marketing representatives to be present during the preliminary evaluation to assess and fold security into their proposed marketing strategies.

Finally, it is necessary to have technical expertise represented on the vendor's team, since we will be trying to determine what the design or system actually possesses in terms of security. The level of expertise should be that of a system architect.

After the groundwork has been laid, there is an initial face-to-face meeting. This usually takes place at the vendor's site. (The Center sponsors the Center's evaluation team at no cost to the vendor.) We discuss the security goals and philosophies of the Computer Security Center and those of the vendor. We try to explain what we've been discussing at this conference—the purpose of an evaluation, the evaluation team, and any questions about the criteria. We understand that the vendor's goal in building any product or enhancement is to make a profit, and we try to see the vendor's view of security, how it fits into his product and what section of the marketplace he is aiming at.

Generally, after receiving documentation, the evaluation team has a number of questions concerning design areas that need clarification or specific questions as to how some part of the implementation meets certain criteria requirements. At the initial meeting, we usually schedule a technical overview of the design or enhancement and go over any questions. We also talk about schedules and try to map out an evaluation timetable to follow the production or design schedule of the vendor.

Communication ties are also discussed at the initial meeting. For example, as previous speakers have noted, it may be possible for the Center to make available verification technology. One way that might be done is via the ARPAnet. We try to introduce the vendor to communication ties and media of the general computer security community.

After the initial meeting, there are several subsequent meetings. It has been our experience that there are perhaps three or four meetings in a year or a year and a half. Again, this is driven by the vendor's design or development schedules. What primarily happens at these meetings is technology transfer or exchange. We try to help the vendor team interpret and meet various criteria requirements. For example, at the implementation level, there is often a large number of questions concerning levels and categories, how categories are combined, etc. Another area is formal methodologies, a new technology. We try to work as much as possible from internal documentation. In the case of a new design, we may be working from notes written on napkins. We try to influence, as far as possible, the actual security design process.

Additionally, members of the evaluation team may attend vendor-sponsored operating system or internals courses. We invite the vendors to seminars such as this one. During this period, we try to look closely at the proposed or existing mechanisms to see how they support the requirements of the criteria-marking, mandatory security, discretionary security, accountability and continuous protection. We look for an implementation of the reference monitor concept, something that can be identified as a TCB. Where required by criteria (Division B and above), we help the vendor define the security perimeter of the TCB. We look at the kinds of security assurance provided by the vendor testing, validation and verification strategies. If a formal model is used, we try to make a judgement on the security provided by the model.

A preliminary evaluation can end because the vendor is finished with the design or enhancement, or the vendor can voluntarily drop out, deciding not to be evaluated. A preliminary evaluation report will be produced for the vendor. One of the main reasons for the report is to capture knowledge about the system and the work that has already been done, particularly if a formal evaluation is forthcoming at some later time. The preliminary report will probably contain proprietary information and will not be public. It will not be a rating, but essentially a consensus document between the evaluation team and the vendor that will give the vendor an unofficial class fitting and tell him, in terms of the criteria, what are the strengths and weaknesses of the system. The vendor can then decide how he wants to use the information.

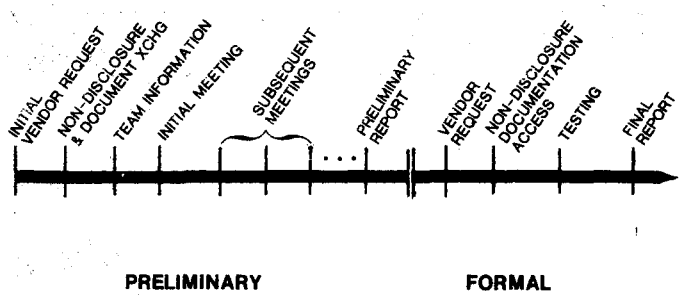
Continuing from a preliminary evaluation or at some later point in time, there will be a formal evaluation. Again, there is a vendor request for the formal evaluation. At this point, due to the commitment of resources necessary to do a formal evaluation, the Center will screen the requests and look first at those systems that will produce products most useful to the DoD. If a non-disclosure agreement has not been signed or if a significant amount of time has passed, there may be a new agreement signed.

One new requirement for formal evaluation is access, by the evaluation team, to the system (software and hardware as needed) being evaluated. This is necessary to perform penetration testing of the system and perhaps code review for higher level systems. At the initial formal evaluation meeting, the parameters of this access are defined. Additionally, a schedule for evaluation is laid out. We think six to nine months is a reasonable framework, although this may be subject to change.

The last phase of the formal evaluation is a final report. This is a public report that does not contain proprietary information. It is essentially a rating, but a rating with comments. The comments are necessary because different systems provide differing functionalities so that the security of a given system is dependent on the functionality that the system provides. For example, in a communication system, it may be sufficient to provide total isolation of processes, whereas in a time-sharing operating system, a more standard multi-level approach, including the star property, has to be considered. Thus, the comments will be part of the rating. The rating and final report will be inputs for EPL (evaluated products list) placement.

In conclusion, this briefing reflects how we currently see the evaluation process. Again, we are currently engaged in some preliminary evaluations and one formal evaluation, so some changes in the process may be forthcoming as we gain more experience. □

Evaluation Time Line



Preliminary Evaluation

- Non Disclosure Agreement
  - Proprietary Protection
  - One Document
- Document Exchange
  - Architecture/Internals Manuals
  - Evaluation Criteria, Methods, Models

Preliminary Evaluation

Team Formation

- Evaluation Team
  - Members from the Computer Security Center, MITRE, DOE, NRL, Academia, Private Research
  - Computer Security, Experimental, Theoretical and Formal Expertise

Preliminary Evaluation

Team Formation

- Vendor Team
  - Evaluation Contact
  - Marketing Management
  - Technical Expertise

Preliminary Evaluation

Initial Meeting

- Goals, Philosophies
- Technical Overview
- Schedules
- Communication Issues

Preliminary Evaluation

Subsequent Meetings

- Driven by Vendor
- Technology Exchange
  - Help with Requirements
  - Work From Internal Documentation
  - Courses, Seminars

Preliminary Evaluation

Preliminary Report

- Not Public
- Not a Rating
- Consenus Document
- Class Fitting

Formal Evaluation

- Formal Vendor Request; DoD Use
- Non-Disclosure
- Initial Meeting
  Access Requirement
  6-9 Month Schedule

Formal Evaluation

Testing Phase

- Closed Evaluation
- Penetration Testing, Code Review

Final Report

- Public, Non-Proprietary
- Rating with Comments, Best Usage
- Basis for EPL Placement

COMPUTER APPLICATION EVALUATION—SCIENCE OUT OF ART

Stephen F. Barnett
Chief, Applications Evaluation
DoD Computer Security Center

Steve received a B.S. in Mathematics from the State University of New York at Stony Brook, and an M.S. in Mathematics from George Washington University (D.C.). He joined NSA in 1963 as a Mathematician analyzing the security of U.S. communications security systems. Since 1972, he has held managerial positions in organizations working on the application and analysis of security measures to computer systems.

You've been hearing about the evaluation products, the evaluation criteria, and the evaluated products list. I'd like to talk about security evaluations of applications of computers to data processing systems. A system may consist of a general purpose computer and a collection of terminals and other input/output devices needed for users to do useful work on the system. A system may consist of many computers, interconnected by a local area net or common user data network such as the ARPAnet or the Digital Data Network being developed by DoD. The system may consist of a collection of special purpose computers to provide command and control of a weapons system such as Minuteman or MX. In each case the system is intended to provide a specific set of functional and computational services to a specified set of users, collocated with the computer or widely dispersed. These systems may be operational with security requirements "satisfied" by imposing considerable operational and environmental controls. These systems may, on the other hand, be in development and thus provide the opportunity to design security controls into the hardware and software or use evaluated trusted products to provide such controls. The goal is to allow the system to be used in a less restrictive mode. Applications evaluation has its feet in both worlds—the here and now systems and future systems. For the security evaluator, each system represents a unique situation, to be considered on its own merits, using a well-defined set of evaluation tools, techniques and criteria.

This morning I will discuss, from a evaluator's point of view, the approaches which have been, are being and will be used for assessing security of the *total* system. We are, I believe, in computer applications evaluations truly evolving from using the "arty" game of wits played by tiger teams, to applying the scientific approach developed with security kernel research and being applied to the product evaluations you're hearing about.

Current government policy states in effect that information which is stored and processed in computer systems will receive the same degree of protection as it does in a manual system. This protection is provided by suitably combining "classical security techniques," i.e., physical security, personnel security, and communications security, with security measures in the computer's hardware, software and operating procedures. As computers become more sophisticated, so do the operational requirements, particularly for increased sharing and remote access. Thus the computer had an ever-increasing responsibility to protect information against unauthorized access and unauthorized modification. In some cases it also has a role in preventing or at least detecting attempts at unauthorized disruption or denial of service.

Computer systems applications are vulnerable to such attacks because of poor design wih respect to security, poor use particularly of security measures, and failures or sabotage. For the last ten to fifteen years there has been a growing effort to reduce the vulnerability of computers by instituting managerial and technical solutions to these root problems. However, as long as people use, manufacture, maintain and operate computer systems there will always be residual vulnerabilities in any given system. The objective of security evaluation is to assess the effectiveness of in-place safeguards and the likelihood of successful

exploitation of the system. Just as the techniques for providing safeguards are maturing, so too are the techniques for evaluating the systems and interpreting the results.

The initial approach to improving computer hardware and software protection in applications of large scale computers was "find and fix." Tiger teams conducted extensive tests of applications and operating system software. When flaws were found, they were "fixed" and the testing continued. One was quickly led to the realization that this form of evaluation never finished because the systems were not designed to be evaluated. One could not prove that a system was secure. Not finding flaws only proved that the evaluator had not found or recognized any flaws where he was looking. With such an approach, the penetrator has the advantage. He only has to find *one* flaw; the evaluator *must* find *all* flaws.

For government applications an alternative approach was taken which might be characterized as using "built-on" controls to achieve security. These controls are intended to make each system compliant with the security requirements of applicable ADP security policy directives. Conceptually, these requirements are the same as those expressed in Chapter 2 of the trusted computer system evaluation criteria. This is not surprising since these requirements have a common origin in government policy for protecting and handling sensitive and classified information.

For ADP applications the requirements are described in the context of a "mode of operation." The decision to approve use of the computer in a particular mode of operation, that is accredit it, is based upon verifying that the security controls which are added to the system and its operating environment adequately meet the requirements for the mode of operation. Implicit in this is determining not only that the requirement is addressed, but also that the particular safeguard is effective. I'll say more about this shortly.

Each mode of operation requires environmental controls such as physically securing computer and terminal facilities, placing limitations on user clearance levels and/or on classification levels of information which can be processed, and securing communications. These all control access to the system and reduce exposure of information to outsiders. Some of the approved modes of operation also require some technical controls over access identity, providing reliable classification and handling labels on internal files and output, isolating users from each other, and auditing use of system resources. As General Faurer mentioned, the need for reliable classification labels is particularly critical because, as in the manual world, it is the basis for granting access to the data, for preventing contamination of files with more sensitive data, and for preventing accidental or deliberate downgrading of information. The dedicated and system high modes rely heavily on environmental controls for protection, whereas controlled and multi-level rely more on building trust into the system hardware and software. The system manager's decision on which mode is desirable, although often dictated by operational needs, must take into account factors such as the operating environment, user capabilities, perceived threat, sensitivity of information and degree of trust which one can incorporate and verify in the hardware and software. Since the effectiveness of software controls is potentially limited by inherent design flaws in the underlying operating system, there is always an element of risk when we build on controls in a system which was not designed and built to be evaluated. The evaluator's job is to measure this risk, and make an independent recommendation to the accreditor who must make the ultimate decision on whether or not the residual risk is acceptably low.

The evaluation process itself consists first in determining what the system's security requirements are, and the relative importance of each. In some cases unauthorized acquisition of information may be of primary concern to those responsible for security of the information. In other cases, unauthorized modification of information or denial of service may be considered a more severe threat. The next step is a critical examination of the system to identify whether or not applicable security requirements for the desired mode of operation are addressed and to assess how effective the safeguards are against the postulated threat. Effectiveness can be measured by answering questions such as: (a) does the safeguard correctly implement the policy requirement? (b) is the integrity of the safeguard protected? (c) is the integrity of data used by or produced by the safeguard? (d) is the safeguard invoked at all times or can it be selectively "turned off"? (e) how much time and expertise might it require to defeat the safeguard? (f) are the security markings accurate and safe against unauthorized modification?

Interpreting the evaluation results and making recommendations on how to improve the system's security posture is the next step. The objective is to reduce the residual risk. This risk is a function of the threat, system vulnerabilities, value of the system and its information, and time. Threat can be described in terms of

an opponent's capability, motivation, and opportunity to identify and exploit vulnerabilities or to create exploitable vulnerabilities. To reduce risk one must minimize opportunity, eliminate or reduce the exploitability of vulnerability, or use trusted systems to enforce security controls. The evaluation findings serve as the basis for identifying the most effective ways to reduce risk by imposing stricter environmental controls or eliminating technical vulnerabilities in the system. Thus, as a result of the evaluation the accreditor knows what the strengths and weaknesses of the system security mechanisms are, has an independent assessment of the risk he is taking by using the system, and has a set of actions which could be taken to reduce the risk.

At this point you may be wondering how does one intelligently "look" at a system which has not been designed to be evaluated? What kinds of weaknesses could be found and how are they discovered? Clearly the emphasis has got to be, for operational systems, on assessing the placement, use and effectiveness of measures to establish and enforce a security perimeter around the system. This creates the framework within which one examines the built-on technical controls such as audit trails, user authentication and isolation, and information labelling; and decides if they provide an adequate level of control with the security perimeter. In any cases the shortcomings are not hard to identify; their impact and the necessary actions are fairly obvious as the following cases illustrate.

In many modern systems, user passwords provide the principal means of verifying a user's identity and determining that individual's authorized privileges on the system. The effectiveness depends first of all on protecting the secrecy of the password. However, there are still terminals which cannot suppress the password when it is echoed back. User-generated passwords are often guessable—his or her initials, a last name reversed, a pet's name, etc. Sometimes a thoughtless user will store an active password on a file which may be easily accessible so that a user's password or access authorization can be read and modified.

Some systems do not support an audit trail capability so there is no basis for after-the-fact detection of abuse by even the most unsophisticated attacker. Other systems have audit trails which collect volumes of information, including some not relevent to security and not including data which is. Finally the user logs may not be adequately protected against user access, in which case the veracity of their contents must be suspect.

It is well known that most current computers are not effective in providing internal access control. Unfortunately, in a lot of cases it does not require a great deal of expertise to be able to accomplish the kinds of attacks listed here. This alone underscores the need to establish an operational environment in which one must place and maintain some level of trust in the users when one cannot build or use a trusted computer system.

But we have found that in some cases even the environmental controls leave something to be desired. Often the management of system configuration and maintenance provides opportunities for outsiders to negate controls or even directly access information in spite of these controls. Evaluations often identify blatant gaps in the security perimeter which is enforced by physical security and operational procedures—gaps which provide easy access to information.

The evaluation process here, in comparison with the Tiger Team approach, clearly has better defined goals and boundaries. There is a better sense of when it is finished, and what the results are. However, there are several pitfalls. First of all, the requirements are stated in qualitative terms with no statement of what constitutes compliance for a given application, much less how to measure the degree of compliance achieved or whether the degree of compliance is acceptable. Thus, while it is relatively easy to demonstrate the existence of measures intended to meet a requirement, assessing their effectiveness and measuring residual risk is subjective. This is particularly true of software safeguards on a system which was not designed to be evaluated.

Secondly, evaluation relies heavily on analysis of information about the system, which may or may not be accurate. It also relies on testing to verify that the implementation of a safeguard satisfies its requirement specification. Thus, for testing to be most effective there must be a requirement specification and a means of determining that the implementation meets it. Thirdly, this form of evaluation is most effective in detecting vulnerabilities which an inexperienced penetrator might seek and exploit. It gives less confidence about the

effectiveness of these measures against a determined, knowledgeable penetrator and about the security of the underlying system software and hardware.

With all these negatives, one may ask why bother? First of all it is better than believing that a system is secure because someone emphatically asserts that it is. Secondly, in many applications, the strongest threat may be from the inexperienced system user and casual tinkerer. This would be the case for example, if the system handles a small amount of sensitive information, or if it operates in a heavily protected environment. In that case, augmenting the environmental controls with *effective* user authentication, reliable file labels and an effective and unmodifiable audit trail may provide adequate security, although it is not a substitute for good design. Residual and unknown operating system vulnerabilities are presumed to be unlikely targets of exploitation. Therefore, the evaluation process described previously can provide sufficient evidence of the adequacy of these measures, or perhaps more importantly, identify areas which should be improved. It accomplishes several other important things. It points out where a system's strengths are. The manager then is better able to preserve these strengths during operational use of the system. The evaluation provides evidence to support acquisition of additional security measures, or to justify greater attention to security in follow-on acquisition. It also provides the basis for clearly understanding the security impact of proposed changes to system configuration or use. Finally, it can provide valuable insights which can be applied to other evaluations.

In short, the evaluation process described above is particularly suited to existing operational systems which use built-on controls. It provides a rational basis for deciding if the system, in its desired mode of operation, adequately protects information and for identifying those areas which are likely to contain the greatest degree of vulnerability. It must be used with care so that the system manager does not get a false sense of security by believing that once the identified weaknesses are fixed the system is secure. Without a trusted base, such built-on controls will not achieve their full potential; a well-controlled environment is still the most important issue for these applications.

We are conducting many such evaluations and are continuing to refine our techniques to make them as objective as possible. We are supporting our military counterparts by providing evaluation information, tools and assistance as requested for their operational system evaluations.

In securing new computer applications we are attacking the design issue and developing evaluation methods and techniques which avoid some of the pitfalls identified earlier. We are borrowing heavily from the design and verification techniques which arose out of the security kernel research. The basic theme is to address security policy requirements thoroughly and correctly in the design. Then any vulnerabilities found in the implemenation of that design will exist due to implementation or operation errors, not design errors. The former are far easier to correct, given a sound security design. Evaluation concentrates on proving that the design enforces all the required security controls and the implementation correctly follows the design.

This approach still requires, in fact depends on, using classical security measures to control external access to the system and to support the internal access control mechanisms. However, it provides a constructive means of eliminating, or at least reducing, poor design as a root cause of security vulnerabilities. Most importantly, it requires security evaluation to be an integral part of the entire design development process, not something that is done during or after system test and evaluation and prior to IOC.

Let me reiterate that I am talking about evaluating a "system" which may be as simple as a single computer with collocated terminals or as complex as a collection of local area nets serving many hosts and remote users and interconnected by a backbone digital network.

The evaluation process starts with the definition of the system architecture which identifies at a high level the major system components and their interconnections. It also identifies the security boundaries required around and within the system. It is at this point that one must identify where environmental controls, communications security systems and trusted computer systems, with user authentication, user isolation, marking, and auditing mechanisms, must be used to control the flow of informtion and user access across these boundaries. The evaluator has a big responsibility in this area because it lays the foundation upon which the system will be designed, built and evaluated.

The resulting security architecture takes into account the operational requirements and security requirements. It may offer several options on how to institute the kinds of controls required, and identify issues which the system designers and evaluators must resolve in order to design and evaluate the system.

The basis for the security design is a formal model of the policy which the control mechanisms in the architecture will enforce. One then proceeds as in the development of any trusted system to produce a top-level specification and prove that it corresponds to the model. This must be done at the system level and then for each trusted component. One can then proceed through successive refinements off the specification, each time proving correspondence with the previous level. In this way, one arrives at a design which correctly implements the applicable security policies. Since the design describes precisely how the implementation will behave under all combinations of inputs, one can indeed test the implementation to verify that it appears to follow the design. Testing is necessary because, for example, proving mathematically that the code corresponds to the design is beyond the practical state of the art. We are looking toward the R&D community to make software verification tools and techniques available. The extent of verification and testing is dictated by the operational environment and the amount of assurance one requires.

Because these methodologies are constructive and involve the system design, one can build a greater degree of trust into the system and provide a greater degree of protection against a determined, knowledgeable adversary. But success depends upon a close interaction between the system planners, designers, implementors, and evaluators. For example, the RFP should clearly articulate the security requirements, design and implementation criteria, and contractor verification and validation tasks. It should also describe the support which the contractor will provide to the evaluator, including documentation, timely versions of software, access to development systems, and V&V evaluations.

As with product evaluation the applications evaluator analyzes the system against the design and evaluation criteria which are applicable to the system. These criteria define security "correctness" for each safeguard and for the entire system. They define how each safeguard should be designed, implemented, and verified to meet the correctness criteria. The intent is to have well-defined conditions, which interpret the trusted system evaluation criteria for the planned system mode of operation, and which if adequately met, imply that the system provides an acceptable level of protection and can be used operationally. The existence of these criteria does not guarantee that the system will be approved. The system may, in the evaluator's estimation, fall short of its security goals because of the fidelity with which the developer follows the design and implementation process, not because the criteria are incomplete.

Several challenges must be faced in developing criteria for a specific application. First, the criteria must clearly reflect the underlying security policy and be relevant to the anticipated environment in which the system will operate. This can create problems for the developer and evaluator if the operational environment, function, customers, or configuration of the system changes during the course of its development. Secondly, the requirements set forth in the criteria must be achievable within the development time of the system, and yet must be stringent enough to provide, at a future time, adequate assurance of security in the threat environment for the system's operational life. Thirdly, the criteria should be so well defined that one can, in a repeatable fashion, demonstrate that the system meets them or fails to meet them. Finally, the criteria that are used for a particular application must describe design and verification on requirements which are compatible with those imposed on the trusted products, so that one can make appropriate selections from the evaluated products list and have a consistent basis for certification recommendation.

Security evaluation is an ongoing process. The evaluator's job is to assure that the contractor is following the criteria and guidance provided at the outset. The design must be thoroughly evaluated prior to implementation. This evaluation and the decision to proceed with implementation is based upon analysis of the policy model to be implemented, proof of the correspondence between Top-Level Specifications (TLS) and model, and proof of correspondence between the TLS and subsequent refinements of it. The design schedule must allocate time and resources to accomplish this.

The implementation, and the test and evaluation schedule, must also allow time to conduct necessary analysis and testing of software, particularly that which is security-critical. The evaluator is often expected to make a recommendation at or before IOC. Without proper planning, the software will not be available soon enough before IOC to allow time to finish the security evaluation and testing. Hopefully by paying more attention to the design, we can reduce the dependence, and hence the time, for code test and evaluation.

One final comment in this regard concerns protection of integrity. It is vital to provide some level of control over access to the design and implementation of security hardware and software as it evolves. The

level of control required is dictated by the eventual operational environment. Without such control, there can be no guarantee that the design or implementation provides the degree of protection required.

What, you may be wondering, has our experience been with using this process? At this point it is very limited. The methodology was used in the evaluation of AUTODIN II. We learned a lot from that experience. The Center has been tasked to provide support on the WWMCCS information system. The requirement is defined. We are developing security inputs for the RFP and will be actively supporting the Program Manager during the development and evaluation of the system. We are to provide a certification recommendation. We have also been tasked to provide support for the Inter-Service/Agency AMPE. Currently we are working with DCA on the system architecture and development of evaluation criteria. We expect to provide support during system development, evaluate the result, and provide a certification recommendation. Finally, these security design and implementation principles are being used in the SACDIN software and we have been tasked to provide evaluation support to SAC. I expect other requests for assistance. As General Faurer mentioned, he wants the Agency to be forward-looking in this regard. Therefore, I expect that requests for such assistance will come from in-house too. Thus we expect to gain additional experience with using this evaluation approach on a variety of applications over the next several years.

In conclusion, evaluations of government systems has come a long way. In many cases it must still emphasize assessing environmental and procedural controls. The path towards doing a better job with security design, implementation and evaluation is becoming better defined. It is clear that we can do a better job. There are a lot of hurdles to overcome; some are technical, some are philosophical. But we have a big incentive: The offense is ahead of the defense. □

RESEARCH AND DEVELOPMENT IN SUPPORT OF TRUSTED SYSTEMS EVALUATIONS

Hilda Faust Mathieu
Chief, Research and Development
DoD Computer Security Center

Mrs. Mathieu holds an M.S. in Chemistry from The American University, Washington, D.C. and a B.S. in Mathematics and Chemistry from Marywood College, Scranton, PA. She attended the Federal Executive Institute, Charlottesville, VA and is a graduate of the Federal Executive Development Program (FEDP III). She has been employed by the U.S. Government for over 25 years as a mathematician, systems analyst and technical manager. Mrs. Mathieu has been active in the field of computer security since the mid 60's; her contributions have been in the areas of analysis, research and policy. For the ten years prior to her current assignment she was Chief of Computer Security Research at NSA. She was a contributor to the Defense Science Board's Task Force on ADP Security in 1969 and a member of the Air Force Computer Security Technology Planning Study Panel in 1972. She participated in the formulation of the Department of Defense Directive on "Security Requirements for Automatic Data Processing Systems," the Industrial Security Manual Annex on ADP Security, and the Intelligence Community's Directive on Security Requirements for ADP. Mrs. Mathieu has made numerous presentations on various aspects of computer security to audiences both within the government and at the national level.

INTRODUCTION

In this paper I would like to present some of the research and development efforts that we have ongoing under the Department of Defense Computer Security Initiative. In particular, I will address those R&D goals which are intended to assist the Department of Defense in acquiring trusted computer products.

R&D is being carried out to provide the tools and techniques for performing security evaluations. We are also giving considerable attention to supporting the design and development of secure systems—in particular, systems designs and mechanisms for increasing our confidence that the computer products will provide an appropriate multi-level security environment. Therefore, the emphasis is on architectures that provide the access control mechanisms in which we can place our trust; in the design of these mechanisms in a manner so that they can be thoroughly evaluated; and in the development of the tools for carrying out that evaluation process.

R&D APPROACHES

Two specific approaches which we believe will help to achieve these goals are the consolidation of the generic computer security R&D being sponsored by the DoD, and the transition of the research results into practical application.

The term "generic" is used to define those research and development tasks which are expected to have general and broad application in providing computer security, for example: software verification tools; methods for automated security classification marking of data; access control measures; and secure operating system designs. Generic R&D does not include R&D which is required to meet specific needs of a particular military or defense computer-based development or acquisition. Such R&D will remain a part of the particular project it supports and will not be included in the consolidation.

In the application of the research results we are beginning a concerted effort to bring into practice the products of the computer security research which has been ongoing for the past decade. The intent is to move the techniques and methods from the "research" environment into prototype demonstrations and practical applications. I will give more detail on each of these approaches in turn.

CONSOLIDATED GENERIC R&D PROGRAM

First a few words about the consolidation of the generic computer security research and development within the DoD. I'd like to address this program's objectives, define what R&D efforts will be included, and identify the benefits that are expected to result from this approach.

The resources that are available to be applied to solving the computer security needs are scarce indeed. This is true from a dollar standpoint and, even more importantly, from a people standpoint, both in government and in industry. One major objective in supporting consolidation of the generic computer security R&D is to coordinate the use of the resources which are available. We expect that this consolidation approach will help identify both duplicity and gaps in the current program.

In the past there have been cases where agencies have funded research and development efforts without knowing that these same results were being pursued by another organization or were already available. For example, recent review of the technical submissions to the consolidated R&D program revealed that DCA, Army, Navy and the Center all proposed similar tasks on the verification of software written in the ADA programming language. This is not to imply there should be no parallel studies or alternative investigations. Those should continue when considered beneficially supportive of the task goals. We also expect the DoD components to maintain the lead role in certain R&D areas; for example, Navy's responsibility in military message system R&D, and Army's role in ADA verification R&D. The intent is not to have all the R&D sponsored by the Center, but rather to increase these capabilities throughout DoD.

In addition to finding too much effort or duplication in some areas, we also realized there were other important areas that were not being addressed at all. The independent efforts being pursued by the various DoD components, in the past, were directed to small parts of the computer security problem which seemed most relevant to their needs.

The DoD Computer Security Center with the other DoD components will now undertake long range planning to provide the framework for the consolidated generic computer security R&D program which officially starts in FY 1984. The aim will be to assure best use of available resources and a program that will address the spectrum of the computer security problem as completely as possible. The plan will include R&D in five major technical areas: secure operating systems, secure data base management, computer network security, software and hardware security analysis techniques, and formal software verification techniques.

During the past couple of months, we with the other DoD components have gone through our first exercise of consolidating the generic computer security R&D tasks which have been perceived as necessary to undertake. For purposes of organizing the R&D effort, we identified three research categories and three development categories.

We further subdivided the research efforts into first—those addressing the informal and formal definitions of security, the modelling of the security properties, the standards, and the evaluation criteria. Secondly, under design concepts, we included the architectural issues, the fundamental security design, the investigation of access control mechanisms, and the role of encryption in providing protection in a computer system, data base system or network. A third area of research includes the fundamental logic, mathematics and techniques which will lead to development of an automated capability to support the evaluation and analysis of both computer software and hardware.

The development efforts have been further subdivided into three areas. First, the exploratory and advanced development of secure computer systems; e.g., secure relational data base management systems, secure operating systems and application subsystems. A second development area addresses the secure network issues such as the interfaces required to accommodate end-to-end encryption within computer networks and the issues associated with the protocols which determine how information would pass securely throughout networks and among networks. And, thirdly, the development of the techniques for the evaluation and assessment of the software and hardware, such as the formal software verification systems and tools and the automation of analytic aids. The consolidated generic computer security R&D program proposed for FY 1984 contains over sixty tasks.

How do we expect such an effort at consolidating the generic R&D to help in achieving evaluated trusted computer products? We certainly hope that through the concentration and coordination of the investment by the DoD and by the research and development communities, we can accelerate arriving at solutions for some of our computer security problems; and that those solutions will, in fact, have broader applicability, and therefore result in more widespread benefits for the DoD. Both of these rationales mean that we intend for the consolidation and coordination among the DoD components which will be necessary to carry out this R&D program to result in a better return for the government's and industry's investments.

APPLICATION OF RESEARCH RESULTS

A second focus of the R&D effort will be the application of the research results which have been accruing over the past decade, primarily in the areas of software verification and in the application and implementation of secure architectures, both for computer systems and networks.

VERIFICATION TECHNOLOGY

Software verification is the approach to the more formal demonstration, based on mathematical proof of the correctness, consistency and completeness of the system specification and implementation. I will address the methods, tools and techniques for carrying out this formal verification process. (We are also developing the automated and interactive aids to support the less formal analysis of software programs written in Fortran, PL/1 and other higher order languages.)

In the area of software verification, we believe that it is time to make some of these research tools available to the potential user community; this will give them an opportunity to learn about the intent, the techniques, and the benefits of formal software verification, and the role which it can play in the development and evaluation of trusted computer systems. These tools that are available and the systems that are available are still very experimental. To date there is very limited experience outside the small group of computer scientists who developed these tools and techniques. The number of users of the formal methods and tools is starting to grow. We hope over the next several years many more will become part of that community of users. This is not to say that within the next two years these tools will be production quality. Rather, in the next year to 18 months, the Center plans to support several experimental verification systems (GYPSY, HDM and FDM) on the ARPAnet, for the software developers in industry who are interested in making the investment in manpower resources to begin experimenting and participating in this developing techology.

In order to achieve this in the near term, we will strive to stabilize these tools and systems. By this we mean to make the current tools more complete, and usable; they will not necessarily provide optimum capability. Research in this area will continue; therefore, those tools that will be offered for experimental use in the next 2–3 years will not be the final product. However, they will be a reasonable indication of what will be available in the future. So, while we are continuing to sponsor additional research, we intend at the same time to stabilize and enhance the GYPSY system from U of Texas, to stabilize the HDM/Special system from SRI, and we expect to also be able to offer the FDM system developed by System Development Corporation, and whatever others are mature enough to make available for people to learn about formal verification. Today none of these systems includes much tutorial information about how to use them. We will try to improve their "user-ability" and stimulate their developers to produce better user interfaces and documentation. As already mentioned, we expect to make these tools available on the ARPAnet, and to help people to become knowledgeable in the technology with a lower initial investment on their part. They will not have to obtain a large computer system which is needed to support these verification tools. We hope to provide support to those tools we will make available on the network. But recognize that these tools are still experimental and that the people who provide this support will be, at the same time, investing their efforts in carrying out the enhancements and research for the next generation capabilities.

Long-term verification goals then are to look at the next generation environments which should be developed in order to meet the verification needs starting in three to five years, and to extend the current capability of the near-term tools. For example, there will be a serious effort to address the verifiability of the ADA programming language. This will be done in close coordination with those primarily responsible for the ADA activities and those developing other ADA environments so that the software verification can be

an integral part of using the ADA language. Current verification tools and techniques will be investigated for their extension and potential application to communications protocols and microcode. Here, user feedback, experimentation, and exercise of these tools are going to be critical to identifying what additional capabilities are needed or desired and the current deficiences. Other long-range goals will be to investigate the relationship of formal software verification to hardware failure analysis, to secure design, and to testing; and to pursue alternative approaches for attaining a high degree of confidence in trusted system developments.

DESIGN GOALS

I'd also like to present our near- and longer-range goals in the area of secure architectures and designs. There has already been a reasonable investment in the security kernel technology—e.g., the KSOS projects and the Kernelized Virtual Machine (KVM) development. We have also taken the approach to invoke some of the hardware in the implementation of the security kernel in the Honeywell SCOMP project. We now need to enter a phase in which we experiment with this fundamental architecture and learn more about what is good, about what isn't good and why, and to identify viable alternatives. One of these alternative architectures has already received a certain amount of attention—that is, the concept of capability-based addressing which was pursued under the Provably Secure Operating Systems (PSOS) effort. The requirements for "Guard" systems and programs such as the Military Computer Family (MCF) offer opportunities for further investigation of the potentials of secure architectures.

We encourage and are interested in any experimentation which industry wishes to undertake in these areas and with these technologies. In fact, we would like to significantly increase the interactions between the Center and industry R&D. We had limited interactions in the past. We are committed to expanding this dialogue. We will be doing this in conjunction with the Center's efforts in the evaluation of commercial products, particularly in those instances where the product is in its early design stages—that is, during the preliminary commerical product evaluation.

In the near term also, we would like to reassess trusted computer base (TCB) architectures, primarily those based on the security kernel. We wish to examine their performance characteristics and investigate how their performance might be improved. Several attempts to implement security kernels in the past have resulted in systems with poor performance characteristics. However, to say security kernels are poor performers is probably not a fair assessment of that architectural concept. One major problem with past kernel designs has been that the security attributes have been associated with the wrong level of abstraction from the user's point of view. For example, the user is concerned about a classified message, whereas the system deals with classified files. One of the real possibilities of object-oriented systems (and capabilities are one way to implement an object-oriented system) is that the security attributes can be associated with the appropriate level of abstraction. Therefore, we need to look further at why performance was poor in past implementations of secure systems, and how these concepts can be better utilized. Both performance in the actual building of a trusted system as well as in the operation of the product should be examined.

In the longer range, in the area of secure architectures, there are some very interesting issues. One would be to look at the potential of standard modules that might be implemented in hardware, that would be designed with particular care, thoroughly evaluated, and then used as a building block to create larger systems: controllers, operating systems, data managers. There recently was an article about a systems developer who purports to do just that, although there does not appear to be a focusing on security. We need to examine how to incorporate security into such an architectural approach. Certainly such a concept of standardization and portability is in line with our desire to produce a good design, which can be thoroughly evaluated, and to be able to use that product many times without having to reinvest in the design and evaluation effort.

One of the reasons for the poor performance encountered in KSOS was that it was strictly a software implementation. Software, when compared to hardware, is slow. We need to look more closely at hardware and the implementation in hardware of the access control mechanisms. This was done in the effort that started on the SCOMP project. In addition, we need to look at what should be implemented in the hardware and what the hardware features are that can be called upon to provide the access controls we need. We also need to be able to specify and verify the trusted hardware properties. The trusted computer systems are a combination of software and hardware, and we need to greatly increase our capability to evaluate the

security mechanisms when these are implemented in the hardware. And we need to look at alternatives to the approaches we have taken to date. The VLSI technology certainly opens a realm of interesting possibilities for the meshing of the verification technology and the design and implementation of secure architectures. After all, VLSI design and implementation is going to be driven by software programs, and those programs should readily lend themselves to the kind of formal verification that we have been advocating as a means for conducting the evaluations that establish our trust in the computer product. The hardware approaches to providing checks and adding mechanisms for checking on the quality of performance are additional challenges.

FUTURE CHALLENGES

In summary, the challenges that we are facing are primarily two: First, how can we capitalize on our R&D investments—not just the government, but industry also? Both the government and industry have a limited number of knowledgeable people and "pockets" of interest and expertise in computer security R&D. What is the best way we can capitalize on these resources, and on our investments, and have a cooperative effort without impacting the free enterprise system and the competition that must exist in the commercial world?

The second major challenge is how do we most effectively move the research results into practical application? Our efforts and our plans in the verification area in the near term to make those tools available have been mentioned above. We need, however, to have these tools exercised. The future users, the system/software developers, need to experiment with them, to provide feedback, and to suggest improvements and alternatives. We in the DoD Computer Security Center don't pretend to possess all the knowledge—we would like to be able to guide and lead the technology, but any advancements in the technology will also require the good ideas from the researchers and the vast experience in the practical side of the problem solution from industry.

The DoD Computer Security Center looks forward to working with industry and academia on these challenges in computer security research and development. □

R&D GOALS

SUPPORT FOR

- EVALUATED
- TRUSTED COMPUTER PRODUCTS

R&D APPROACHES

- CONSOLIDATE GENERIC R&D
- APPLY RESEARCH RESULTS

CONSOLIDATED GENERIC R&D PROGRAM

- OBJECTIVES
- DEFINITION
- RESULTS

CONSOLIDATED R&D PROGRAM

OBJECTIVES

- MAXIMIZE SCARCE RESOURCES
- COMPLETENESS

R&D TASK GROUPS

RESEARCH

- SECURITY DEFINITION
- DESIGN CONCEPTS
- ANALYTIC TECHNOLOGY

DEVELOPMENT

- SYSTEMS
- NETWORKS
- EVALUATION TECHNIQUES

CONSOLIDATED R&D PROGRAM

OBJECTIVES

- MAXIMIZE SCARCE RESOURCES
- COMPLETENESS

RESULTS

- ACCELERATE SOLUTIONS
- BROADER APPLICABILITY
- BETTER RETURN ON INVESTMENT

APPLICATION OF RESEARCH RESULTS

- VERIFICATION TECHNOLOGY
- SECURE ARCHITECTURES

VERIFICATION GOALS

NEAR TERM

- STABILITY
- USER-ABILITY
- AVAILABILITY
- SUPPORT

LONG RANGE

- NEXT GENERATION ENVIRONMENT
- EXTENDED CAPABILITY

DESIGN GOALS

NEAR TERM

- SUPPORT IR&D
- EXPERIMENT
- EXTEND

LONG RANGE

- ALTERNATIVES
- PORTABILITY
- H/W SUPPORT

CHALLENGES

- CAPITALIZE ON R&D INVESTMENTS
  - GOVERNMENT
  - INDUSTRY
- ENGINEER THE RESEARCH RESULTS
  - EXPERIMENT
  - FEEDBACK

SOFTWARE TOOLS

James T. Tippett
Chief, Technical Support
DoD Computer Security Center

Jim received B.S. and M.S. degrees in Electronic Engineering from North Carolina State University. He has done over four years of computer science graduate work and teaches computer science graduate school courses evenings at a local university. He joined NSA in 1955, where he had assignments in computer science R&D, on a career development panel, on the Science and Technology Staff for the Director of NSA, and in communications security evaluations and applications. Jim joined the Computer Security Center in 1981.

First, let me explain to you how we fit into the rest of the Computer Security Center. I'm Chief of the Office of Technical Support. We provide technical support for the Director of the Center. We provide support to R/D, as has been mentioned, with emphasis on the technology transfer of the R&D to the experimental and the prototype environments. With respect to the product evaluation office, we support them with software tools in their product evaluation or whatever they need to evaluate products against the computer security evaluation criteria. With respect to the systems evaluation office, our support includes the computers and software tools they need to perform systems evaluation. We want them to tell us what is wrong with our tools and how they can be better used.

Going back to this morning, General Faurer mentioned that the DoD Computer Security Center would make unique software tools available to vendors, along with data processing system support. This represents quite a large investment and requires highly specialized people to provide these tools. On the other hand, these tools are not limited in application to any particular product or application. Therefore, we are taking the initiative to provide the general tools that the vendors may not provide for themselves. We intend to make these tools accessible to government, academia, and industry participants, with each tool being hosted on a computer that can be used from a remote location to our government laboratory. We recognize that considerable work has been performed on various software tools to support formal verification. Additionally, there has been considerable work performed on software tools to support software testing and analysis. In other words, once you have designed a secure computer, you need to know what you really implemented. You need to test and analyze to find out how the black box or the software works against the actual written specifications.

We are interested in both testing and proving tools. We would like to see more of this software tool technology transferred to support evaluation on the computer security-related software against our computer security evaluation criteria. This work must be carefully performed by our organization to ensure that we have the most effective available tools and methods chosen for our evaluation. And looking at the field, I recognize that there has never been, that I am aware of, a comprehensive, systematic comparison of these sometimes-competing individual testing and proving methods and tools. We have a challenge ahead of us. We strive for better understanding of these testing methods and tools used individually and in combination. While we're working on our understanding of this area, we will emphasize technology transfer and the software tools as we foster further R&D on these tools. Our objective is to have the best available tools to support you as we go forth with our criteria and standards. We will seek further education on what other people are learning from their experience with software tools. Further education and training will be obtained from the evolving software testing and proving methods. As the research and development is completed, we want to be the first ones to hear any potentially useful results that you may obtain when using your tools against our criteria.

Our Center will be a showplace for software tools and increased use of automation. We will test and demonstrate the latest technology available, regardless of where it comes from. With this type of

background, just let me mention a few of the things that have gone on in private industry as well as in the computer societies within the last few years.

The IEEE computer society has sponsored a workshop on the effectiveness of testing and proving methods. The Air Force has sponsored a guidebook on software testing and evaluation. Boeing Computer Services has written a general guideline for computer software validation, sponsored by NBS. IEEE has sponsored tutorials on software testing and validation techniques, and has now developed a draft of a test documentation plan. Private industry has written, for the Air Force, a software tool installation guide to be used in installing tools and using them in the most effective manner. And in 1981, NBS, IEEE, and ACM sponsored a software tools fair in San Diego, and I understand another one is being planned in the Washington, D.C. area for July 1983. While this has been going on, there have been multiple software tools directories or indexes published in the last few years which have given increased visibility to these tools.

In addition to making education and training available in software tools, my organization will support computer security conferences and seminars. We will make sure the education and training are available on computer security-related matters. We expect that our Center will not be the only source of education and training, but expect contributions from industry and other places in the government. Our computer security information center will be the clearing house for the evaluated products list, as the Director mentioned this morning. We will make other computer security-related information available to you as it becomes available to us and ready for use. We will publish a computer security newsletter for broad distribution. We have recently published a preliminary product announcement on the Honeywell SCOMP computer. So, as you can see, the Computer Security Center is clearly different from the other parts of NSA. We are dependent upon cooperation from industry, and seek your continuing cooperative inputs.

We really need to do a very good technology transfer, not only in the computer security area, but also in the software tools area. Let us work together. □

PANEL SESSION — INDUSTRY REACTION TO THE TRUSTED COMPUTER SYSTEM EVALUATION CRITERIA

Moderator — Daniel J. Edwards
Chief, Standards and Products
DoD Computer Security Center

Panel Members:

Steven Lipner — Digital Equipment Corporation
Terry Cureton — Control Data Corporation
Theodore M. P. Lee — UNIVAC
Lester Fraim — Honeywell Federal Systems Division
Leslie DeLashmutt — Data General

INTRODUCTION

The first draft of the Trusted Computer System Evaluation Criteria (dated May 24, 1982) was released to the general public at this conference. A copy of the criteria was included with the Conference handout material given to each conference registrant. Copies of the criteria were furnished two weeks in advance to the Panel members and they were asked to participate in a discussion on the computer industry's reaction to the criteria. The panelists were invited to speak as interested individuals working in the computer security field and not as official representatives of their employers. Since the panel discussion centered around the first draft of the evaluation criteria, which will be issued in final form in early 1983, it was deemed inappropriate to include a full, verbatim transcript of the session as part of the Conference proceedings. Included in this section is an edited version of the opening statement made by each panel member and a summary of the key points made during the panel session, prepared by the moderator.

OPENING STATEMENTS

Lester Fraim — Honeywell Federal System Division

Honeywell's name has been mentioned several times as a vendor of products that fall into several of the categories listed in the evaluation criteria. Speaking as Product Manager of the SCOMP Development Program, we have products which we will be describing and offering for sale in the near future. The evaluation criteria came as no real surprise since we have been dealing with the DoDCSC and Mitre for some time. Some parts of the criteria are open to judgement on the part of the evaluator, but that also permits the vendor some flexibility in the features and assurances built into the vendor's products. There will have to be some refinements and adjustments but the existing document is a good base to start from.

Theodore M. P. Lee — UNIVAC

As editor for the committee that wrote the first draft of the 1978 paper which has evolved into the criteria distributed today, I believe that the criteria have definitely improved with each iteration over the past two-plus years. There are still many small things wrong with the current document, but these will be dealt with during the process of commenting on the criteria during the next six months. The biggest thing missing from the criteria is the mapping between the evaluation classes given in the criteria and something which relates to the users' application of the system. It is also important that vendors understand the principles which are behind the various evaluation classes and divisions. It is hard to judge the importance of any particular category without knowing what threats a particular class is intended to defend against and which ones it is not. It is hard to tell at this point what impact the criteria will have on the computer industry. The real impact will be determined by the market reaction: how many people need systems at which evaluation levels.

Steven Lipner — Digital Equipment Corporation

I have been following successive iterations of the criteria for the past eight months and I think this is definitely the best. Concerns remain with criteria requirements that appear to be placed too low or too high in the structure. The major portion of the criteria seems about right and structure appears to be relatively modifiable. The residual concerns relate to relatively small portions of the document but these concerns must be addressed before we get a document we can live with.

Terry Cureton — Control Data Corporation

My primary function has been as an interface between Control Data and the DoDCSC for the preliminary evaluation that is under way. I am quite happy with the document as it stands, although some details need clarification. I think the document is headed in the right direction and when it is completed it will have a rare blend of conceptual rationales, functional specifications, and concrete descriptions. Achieving that will be a tall order, but the framework is there in the document. I was particulary pleased with the explanatory material in the beginning of the document on how the evaluation process is intended to work. I found with very little technical jargon, management can understand that part of the document.

Leslie DeLashmutt — Data General

My job is taking operating system security from concept down through implementation. Data General supports the concept and implementation of the Center is we see it now. Likewise, we support the idea of evaluation criteria, and with few exceptions the current evaluation criteria document as we understand it. We intend to cooperate with the Center to the greatest extent our resources and the Center's resources allow. Overall, the criteria look very usable. One major exception to this is the one-dimensional nature of the criteria. We feel that the criteria are really representing two dimensions — quality of software and security model. For example, one could have a discretionary system proved correct to the A2 level and still receive a very low rating on the evaluation scale. Other parts of the criteria need more quantification, such as, what is an acceptable bandwidth for a storage channel? The biggest concern we have is the need for tools even to do some of the Division B Mandatory Security work. We find it difficult to spot problems without using tools. The knowledge on how to address problems such as Mandatory Security rests with a small group of people. Making those tools and people available to the vendors is really needed. It is not Data General's intent to advance the state-of-the-art in computer science. Instead, we plan to do the best we can using security technology that is well understood, well documented, and when appropriate, supported by production-quality tools.

SIGNIFICANT POINTS MADE BY ONE OR MORE SPEAKERS DURING DISCUSSION

- The criteria introductory material needs to be expanded to communicate with people who have no DoD background or experience. DoD forms a small part of the overall marketplace. Other non-DoD users need to understand what is required and how it is useful in solving their problems.
- The many individual criteria requirements need to be mapped back to the five basic requirements given in the criteria introductory material.
- The criteria need to be more specific in several areas including audit requirements and acceptable storage channel bandwidths.
- The criteria should give credit for security features which are implemented in hardware.
- Implementing security features in hardware should be viewed as an assurance factor rather than meeting functional requirements.
- The criteria appear to be oriented towards a security kernel implementation of Mandatory Security requirements.
- The criteria are relatively requirements-oriented and relatively neutral about architecture.

- It would be a major task to take a mature operating system and reimplement it to meet the requirements of class B3.
- The DoDCSC has made a commitment to make available formal verification tools. Besides that, the Center will have to make support people available to help vendors understand and use the tools.
- Even with the best Center support, vendors may be reluctant to be dependent on a set of formal verification tools and services outside their control. □

INFORMATION PROTECTION IN AN INFORMATION-INTENSIVE SOCIETY

Melville H. Klein
Director
DoD Computer Security Center

Mr. Klein received a BSEE from the University of Pittsburg, pursued graduate studies at the University of Maryland, and received an MSEE from Purdue University. He began his career with the Army Security Agency where he worked as an Electronic Engineer and an Electronic Scientist. Mr. Klein joined NSA in 1953 as an Electronic Scientist, and has been Chief of a Transmission Engineering Branch, an Analysis and Integration Division, a Technical Planning Group, a Special Projects Division, and of Security Communications Development. In 1980, he became Assistant Deputy Director for R&D, and in 1982 became Director of the Center.

During the past decade, the growth of the "information industry" has been revolutionary to say the least. The rate at which data is being generated and converted to "electronic form" quadruples every two years. Remote access to distributed data bases and processing resources has boosted terminal sales correspondingly and has launched a secondary revolution in data networks. Industry's response to this burgeoning demand for automated information services on a worldwide basis has for the most part sidestepped the issue of information protection. How vulnerable are these information resources and the networks to abuse and disruption? Last Friday's *Washington Post* reported on the prevalence of computer crime and abuse uncovered by the GAO in a recent study of government ADP systems. Despite such "exposes," an ADP industry hard pressed to meet current demands has found little time or much enthusiasm for providing "secure" or "securable" products. By the same token, the government in general and the national security establishment in particular, have not been in a good position to articulate their needs. But our nation's defense posture must be able to insure that the classified information entrusted to its ADP systems is safe from abuse, and that the networks conveying this information are impervious to electronic sabotage.

Concern for the security and integrity of ADP resources beyond physical, procedural and communications security measures which are an integral part of C^3 and intelligence networks is not new. In the early 70's, Air Force recognized the need and took the initiative to pioneer research on "trusted" systems. Pockets of expertise have since sprung up throughout the DoD. The technical expertise in the DoD necessry to cope with the emerging threat and growth of ADP did not materialize, however. The task of marshalling the resources within the DoD and stimulating the ADP industry to trusted products has not been an easy one. To divert the momentum of a highly competitive industry whose market is doubling every five years to a "trusted" product line is a formidable task. Industry's plight becomes particularly acute in view of the investment operating system software represents vis-a-vis the envisioned market share for trusted products. This, coupled with the pressing operational availability schedules DoD acquisition managers continually face in introducing ADP systems into defense inventory, has not augered well for trusted systems.

The DoD Computer Security Center was created to help bridge this gap by providing a focus for the development and application of technical computer security measures for defense and intelligence needs. Heretofore a voice in the wilderness, I see the climate for trusted systems changing. Today, though the national security market share remains small, its needs are vital and urgent. The prospects for much larger markets for trusted computer systems are beginning to surface throughout the civil agencies in the Federal Government and the private sector. Compliance with the Privacy Act of 1974, the Paperwork Reduction Act of 1980, and the initiative to combat waste and fraud are major areas of neglect pointed out in the GAO report, mentioned earlier, which have the executive branch playing catch-up. Similarly, the latent ADP "Three Mile Islands" are as disturbing to those in the financial sector as they are to others in the private sector concerned about computer "terrorism" and the threats to personal privacy. A recent Frost &

Sullivan report states that the average bank robber grosses $5K, whereas the average loss in the financial sector through computer crime is $300K. This disparity in rewards/risks balance places computer crime, if unchecked, in an ominous position. I see all these factors having a significant influence on the market share for trusted systems in the $200 billion ADP market of the 1990's.

As the slide shows, the trade press predicts ADP sales to the government will grow to $15.6 billion, with the Department of Defense's share expected to reach $2.6 billion, exclusive of imbedded and personal computers.

The Center's strategy in R&D, in the *development of standards,* and in the *provision of evaluation criteria and tools,* is to provide a technological *stimulus* to complement these emerging needs by working closely with industry. The Center is also striving to be a state-of-the-art practioner of trusted computing. We will use this knowledge to assist system acquisition managers to unequivocally identify and specify their computer security needs, and to advise designated approval authorities on their accreditation decisions.

Evaluation criteria are *basic* to the validation of the hardware and the software in trusted products—a critical link in the protective chain that forms information security. The criteria previewed at this Conference have proven to be as hard to define as they have been to apply. I know of no more apt theme for this Conference or a more challenging problem for this group of experts to tackle.

I'd like to reiterate *how* the Center plans to work with you throughout the evaluation process by first reviewing some recent history.

At last August's seminar, when Admiral Inman announced the establishment of the Center, he made some cogent observations on the relational nature of the Center to the private sector. In particular, Admiral Inman observed:

> "Because the private sector computer manufacturing community is the primary source of ADP systems, the Center's role will be to work with the manufacturers, deriving as much system integrity as possible from industry-developed systems. This is a rather sharp contrast to the NSA's more traditional communications security role where the government has the dominant technical role."

General Faurer reenforced this need in his address to the IEEE the following month. He noted the "enormous reliance" we place on industry to provide trusted ADP products.

In putting this advice to practice, we must supply two key ingredients for a successful relationship. The first has to do with the accessibility of the industrial sector and the computer security R&D community to the Center, and the second with the discretion the Center exercises in our dealings with clients and vendors. This relationship, be it with other government agencies, the data processing industry, or academe, is predicated on a modus operandi of accessibility, discretion, and candor. The accessibility of Center personnel has been characterized by a desire to be cooperative and forthcoming. Our exposure has been ubiquitous by NSA standards, but as Admiral Inman and General Faurer observed, this is essential if we are to successfully define and achieve mutually compatible objectives on trusted systems.

Two results I have observed thus far have been the Center's growing acceptance by the ADP vendor community, and the joint recognition that the computer security needs of the DoD are not incompatible with those outside of the defense sector.

The evaluation criteria presented here are representative of this interaction. They were developed in a climate where the technical exchange has been free and frank. Government, industry and academe will have continued access to the process that fine tunes the criteria. I encourage those of you from industry to start to apply these criteria to products you believe can benefit the defense and intelligence establishment as we proceed to the final criteria. The criteria strike the best collective balance between our understanding of operational needs and the state of the art. We are aware of their limitations and are prepared to adjust them to accommodate technological breakthroughs and changing needs. One test of the criteria's underlying validity will be how well the basic tenets stand up to technological change. Similarly, a test of our interrelationship is how well it can manage changing needs.

The principles underlying the draft criteria now being used in the evaluation of Honeywell's Secure Communications Processor (SCOMP) were also used by Honeywell in its design. They are also being applied

to other evaluation efforts in process. These include preliminary evaluations of products now in development from a number of manufacturers.

I am encouraged by the industry willingness to have us work with them in the *early design* and development phases of new security features. Later in the Conference, we will be hearing from several manufacturers on how they view this relationship.

As Steve Barnett has noted, the criteria have also proved to be helpful in evaluations where computer security was incorporated as an afterthought. Many of these cases were complicated by the fact that our evaluation teams had to work closely with the developers to determine how security was meant to be enforced *before* they could assess the adequacy of the systems' protection mechanisms.

These examples demonstrate the need for and benefits of two-way communications.

Proper and timely dissemination of information and technology developed in this process is also an important facet of the Center's mission. Though not all of the mechanisms for providing this are in place yet, it is clear from the demands for technical data on multi-level security, access control and audit mechanisms that we must improve our ability to respond. This type of information, as well as up-to-date product information, will be accessible through the Center's Information Services Division and at seminars like this one.

If I speak of "widespread accessibility of dissemination," I am referring to access to generic computer security technology information, particularly the results of the generic R&D program. On the question of specific disclosures, I want to stress that it is the Center's policy to appropriately protect proprietary data of vendors as well as sensitive information of clients. This obligation includes our dealing with system vulnerabilities as well as product evaluations. As we go forward with evaluations, any vulnerabilities identified are discreetly communicated to the requesting authority and vendor.

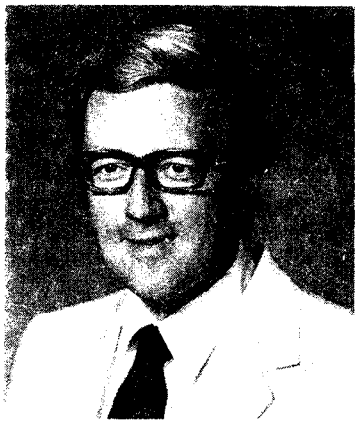
The formal promulgation and application of the evaluation criteria and test methods will help to speed up acquisition and increase the credibility of the certification process by reducing the need for waivers during development, and by providing unambiguous data to accreditation managers.

I look forward to nurturing the relationships established over the past year. They have provided a sound basis for addressing not only a technically challenging endeavor, but one that is vital to our nation's defense as well as our individual freedom. □

PROJECTED GOVERNMENT PURCHASES
(1990)

TOTAL ADP	$200 BILLION
FEDERAL GOVERNMENT	$15.6 BILLION
DEPARTMENT OF DEFENSE	$2.6 BILLION

DOD PERSPECTIVE ON COMPUTER SECURITY



Stephen T. Walker
Director, Information Systems
ODUSD (C^3I)

Steve has a B.S.E.E. from Northeastern University and an M.S.E.E. from the University of Maryland. He was employed at the National Security Agency from 1966 to 1974. From 1974 to 1978 Steve was a program manager at the Defense Advanced Research Project Agency (DARPA) where he was responsible for managing research efforts in computer security, computer networking, automated message technology, and intelligent terminal systems. He is presently responsible for major C^3I information systems such as the WWMCCS Information System, and the Defense Communications System including communications networks such as the Defense Data Network, AUTODIN, AUTOVON, and the ARPA network. He established the DoD Computer Security Initiative in 1978 and fostered the establishment of the DoD Computer Security Evaluation Center at NSA. Steve is Chairman of the Subcommittee on Network System Security of the National Communications Security Committee and the Technical Liaison Group on Trusted Computer Systems of the Technical Cooperation Program (U.S., U.K., Canada, Australia, New Zealand).

It is indeed a pleasure to be here for this Fifth Seminar. I am very pleased to observe the progress we have made since the first of these seminars in July 1979.

A number of people have asked me, "What are you going to do now that the Computer Security Evaluation Center is established?" The answer to this question will become obvious during my presentation. I am now involved in a number of projects, all of which have significant computer security implications. I would like to tell you about several of them.

First, I would like to review just a brief history of the DoD Computer Security Initiative. We have heard a lot about it in the last couple of days, but there are several things I would like to stress.

The slides that I am going to use now are ones that many of you have seen many times before. I put most of these slides together back in 1978 when we first started the initiative. They are beginning to get a little ragged around the edges but are still remarkably accurate. It is exciting for me to see how we have progressed from what we thought might be possible back in the 1978 time frame.

The thrust of the DoD Computer Security Initiative is to achieve widespread availability of trusted computer systems. That was the objective we set four years ago and toward which we have come a long way. Many of the things that Mr. Klein just said are indications that we are quite a ways down that path. The next slide gives the definition of widespread and trusted. We started out talking about the widespread availability of secure computer systems and my friends at the General Accounting Office would ask, "What do you mean, you don't have secure computers now, you are processing classified information on computers that are not secure?" I realized very early that we had to clear up that definition. Our computers are secure physically and administratively; it is just that we can not trust the hardware and the software to separate access from one user to another. So we use the phrase, "trusted computer system." Widespread availability means that we have to get the manufacturers involved. We can not continue to build these systems as special-purpose projects that we build ourselves. The panel session yesterday was an interesting indication that the computer manufacturers are involved and do care.

The third slide is the classic Willis Ware diagram from the 1970 Defense Science Board report pointing out all the different things that can go wrong with computerized systems. Emphasis in that report was on the hardware and software area. Willis was here in 1979, at our first seminar. He and Dr. Dineen were the keynote speakers and he talked at some length about the fact that this area is still the significant problem area.

The fourth slide indicates that the areas of concern in computer security cover a very broad spectrum. As Mr. Klein just indicated, the area that we are most concerned about right now is the hardware and software area. That is where the primary emphasis of the Computer Security Initiative has been focused and will be the major concern of the Evaluation Center.

Gene Epperly and I put together the fifth slide back in 1978 to indicate how system approval works thru DoD Directive 5200.28, the DoD policy for computer security. This Directive establishes the concept of a Designated Approving Authority (DAA). The system builder brings his system to the DAA, a specific individual depending on the kind of data you are dealing with and where you are in the organization. If that DAA is convinced that all of the measures you have taken to protect that system are sufficient, he gives an individual site approval. Things that he worries about are physical, administrative and personnel security, TEMPEST, and communications security. Those are the measures we have used for over twenty years to protect information within computers. But of late, we have had a growing need to go beyond just having everybody on that computer cleared for access to all of the data. If we want this additional measure, we are going to have to rely on hardware and software mechanisms to provide this protection. The dotted arrow indicates that we were not in very good shape in this area in 1978. What we needed to do was strengthen that particular aspect as shown in the next slide, which indicates how we hoped things would be "soon," like in 1982. We needed a way to make that dotted arrow into a solid arrow. We needed an organization to provide advice to the DAA in the hardware and software security area similar to the advice that he could already get in the physical, administrative, and personnel areas. We also needed to get industry involved, because if we didn't we would have to keep developing all kinds of special-purpose systems. If we didn't get industry involved, we would never have trusted computers for our financial, logistics and administrative systems because we would never be able to afford them.

So the idea back then was that we should establish some kind of evaluation center—a center of excellence where really smart people can work closely with industry, understand the quality of their systems, and convey that understanding, by means of an evaluated products list (EPL), to the DAA. The EPL would provide the DAA with the information he needed to complete this additional aspect of his determination. He still has to worry about the physical, administrative, procedural and other aspects, but with the EPL he can, without becoming a computer science expert, figure out the significance of the hardware and software measures and what role they should have in his approval process.

The next slide is the version of an evaluated products list that came out of an NBS workshop in late 1978. Ted Lee was the author of that original report. Various versions of the EPL have evolved over time and this will have to be updated to the new system described yesterday. This slide indicates the hierarchical structure with the technical features that you can observe in a system listed on the left. These features are cumulative; a level 4 system assumes everything in level 3 and above. The environments where the system might be suitable for use are shown on the right. The DAA determines what those environments are with advice from the Center. What we need is an organization to develop the technical column and to evaluate systems to determine which ones fit where. That is the role of the Computer Security Evaluation Center.

The next slide shows the target schedule we came up with in 1978. I carefully put that dotted line in the middle between years because I wasn't sure when the formal part was really going to happen. As it turns out, the Evaluation Center was formed in July 1981, right on schedule. There are three parts to this process. The bottom line shows the evaluation phase. The portion we did under the auspicious of my organization was called informal evaluations. We initiated the examination of a number of industry systems. The specification phase (middle line) was the early work that we did to come up with the technical criteria for the EPL. The top line indicates the education phase, the various seminars and workshops that we put on, including this series at NBS, which I am very happy to see is continuing.

When I started back in 1978, it was clear that while something like this could start from the Office of the Secretary of Defense (OSD), we had to find an organization to take over the job, to "institutionalize" it. While we were carrying out the activities on this slide, we were also searching for the right place to do the job. On January 2, 1981, the Deputy Secretary of Defense assigned the responsibility for computer security evaluation to the Director of NSA. The rest of it you have already heard about. The Center formally began in July 1981, and you are seeing some of the results of its efforts at this seminar.

I wanted to go through a little of that history just to give you some perspective on where we have been. I think that the things you heard yesterday indicate that the strong thrust we have begun will continue. Now I would like to talk to you about some of the major DoD systems that are dependent upon a successful solution to the computer security problem.

The first system that I want to mention is the Defense Data Network, and if you don't mind my digressing a little bit, I want to discuss the AUTODIN II system and the process we have been through over the last two years.

The next slide indicates some of the history over the last 10 or 12 years in networks in the Defense Department. Back in 1969 the ARPAnet, the grandfather of all packet switch networks, began. By 1975, we transferred the operational management of the net from ARPA to the Defense Communications Agency (DCA). Versions of the ARPAnet were installed in various places. The World Wide Military Command and Control System (WWMCCS) began to explore using ARPAnet technology to link its major computers back in the 1975 time frame. The COINS network, which is an intelligence community network in the Washington area, started in the 1975 time frame and became operational about 1977. The Movement Information Net (MINET) is a version of the ARPAnet to be installed in Europe early next year.

In 1977, DCA began developing the AUTODIN II system which was to build an operational network. The target Initial Operational Capability (IOC) for AUTODIN II was 1979. In 1981 a partial IOC for AUTODIN II was declared. This slide tries to show how these various nets have evolved. Let me run through some of them now just to give you a perspective.

This next slide shows the ARPAnet. Right now it has about 95 nodes and 200 hosts. It extends from Hawaii to Norway. The next slide is a logical diagram of the net.

The next slide shows the WWMCCS Intercomputer Network (WIN). The major top secret command and control systems in the Defense Department are linked together in this network. Next is the Movement Information Net (MINET), basically an unclassified net in Europe. Initially, it will provide electronic mail for the transportation people to keep track of the movement of logistics material. It will rapidly become a packet switched backbone net within Europe.

The next slide is a diagram of the AUTODIN II system. There are four nodes in the initial system, as of October of last year. There are computer security implications throughout the design of this system, and I would like to point out a few of them. One of the decisions early in AUTODIN II was that the network should be able to process information at all security levels within the packet switch. Western Union decided to use security kernel technology. One of the implications of this decision was that all of the data on the network appears in the clear in the packet switch. This means that the switch has to be located in a physically secured area at system high, the highest level of data flowing in the net. All the people at the switch have to be cleared to have access to all the information on the net. These facilities and clearances are expensive and so the tendency is not to have very many. So in contrast to some of the nets I showed you which are proliferated to many sites, AUTODIN II was a net with very few nodes and very long access lines to those nodes.

The next slide depicts some of the activities that have happened over the last year and a half. In July 1980, AUTODIN II had just slipped to December 1980 and they were having difficulty running the systems tests. The Assistant Secretary for C^3I became concerned. He wanted to know, "What happens if Western Union could not deliver the system?" We were two years late on an original two-year program. He asked DCA to look at the options in case, for some reason, the system was not able to achieve IOC. In December, it slipped again to May 1981. The alternatives that we began to look at consisted of these other nets that I have shown you. The alternatives were not to start over again, not to build a new net, but to press on with the nets that we already have in place. In July of 1981, AUTODIN II achieved a partial IOC. The switches themselves were accepted, the terminal controllers were delayed for several months. In August 1981, the new Deputy Under Secretary for C^3I directed a detailed review of AUTODIN II and its alternatives. DCA formed three teams. One was designated to do the best job it could of enhancing AUTODIN II. The second team was to look at the ARPAnet, WIN and MINET and pull them together into a common network. The third team, headed by the Vice Director of DCA, reviewed the two other

teams' results. This review was an internal DoD review of existing systems, AUTODIN II and the other nets. In September 1982, the Defense Science Board was charged with a review of the DCA evaluation.

As shown on the next slide, in February of this year, the Director of DCA concluded that the replica system (the copy of the ARPAnet and WIN) provides the best Defense Data Network (DDN). In March, the Defense Science Board agreed with that conclusion. On March 10, the Telecommunications Council of DoD agreed that this was the right way to proceed. On March 12, the Under Secretary of Defense for Research and Engineering was briefed.

The next slide indicates the conclusions of the Deputy Under Secretary of Defense (DUSD) for C^3I. On April 2, the DUSD (C^3I) directed the termination of the AUTODIN II program and the immediate initiation of the Defense Data Network.

The next slide indicates how the DDN will evolve. The WWMCCS computer net will continue as a top secret system high net to which the intelligence community will be added. The critical element in this integration is a device called the Internetwork Private Line Interface (IPLI), a means of achieving a simple form of end-to-end encryption. I will talk more about this later.

The ARPAnet will be partitioned into two nets. One will be a residual R&D net. This will link the various universities and continue the R&D activities in networking. The other will be a military net, MILNET, which will consist of the operational military users. Primarily it will be an unclassified net, but the assumption is that all the users on it have sensitive data. All of the links on the MILNET will be protected with some form of encryption, and all of the nodes will have some degree of physical protection. MINET will be integrated into the MILNET sometime in 1983. SACDIN and a few other nets will be treated as special cases. They will be operated as dedicated networks until a production version of the IPLI or the Blacker system is available. At that time, we will be able to integrate all these pieces into one overall DDN.

The important thing to point out here is that we cancelled AUTODIN II because we already have a better alternative available. We do not have to begin a massive new development effort. We can build upon working systems in the field today.

I mentioned the computer security aspects of the problems with AUTODIN II. The DDN will use end-to-end encryption as its primary protection means. The data will be encrypted within the net, and we get away from the problem of having the switches in top secret system high facilities. That problem was one of the major flaws in AUTODIN II.

The next slide is a brief summary of the parts of the DDN. We are not starting over again. Many people ask, "When is the new RFP for the DDN going to happen?" It's not. We are evolving things that we have already in place. We will competitively procure a number of pieces of this, but we are not going to go out with any major procurement.

Now, I would like to talk about another of my favorite projects—the WWMCCS Information System (WIS)—which has very significant computer security implications. Steve Barnett mentioned this yesterday. I would like to amplify on what he said. This slide shows a map of the major sites where there are WWMCCS systems. These are the major command and control centers throughout the Department of Defense. The first place we get into a problem with computer security is looking up at that diagram. The majority of the systems in WWMCCS operate at the top secret level. There are a few that are intelligence community systems and then there are a few that operate only at the secret level. There are one or two unclassified systems for testing and development. When they decided to build the WWMCCS Intercomputer Net to link all these computers together, they were immediately stuck with the problem of how do we link someone cleared only to the secret level to an intelligence community or a top secret system? They decided that they can not do that, since we don't have a solution to the computer security problem. So they created a net that operates at the top secret level. Then they discovered that this network is really useful. I can send information all around, but what I really would like to do is get at Joe's information; unfortunately Joe is only at the secret level. How do we do that? One of the things that can be done is to upgrade Joe's facility to top secret. At FORSCOM in Georgia, the Army maintains status of forces information for the active Army and the national guard armories and a number of places like that around the country. It operates at a secret level. It doesn't need to operate higher than a secret level to do the job it is doing, and it can not, in some sense, because many of those places are hard to clear to higher that a secret level.

If you made the FORSCOM system top secret, you would have to upgrade all these National Guard armories and places like that to top secret. But various people said, "I really need that Army status of forces stuff." It was about a year and a half ago they decided to upgrade the FORSCOM system to top secret and put it on the WIN. What do we do about all these poor guys out there that are at the secret level? Well, one of the more interesting computer security experiments going on right now is called the FORSCOM GUARD; you may have heard of the various guard systems that are around. There is one that the Navy has at the ACCAT in San Diego. Another one is the FORSCOM guard to allow people to operate at a secret level and access secret level data on an otherwise top secret system. We are approaching real computer security solutions with efforts like this.

I went through this example to tell you about the nature of the computer security problem that WWMCCS has. Now I would like to describe what is going to happen to the overall WWMCCS system. We now have 28 sites, 35 systems and 87 CPUs involved in the WWMCCS system. They are primarily Honeywell computers with large mainframes, and many of them need to be replaced. The peripherals and the processors are getting old and the operating systems are not up to the new demands being made. We have a lot of options as to how to proceed. One of them is to carry out a massive replacement of all these big systems. We are not going to do that! Technically, we couldn't do it; politically, we couldn't do it. We are moving toward an evolution as illustrated in the next slide. We would like to get to Phase 4, a local net installed at each of the sites. This secure local network will have connections to the Honeywell 6000 and the WWMCCS Intercomputer Net so it can talk to the rest of the sites. This local net will also have specialized modules to do special functions. One of the more obvious ones is automated message handling.

Eventually, we plan to move things off the Honeywell 6000's and gradually eliminate them. That will happen at different times and different sites. It will happen in an evolutionary manner. So instead of going out with a massive RFP to totally replace the existing systems, we want to get into an evolutionary structure which will stay alive throughout the future.

One of the more interesting challenges before us in this is computer security. Right now these systems are basically run system high. Everybody is cleared to the same level on the Honeywell machines. This is unfortunate when you look at the experience, for example, at the Air Force Data Services Center where we have had a Multics systems in place for the last six or seven years that has been running top secret and secret. Users at both top secret and secret are on that system at the same time. If we can not at least achieve that kind of a capability in the upgrade of WWMCCS, then as far as I am concerned all of us computer security folks should just roll up our tents and go home.

We have to worry about that local net; we have to make sure that the local net is capable of handling, at least, those two levels of security. We would like to operate, at least, at the top secret and secret level. We can have individual hosts, like the Honeywell machines, that will remain at system high top secret, but we would like the message handling capability, for example, to operate over some range. Initially, it may only be top secret and secret; eventually it ought to be over a much broader range. It is crucial that we get the necessary pieces into this initial structure to allow us to do that. Otherwise, it will be another 30 years before we have the opportunity again, and we won't have multi-level security in the system until then.

That is why, as Steve Barnett mentioned yesterday, I believe the Center's involvement with WIS is absolutely crucial. We have got to succeed in getting the elements of computer security properly ingrained into WIS. Of course, this will also, I believe, serve as a model for a lot of other systems in the DoD. Nobody has as much energy or resources to put on an effort like this as these folks. And if they can do it right, we are going to see this same kind of structure apply many other places.

One of the things we have done recently to make sure that this all can happen is establish a WWMCCS Joint Program Manager. The next chart shows that there really is a joint program manager, Major General Don Evans. Many of you may have known him from SAC, one of the sharpest computer people at the general officer level in the Services. General Evans understands these issues and knows how to make these things happen. You will hear a lot more about this in the next few months, and years. It is probably going to be the most significant information system modernization the Defense Department is going to have over the next decade. Computer security will be a very significant part of it.

Another program that was mentioned yesterday is the Inter-Service/Agency Automated Message Processing Exchange (I-S/A AMPE). Some of the ironies of our computer security problem are well exemplified by this

system. Today we have a system called AUTODIN that was approved for use over the GENSER, unclassified through top secret community in 1965. In 1968, it was accredited for handling intelligence data. We now know some of the pitfalls in a system like this that we didn't really understand very well then. It would be very difficult to get this system accredited now. The irony of this situation is that in 1982, we have a system that we have been using since 1965, that is so difficult to replace that it is likely to be in the early 90s before we can.

The next slide shows how our message system might look in the 1990 time frame. The DDN will provide the transmission function, but the AMPE will have to provide the message integrity function. Today, once an AMPE sends a message to AUTODIN, it is guaranteed that the message will be delivered intact or that it will be notified to the contrary. With the DDN the message integrity function has to be part of the I-S/A AMPE.

This next slide is an attempt to show how we are going to transition from one to the other. The first I-S/A AMPE will talk to the AUTODIN system and through it to the rest of the AMPEs. The second one will have to do that, too, but hopefully the two of them will talk to each other over the Defense Data Net. Then, as we begin to put more and more new I-S/A AMPEs in, we will be able to phase out the AUTODIN system.

There are computer security problems all over this program. As I said, we accredited this system back in the '65 and '68 time frame. We have some tough problems to build a replacement for this using the technologies that we now have and knowing the risks that we now know.

The Computer Security Evaluation Center has a crucial role in making this happen. This is a very difficult process, and we did not create this Center any too soon. In fact, Steve Barnett, among others, may say that we did not create it soon enough to catch up with a problem like this. The fact is we have got to solve the computer security problems here. We are just beginning to see our way through. What we are going to do, probably, is a mixture of trusted computer systems and end-to-end encryption. Notice that when the I-S/A AMPEs come on to the Defense Data Network they will be coming on with end-to-end encryption boxes.

We are going to try to build a trusted computer base into the I-S/A AMPE initially that is sufficient to use over the GENSER community, top secret to unclassified. Eventually we want an A2 system, but we are probably going to shoot for an A1 system in the beginning. We don't believe the state of the art is there to be able to build a full trusted system across the board. And so we are going to use a combination of end-to-end encryption capabilities and trusted computing base capabilities.

Now I will spend just a few minutes talking about end-to-end encryption. I will give you some factual information and then some of my own observations.

Back in 1974, when I first went to ARPA, I was immediately hit with a problem. We have this great thing called ARPAnet, good for unclassified use. But we have a bunch of experiments running at ARPA that would like to be able to send classified data over the net. How do we do that? Well, we could secure the whole net, put all the nodes in secured facilities and use link encryption. It is a little hard to envision a secure facility at Berkley or MIT and some of these other places. And we didn't really want to build a totally new net, we wanted to work with what we had. In fact, we had a rather amazing set of constraints. We had host computers that were plugged directly into the net. The constraints we were given were: you can not change the net, you can not change the host computers. What you want to do is build a gadget that sits between the host and the net. We went to NSA and asked for an end-to-end encryption box that does that. And they asked, "When would you like it, in 95 or so?" And we said, "No, we would like it next year." They said, "You don't understand, it takes eight or nine years to build a new crypto system." So we thought about it for a little while and asked, "Do you have an existing device that we could use in this mode?" And after much searching around, we found a suitable encryption device. That was the third constraint and we were not allowed to change it either.

We started out with just a minicomputer inside the box. We planned to plug it into the host and plug it into the net and plug it into the KG and everybody would be happy. Well, wait a minute! No, unless you are going to solve the computer security problem, you better do something else, too. So, we put a big bulkhead in the middle there, and used a minicomputer on the host side and a minicomputer on the network side. The data comes in from the host, runs through the KG, and goes out to the net to another device,

through the KG, and on to the host. Well, amazingly enough, we built the box and it actually worked. There are a dozen or so of them in use right now.

One of the interesting points about this device is how it decides where to send the encrypted data. To do this, we poked a little tiny hole in the bulkhead, and we let a couple of bits of information dribble through. In reality, there is a table on each side of the bulkhead of the allowable addresses that this host can address. So when the device gets a message from the host at the same time that he is sending through the KG, he looks up the address and passes the index through the bypass. On the other side of the bulkhead the index is used to get the address, append it to the encrypted data and then the network knows what to do.

There are a number of applications of this device: it can be used for hosts talking to other hosts, or for terminal concentrators that can talk to hosts. The original device was approved in 1977. It only works on the ARPAnet. The little hole that we were able to poke through the bulkhead contains five bits of data for index and three bits of data for the size of the message. So there are only 32 subscribers that you can have on the net; that is far more than we have needed thus far, but it will be a constraint some day.

This device is a big box. There are two minicomputers. I don't have a picture of it, but it is about seven feet tall and weighs about 500 pounds, because it is a full TEMPEST box. It costs about $100K. And as I said, there are a dozen or so of them out there on the net. Well, we realized early in 1980 that we were likely to need a better version of this so we began an effort to build the Internet Private Line Interface (IPLI). I am talking about this in part because this is an absolutely critical part of the DDN. The ability to send classified data over a net that doesn't have to operate system high.

The Internet PLI provides the critical internet element allowing operation over multiple nets, from a local net at a WWMCCS site, across a common user net, perhaps to another common user net, to another local net. Well, that is four nets. And I have to be able to address multiple nets. This device will use the DoD Standard Internet Protocol, to allow operation over multiple nets. We made the hole in the bulkhead a little bigger. There are going to be 128 common subscribers on this device.

I have to comment that in poking the little hole we suddenly got a certain amount of sensitivity to the software that surrounds that hole. We do have pretty careful reviews of the software that drives the little hole. We are going to have more concern about that software now. It is not the software in the total system, it is just the software that actually makes use of the little hole.

We are getting down now to a pretty small piece of equipment, 2 cubic feet or so, about $25,000 apiece. We are now under contract and we hope to have them early in 1984.

What we would like to be able to get to, of course, is what people have talked about at some length—the notion of full end-to-end encryption. The PLI provides the simplest form of end-to-end encryption. The keys in these KGs are changed as frequently as required. All hosts who talk to one another have to have a common key. What you would like to be able to do is get into the position where you can have a key per connection, so that when one host decides to talk to another, a unique key is generated for that connection. Well, how do you do that? The essential new ingredient is a key distribution center which has some kind of access control. When one host decides to talk to another, he is first connected by his E^3 box to the KDC. It first checks whether he is allowed to make the connection, and if so it generates a key which is distributed to each host so that they can establish a connection. Nobody else ever has that key. It dies at the end of that session.

We have been working on this process for quite some time. It is the right long-term answer to the Defense Data Network and to a lot of similar applications.

Now I would like to give you just a little bit of conjecture on my part about the relationship between trusted computer systems and end-to-end encryption.

A trusted computing base is, as mentioned yesterday, the security perimeter, the part of the system that is relevant to making security decisions. We have done a lot of research on security kernels over the last few years and we have had some problems, in particular, performance problems. A lot of the hardware we are working on is not well suited to the kind of software we are finding we have to build. SCOMP is a good example of where we have made changes to the hardware to better adapt it to the software and see significant efficiency improvements. A lot of people though, in the last year or two, have been saying,

"Well, security kernels are really hard to build and they don't work very well and so maybe end-to-end encryption is really the right alternative to follow." What I would like to do in the next few minutes is give some of my reflections on that.

First of all, I would like to point out that security kernels and E^3 are complementary. There are places where you can not do anything other than with end-to-end encryption. But the critical part of an end-to-end encryption system is a trusted computing base that is included within it. A lot of people lose sight of that.

An example of what I am talking about is a message handling system. In a trusted message system you have a number of people who are on terminals, and each of them has a process that runs within the host computer. There is a trusted computing base aspect to that, shown in black on the next slide. These computers may, in fact, be connected over a network using some form of cryptographic protection. A lot of people say, "Trusted computing bases are hard to build," and "Why are we bothering with that? Let's try to find some easier way to do it." One easier way that has been conjectured is to put an end-to-end encryption device by the terminal and encrypt all the data. Then we don't need a trusted computing base in the computer because the only thing the computer is going to deal with is encryption.

A first glance at an E^3 message system on the next slide gives the impression that there are not any trusted computing bases in there at all. Well, a second glance (next slide) shows in the E^3 box, itself, there is a small problem because once the data has been encrypted, we have to worry about it being sent to the right place. So there is a trusted computing base there. There also is a big worry in the key distribution center. There has to be a considerable TCB there. Now as we look at this a bit further (next slide) we discover a problem with the location of the E^3 box. Who is going to do the processing of the text? That used to be done in the main computer. In this configuration the editing can not be done there because now the data is encrypted. So we have to add a processor on the outside of the encryption box in order to do the message generation editing and things like that. That makes it a little more complicated.

And then as we look at this problem even further, next slide, we realize that we have really got a trusted computing base of some sort in that new processor because it could really mess things up before the data was encrypted. And then we have to worry about the main processor dealing with things correctly. Is the data being put out on storage the right way? How about key management?

It turns out that we end up with quite a few trusted computing bases in this system. The point I want to make is that it is not necessarily easier and may be much more difficult to build an end-to-end encryption message system than to build a trusted version of the same thing. Trusted computing bases are an integral part of any end-to-end encryption system. We really don't understand the complexity of some of these systems. They could be a lot more difficult than we realize.

What I would like to do now is to give you some observations on some things that I have seen of late. I am really excited by what is happening here, for three reasons. One is the existence of the Center. The fact that there is now this institution with all these people working to make the concepts of the Computer Security Initiative happen. The second reason is that there is a realization in WIS and in the AMPE program and dozens of other programs that there really is a computer security problem and we have got to do something about it. It is ironic that a few years ago nobody would admit that they had a problem. People don't want to say that there is an integral part of their system that they don't have a solution for, because if you do, the budgeteers will cancel your system. And so people would not admit that they had a computer security problem when, in fact, they were tripping all over it. People are really coming out of the woodwork now. Part of that is because places like the Center are coming into existence and is trying to deal with the problem. Part of it also is the involvement of industry. I sat here yesterday listening to the discussion of the criteria and realizing that we have come a long way in the last few years. Then I sat here listening to six industry panelists talking about their own reactions to the criteria. There is a process that is being proposed for the evaluation of industry systems and nobody was violently upset about that. In fact, some people were pretty excited that it is actually going to happen. We have come quite a ways in the last few years.

Ted Lee (I hate to keep picking on Ted, but he is a great guy to pick on) made a couple of comments yesterday. He said, "Well, the vendors are just sort of waiting for the customers' reaction to these criteria. If the customers don't pay any attention to them, then we are not going to pay any attention." He also asked, after looking back at the past RFPs, where these criteria fit into the RFPs. The fact is the past RFPs

have been deficient in the computer security area largely because there weren't such criteria. We didn't know how to do that evaluation process. We are now beginning to understand and I can assure you, in programs like WIS and AMPE and dozens of others, these criteria are going to show up in RFPs. You are going to start seeing specific capabilities called for. There is a tremendous latent potential out there asking for trusted computer systems, and once we get the technical evaluation part of that process straightened out, it is going to be incorporated in these systems.

A lot of other exciting things are happening. Hilda Faust Mathieu talked about the consolidated R&D program. I want to point out that this is one of the things I have spent a great deal of my energy on in the last eight years: trying to coordinate R&D activities in computer security in the Defense Department. We did it very informally at first. I started a large number of programs when I was at ARPA, with a little funding from here, a little from there. Some of the Canadians who are here remember the KVM memorial "tank" that we bought. The Canadians wanted to contribute to the KVM effort back in the 1975–1976 time frame. It took us 27 months to figure out a way for them to contribute. And the only way we could get their money here was for them to go through the Foreign Military Sales Division as if they were buying a tank. And then the Foreign Military Sales guys gave us the money to put on the contract, having taken out their percentage for their services. So we have the KVM memorial tank, courtesy of the Canadians, for which I am very thankful. We also had joint programs with NSA, DCA, the Air Force and others.

When I moved to C^3I, and we formed the Computer Security Initiative, we had a very good informal relationship with all the Services, getting together and allocating the meager amount of money that each of us had to the important things that had to get done. Now we have a different situation. We have a focus on computer security in the Evaluation Center. It is crucial, as Hilda described yesterday, that the Services continue to be actively involved in this. But we must find a way to increase the level of energy. There was the DoD Committee on Fraud and Abuse. Jimmy Carter got mad about accusations of fraud in the government back in 1978. He wanted everybody to really give this high level attention, so each of the executive-level departments formed a Fraud and Abuse committee. The Deputy Secretary of Defense was the Chairman of the DoD Committee and there was a subcommittee on computer fraud. Hans Mark, who was then the Under Secretary of the Air Force, was head of that committee. They looked around trying to figure out what they could do to help solve our audit problems. They asked me to give them a briefing on what we were trying to do and about half way through, Hans Mark said, "What could you do if you had more money? How much money could you really use?" Well, we agreed that we ought to have a little bit more money associated with this. What followed was an incredible struggle to get these funds actually allocated for computer security R&D.

I just want to point out that even when you have the highest level of management behind you, the bureaucracy is so incredible at times. We spent months and months even when we had direction from the top to put money into these areas. What is happening in the Services is that we are competing with tanks, airplanes, ships and things like that. It is tough to compete for ADP resources, especially things like computer security R&D, against those kind of other alternatives.

Now we have a situation that is different. We have a new management structure with the Evaluation Center. It reports directly to the Director of NSA and from there to OSD. The money that is going to be in that consolidated R&D program is not going to compete with ships, tanks and airplanes. It is going to have a much more understanding path to getting approved. It is the kind of environment where if somebody at a high level decides this needs this kind of attention, there is a much better chance that it is going to survive through the bureaucracy.

The way that the consolidated program is going to work is as follows. Each of the Services and Agencies that want to participate will identify the things it wants to do or the things it thinks are necessary in the generic R&D sense. We are beginning in the FY84 budget submission, which is going in right now, to consolidate these ideas into a list. There is an amount of money associated with this list which the Director of NSA will submit directly to OSD. It doesn't go through all these various other drills. That is not to take the place of specific Service applications. Things that the Services are spending money on in computer security for particular projects will continue within the Services. The Consolidated Program is a valuable vehicle for providing additional money for the fundamental generic R&D that needs to be done.

There are other things happening that are exciting. Marv Schaefer mentioned that when he first met me, he was talking about a trusted data base management idea back in the 1975 time frame. This summer the Air Force is putting on a summer study on trusted data base management systems. It just so happens that Marv Schaefer is the Chairman of that very exciting activity. We hope from this summer study will come all kinds of ideas for R&D in trusted data base management systems. One of the more pressing needs we see in these systems is the ability to handle data base management problems in a computer security context.

I would like to personally thank Pete Tasker and all the folks at MITRE who have been involved with me from the beginning back in 1978 who have made the things that have happened possible. Pete and his folks are still involved with the Center and will continue in the future, and I am really glad to see that.

I would like to issue a challenge to each one of you here. If you work for the computer manufacturing industry, I challenge you to get your company involved. A lot of you have already; there is a lot of progress out there. It is important that you dive in and get your company involved. If you are from the government and you are involved in any of the systems like the ones I was talking about today, don't be afraid that computer security is something you will never achieve in that system. Get involved with the Center. Understand what can be done. Make sure the computer security requirements are fully addressed in the systems you are building.

If you are from industry, but not working for manufacturers, it is absolutely crucial that the manufacturers get pressure not just from the Defense Department for computer security, but also from the banks and financial institutions. It is also very important that the Defense Department, in putting out its demands for computer security capabilities, does so in a context that is suitable for other applications, for banks and insurance companies and so forth. If we start asking for things that are useful only to us, that nobody else can make use of, the manufacturers are going to resist, claiming we are too small a part of the market. We have a responsibility to make sure the things we are asking for are things the manufacturers can sell elsewhere. In that sense, the Center and our activities here can serve as catalysts to significantly improve the quality of computer systems in the United States and beyond.

If you are a part of the Computer Security Evaluation Center, I really challenge you. You have a very tough job. As you can see with the various systems I talked about, and there are dozens of them out there, we have to do something to get computer security cranked into these systems. Mel said he wasn't doing a sales job; Hilda said she was; I am too. For any of you out there who are looking for interesting and challenging things to do, there are openings at the Center. There are plenty of places where good, qualified people can come to work.

Several years ago, Ted Lee sent me a cartoon that appeared in the *New Yorker*. A security-conscious homeowner is attaching a seventh lock on his door as the floor is being sawed out from under him. It is applicable to those of us worrying about computer security. You have to make sure that you don't put too much emphasis in one place and forget about the others. This cartoon has served as a reminder to me that we must not go overboard in any one of these areas. Make sure you look all the way across the board.

I am very pleased to be here. I am excited at all the things that are happening. I am proud that I was able to be part of it. □

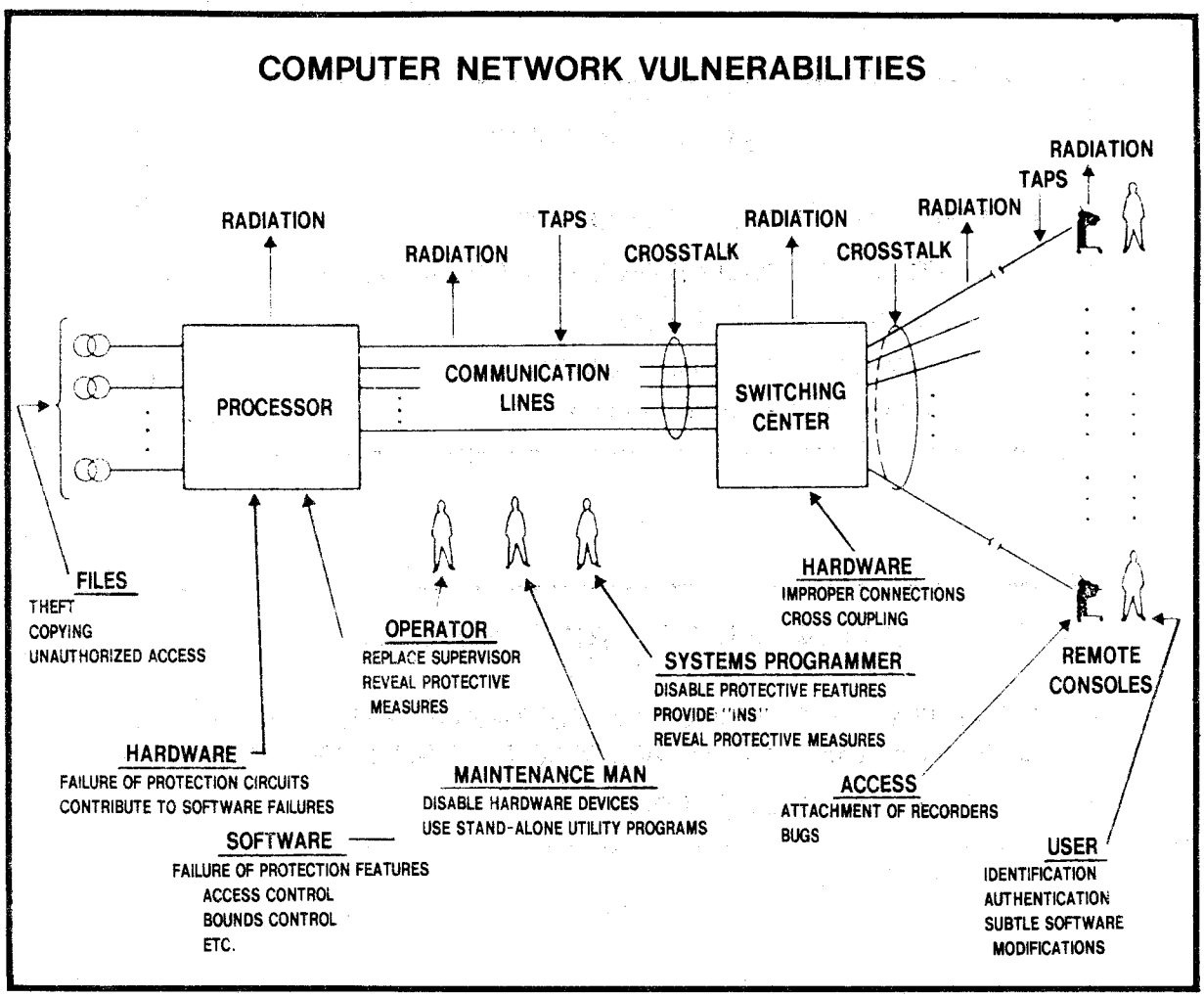
STATUS REPORT

- COMPUTER SECURITY INITIATIVE
- RELATED DOD COMPUTER SECURITY ACTIVITIES
  - — DEFENSE DATA NETWORK
  - — WWMCCS INFORMATION SYSTEM
  - — INTER SERVICE/AGENCY AMPE

COMPUTER SECURITY INITIATIVE

TRUSTED: SUFFICIENT HARDWARE AND SOFTWARE INTEGRITY TO ALLOW SIMULTANEOUS USE AT MULTIPLE SECURITY/SENSITIVITY LEVELS

WIDESPREAD: COMMERCIALLY SUPPORTED


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

COMPUTER SECURITY

PHYSICAL SECURITY

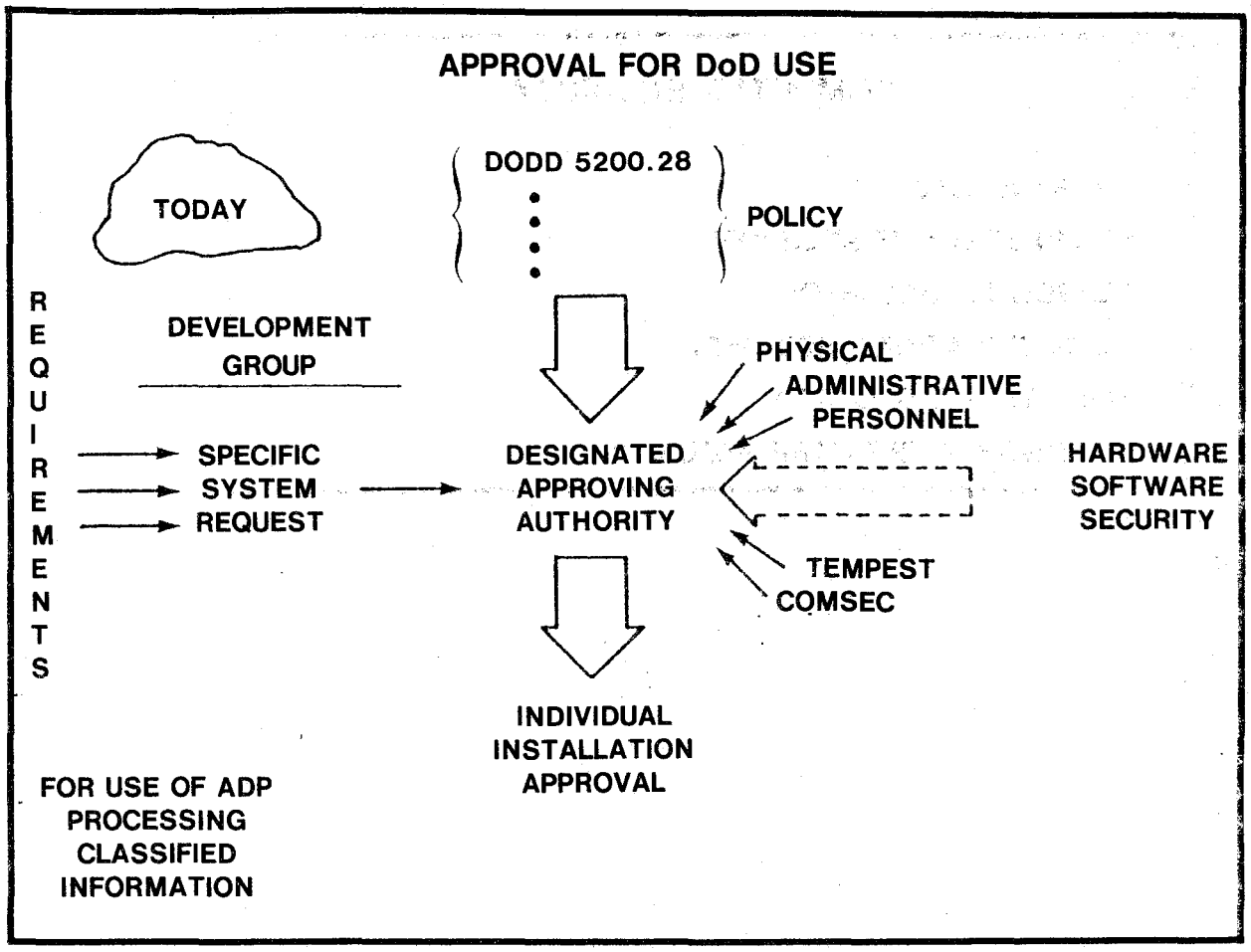
ADMINISTRATIVE SECURITY

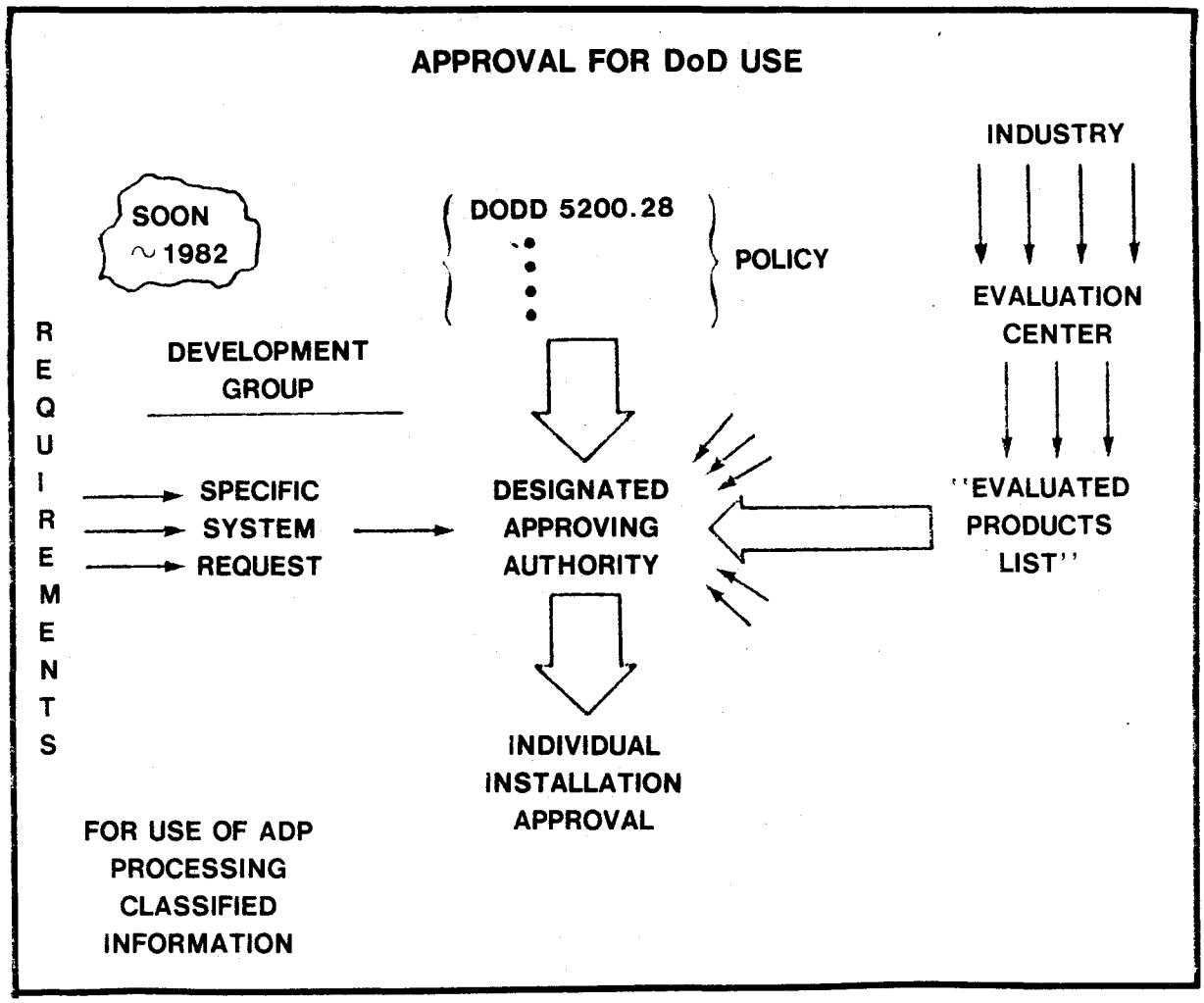
PERSONNEL SECURITY

COMMUNICATIONS SECURITY

EMANATIONS SECURITY

HARDWARE/SOFTWARE SECURITY

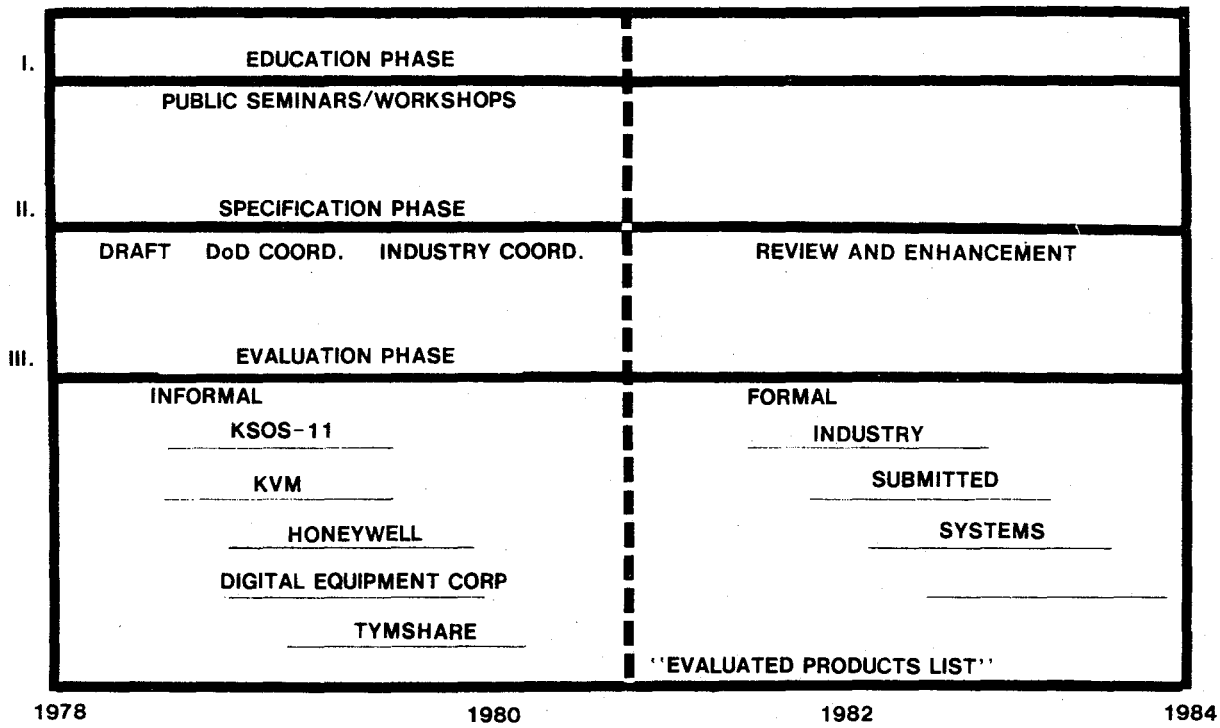

APPROVAL FOR DoD USE
TODAY
DODD 5200.28
POLICY
REQUIREMENTS
DEVELOPMENT GROUP
SPECIFIC SYSTEM REQUEST
DESIGNATED APPROVING AUTHORITY
PHYSICAL
ADMINISTRATIVE
PERSONNEL
HARDWARE
SOFTWARE
SECURITY
TEMPEST
COMSEC
INDIVIDUAL INSTALLATION APPROVAL
FOR USE OF ADP PROCESSING CLASSIFIED INFORMATION


APPROVAL FOR DoD USE
INDUSTRY
SOON
~1982
DODD 5200.28
POLICY
EVALUATION
CENTER
REQUIREMENTS
DEVELOPMENT
GROUP
SPECIFIC
SYSTEM
REQUEST
DESIGNATED
APPROVING
AUTHORITY
"EVALUATED
PRODUCTS
LIST"
INDIVIDUAL
INSTALLATION
APPROVAL
FOR USE OF ADP
PROCESSING
CLASSIFIED
INFORMATION

EVALUATED PRODUCTS LIST

CLASS	TECHNICAL FEATURES	EXAMPLES	POSSIBLE ENVIRONMENTS
1	—	MOST COMMERCIAL SYSTEMS	DEDICATED MODE
2	FUNCTIONAL SPECIFICATION REASONABLE PENETRATION RESULTS	''MATURE'' ''ENHANCED'' OPERATING SYSTEM	BENIGN, NEED TO KNOW ENVIRONMENTS
3	REASONABLE MODERN PROGRAMMING TECHNIQUES LIMITED SYSTEM INTEGRITY MEASURES	MULTICS	AF DATA SERVICE CENTER TS-S
4	FORMAL DESIGN SPECIFICATIONS SYSTEM INTEGRITY MEASURES		NO USER PROGRAMMING TS-S-C
5	PROVEN DESIGN SPECIFICATIONS VERIFIABLE IMPLEMENTATION LIMITED COVERT PATH PROVISIONS	KSOS KVM	LIMITED USER PROGRAMMING TS-S-C
6	VERIFIED IMPLEMENTATION AUTOMATED TEST GENERATION EXTENDED COVERT PATH PROVISIONS REASONABLE DENIAL OF SERVICE PROVISIONS		FULL USER PROGRAMMING TS-S-C

COMPUTER SECURITY INITIATIVE



COMPUTER SECURITY INITIATIVE

ON JANUARY 1, 1981 THE SECRETARY OF DEFENSE ASSIGNED RESPONSIBILITY FOR COMPUTER SECURITY EVALUATION FOR DOD TO THE DIRECTOR, NATIONAL SECURITY AGENCY.

DEFENSE DATA NETWORK

— HISTORY

— FUTURE

STATUS REPORT

RELATED DOD COMPUTER SECURITY ACTIVITIES

— DEFENSE DATA NETWORK (DDN)

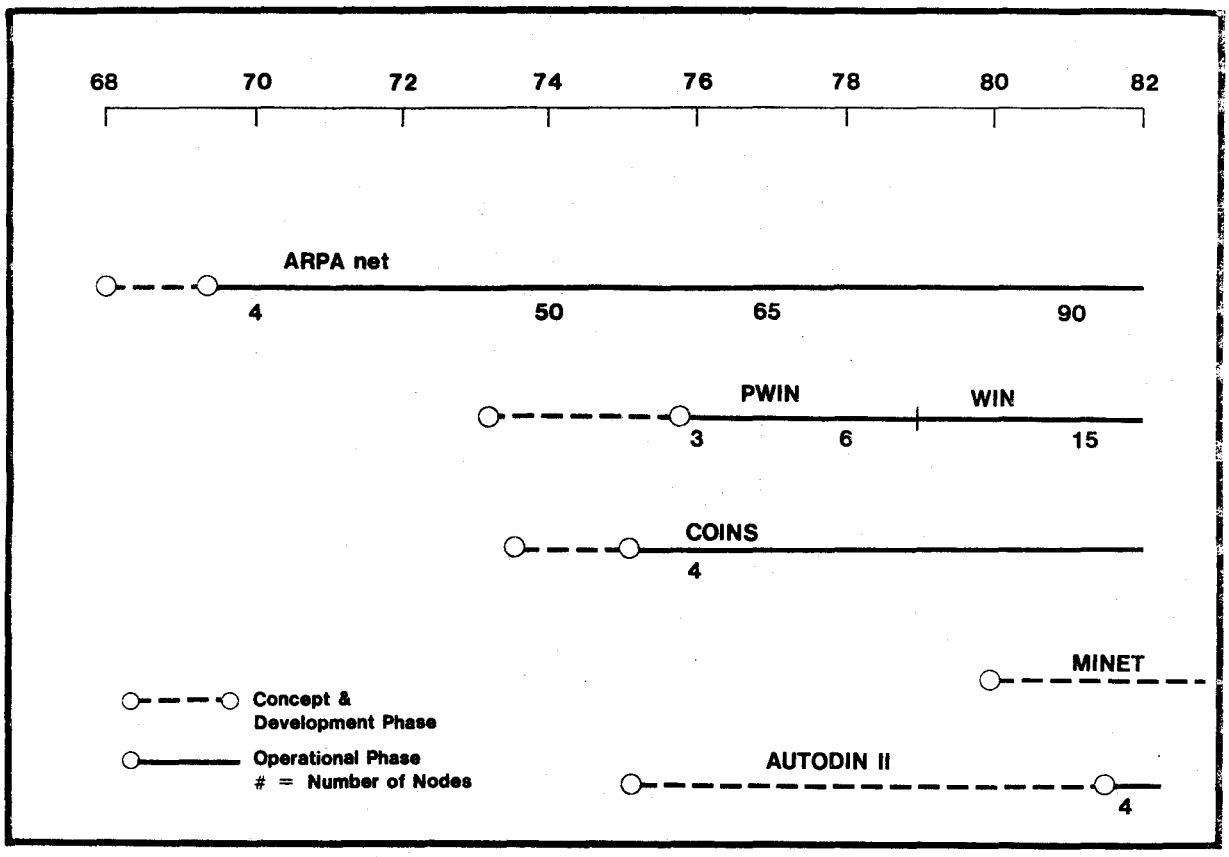
— WWMCCS INFORMATION SYSTEM (WIS)

— INTER SERVICE/AGENCY AMPE

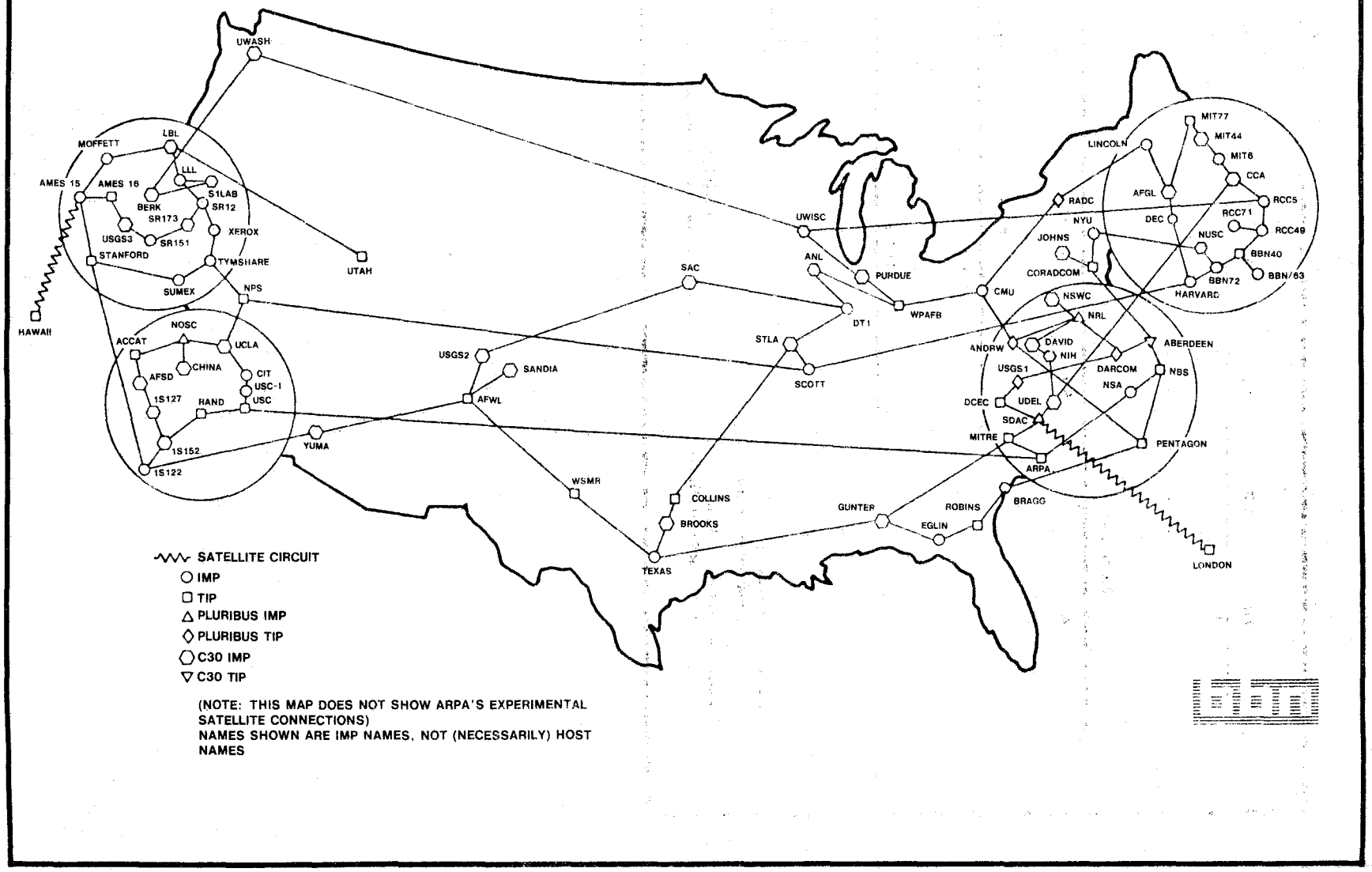
DOD DATA NETWORK HISTORY

- 1969 ARPA NETWORK
- 1975 PWIN — WIN (WWMCCS INTERCOMPUTER NETWORK)
- 1975 COMMUNITY ON-LINE INTELLIGENCE NETWORK (COINS)
- 1982 MOVEMENT INFORMATION NETWORK (MINET)

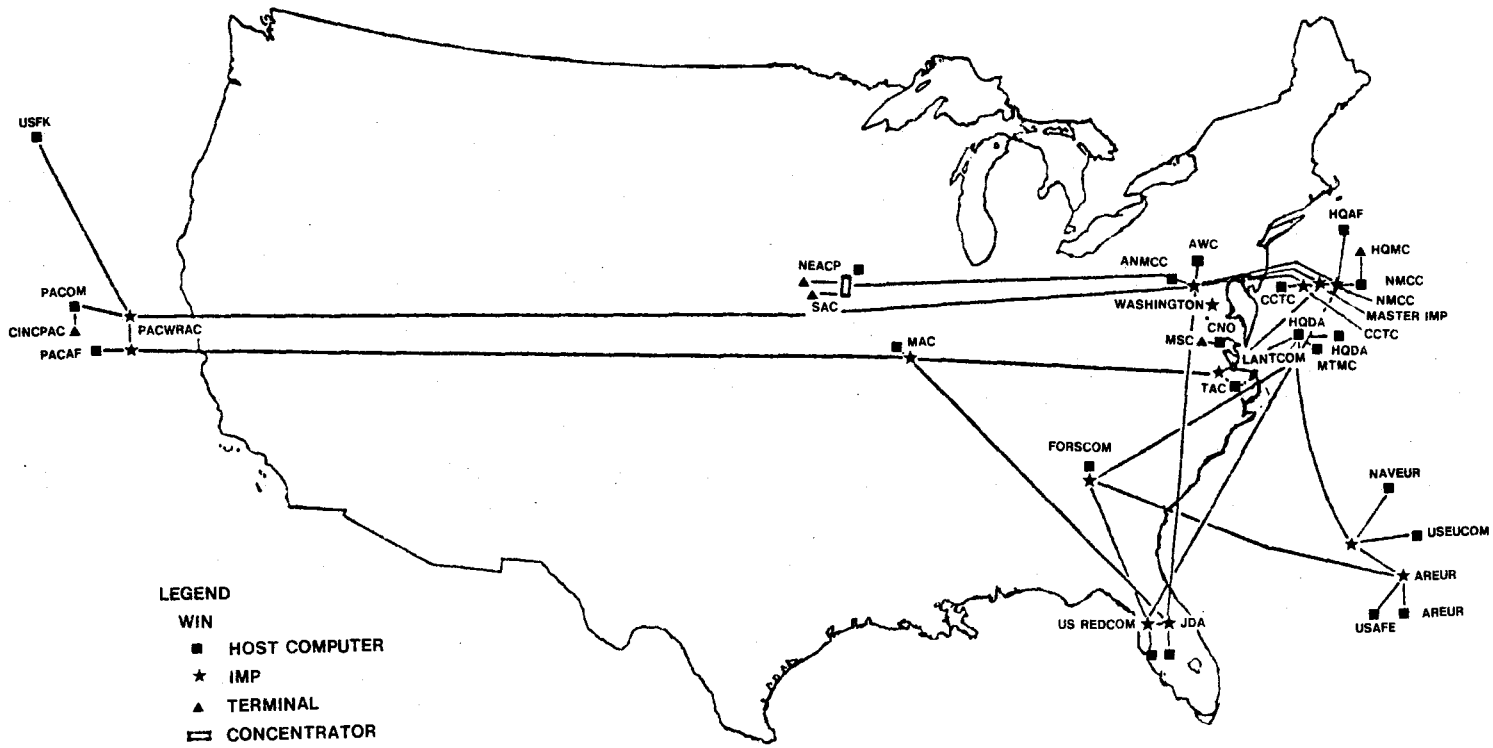
- 1977 AUTODIN II DEVELOPMENT BEGUN
  — TARGET IOC — 1979
- 1981 — PARTIAL IOC DECLARED


68
70
72
74
76
78
80
82
ARPA net
4
50
65
90
PWIN
WIN
3
6
15
COINS
4
MINET
Concept &
Development Phase
Operational Phase
= Number of Nodes
AUTODIN II
4

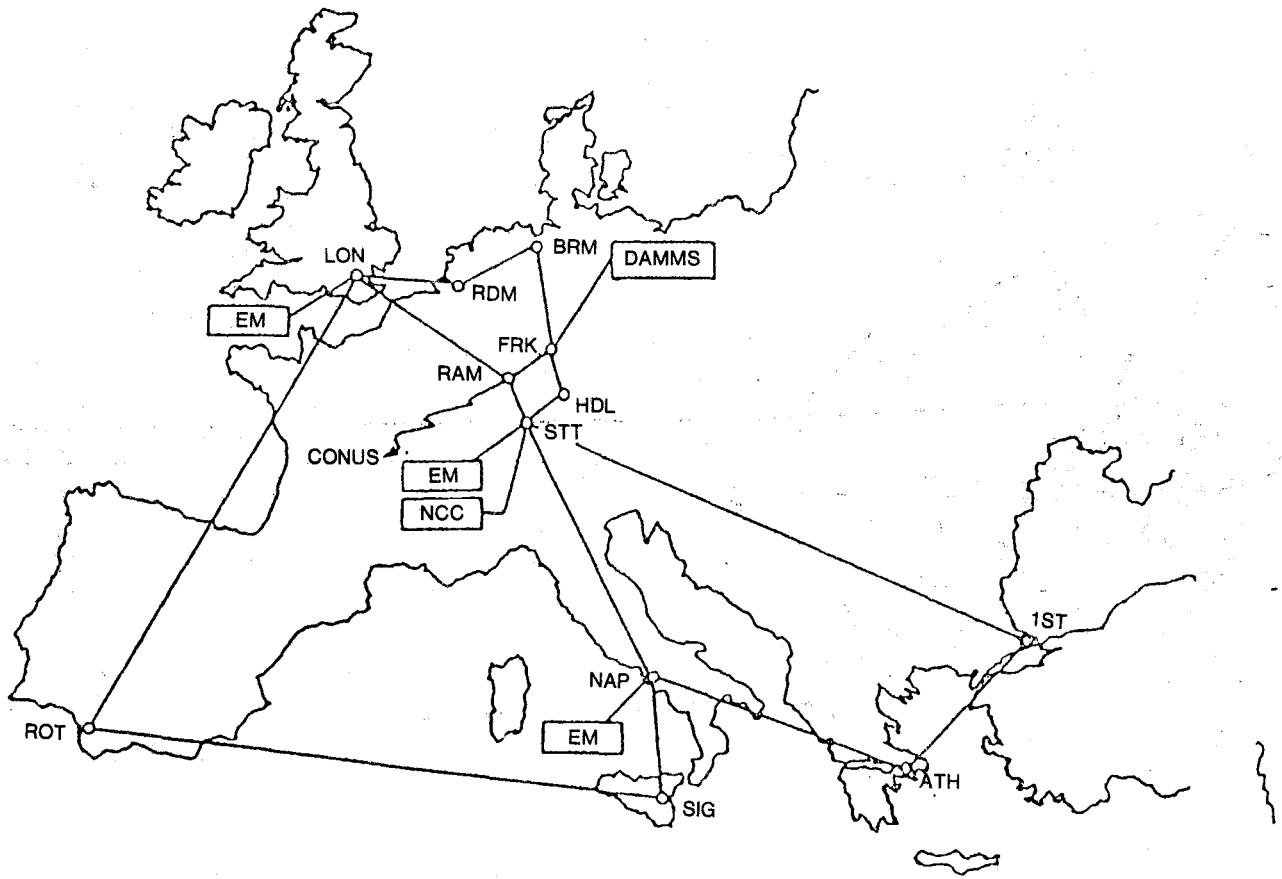
ARPANET GEOGRAPHIC MAP, APRIL 1982



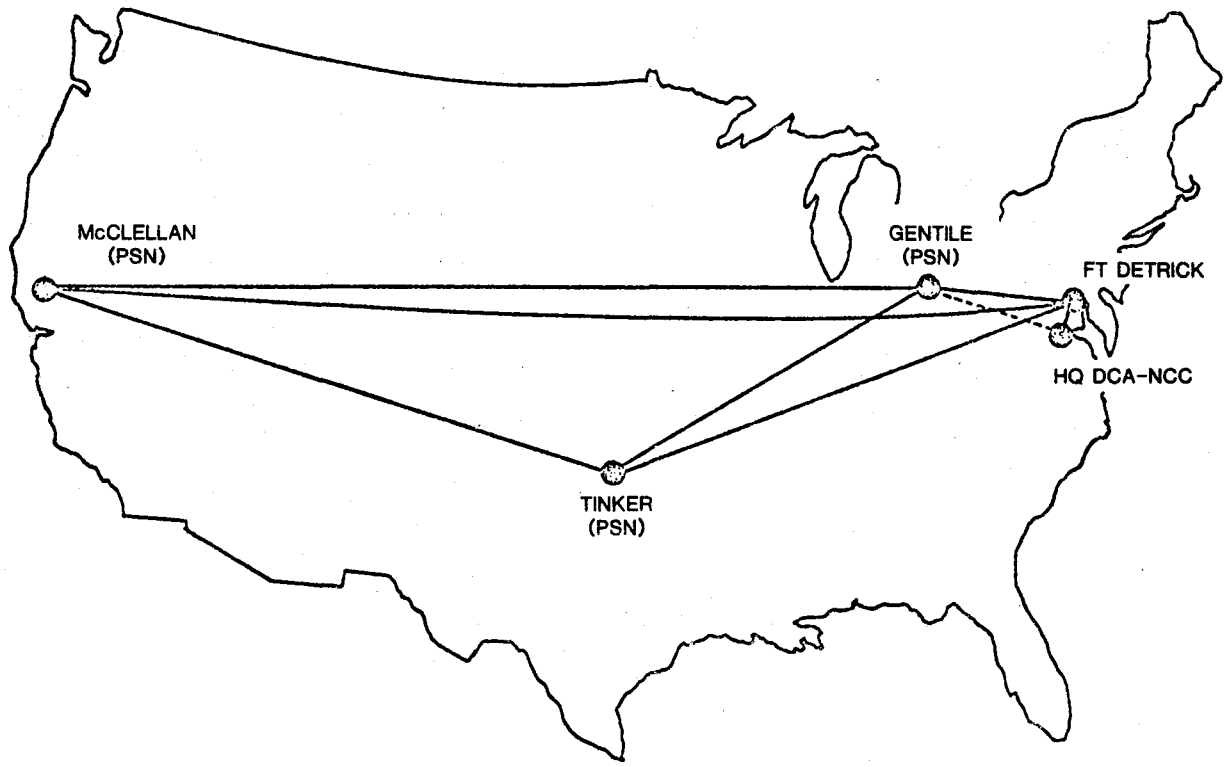
WWMCCS INTERCOMPUTER NETWORK



MOVEMENT INFORMATION NETWORK (MINET)



AUTODIN II
PHASE I
INITIAL 4-NODE CONFIGURATION



RECENT DATA NETWORK EVENTS

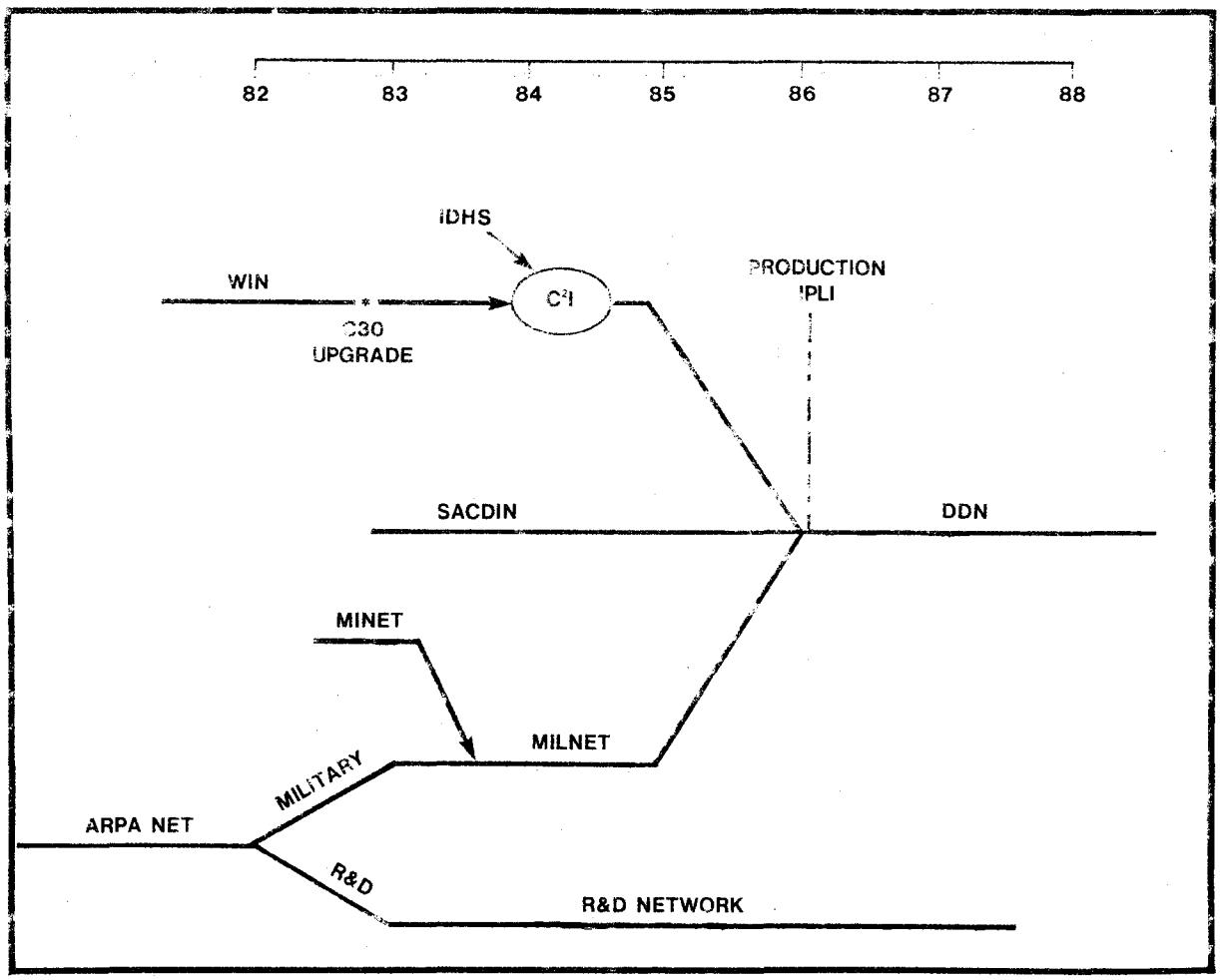
- JULY 1980 — AUTODIN II IOC SLIPPED TO DEC 1980
  — ASD (C^3I) DIRECTS REVIEW OF ALTERNATIVE IF AUTODIN II SHOULD FAIL TO ACHIEVE IOC
- DEC 1980 — AUTODIN II IOC SLIPPED TO MAY 1981
  — ALTERNATIVE CONSISTING OF EVOLUTION OF EXISTING DATA NETWORKS DESCRIBED
- JULY 1981 — PARTIAL IOC OF AUTODIN II DECLARED
- AUG 1981 — DUSD (C^3I) DIRECTS DETAILED REVIEW OF AUTODIN II AND ALTERNATIVE
  — DCA BEGINS 5 MONTH REVIEW OF BOTH OPTIONS
    — 3 TEAMS FORMED
      — AUTODIN II
      — ARPAnet/WIN REPLICA
      — EVALUATION TEAM, CHAIRED BY DCA VICE DIRECTOR
- SEPT 1981 — DEFENSE SCIENCE BOARD AUTODIN II TASK FORCE CHARTERED

EVENTS LEADING TO PRESENT POSITION

- FEB 26, 1982 — DCA CONCLUDES:
  ''REPLICA APPROACH PROVIDES BETTER DEFENSE DATA NETWORK''
- MAR 9, 1982 — DSB TASK FORCE RECOMMENDS:
  ''OUR REVIEW FAVORS WIN/ARPAnet.
  MAKE CHOICE PROMPTLY.''
- MAR 10, 1982 — DOD TELECOMMUNICATIONS COUNCIL MEETING
  — DCA AND DSB CONCLUSIONS BRIEFED
  — DUSD (C^3I) ANNOUNCED INTENTION TO RECOMMEND TERMINATION OF AUTODIN II
  — SERVICE COMMUNICATIONS CHIEFS CONCURRED
- MAR 12, 1982 — USD (R&E) BRIEFED ON DCA, DSB, AND DUSD (C^3I) RECOMMENDATIONS

DUSD (C^3I) CONCLUSIONS

- GO WITH REPLICA SYSTEM
- TERMINATE AUTODIN II IMMEDIATELY
- AGAIN UNDERSCORE COMMITMENT TO COMMON USER SYSTEMS
- PLACE ALL DDN RELATED DEVELOPMENT ACTIVITIES UNDER A SINGLE PROGRAM MANAGER AT DCA
- FOLLOW EVOLUTIONARY APPROACH TO REPLICA SYSTEM
  - PAY PARTICULAR ATTENTION TO PROBLEMS OF SECRET LEVEL USERS



SUMMARY OF ARPANET/WIN APPROACH

- EVOLUTION OF EXISTING DATA NETWORKS
- COMMON TECHNOLOGY USED THROUGHOUT
- PROVEN APPROACH WITH OVER 10 YEARS' EXPERIENCE
- OVER 50% OF COST IS GOVERNMENT LEASED COMMUNICATIONS CIRCUITS FROM COMMON CARRIERS
- ALL NEWLY DEVELOPED HARDWARE WILL BE COMPETITIVELY PROCURED
- BOLT BERANECK AND NEWMAN WILL FURNISH
  - NODAL HARDWARE FROM EXISTING CONTRACTS
  - SYSTEM ENGINEERING SUPPORT

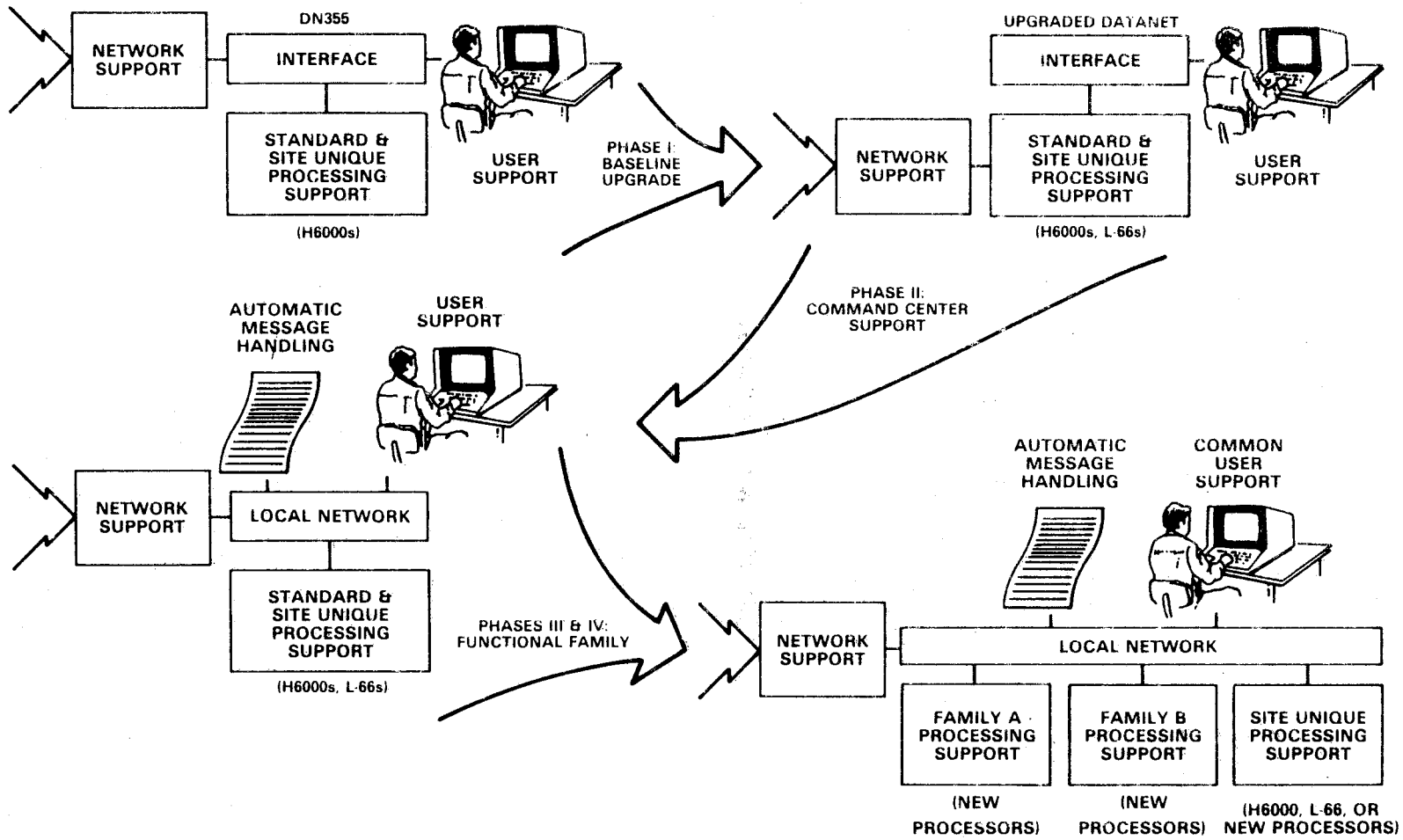
WWMCCS INFORMATION SYSTEM

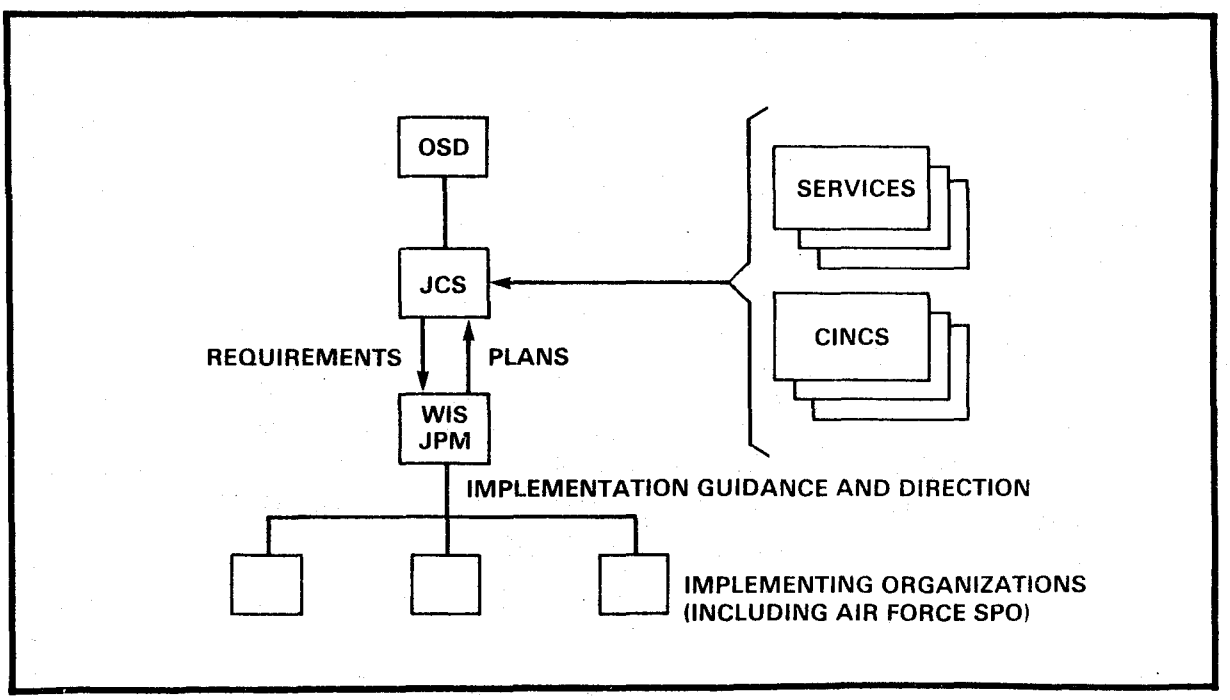
- ARCHITECTURE
- MANAGEMENT

WWMCCS INFORMATION SYSTEM

(Site Map Not Available)

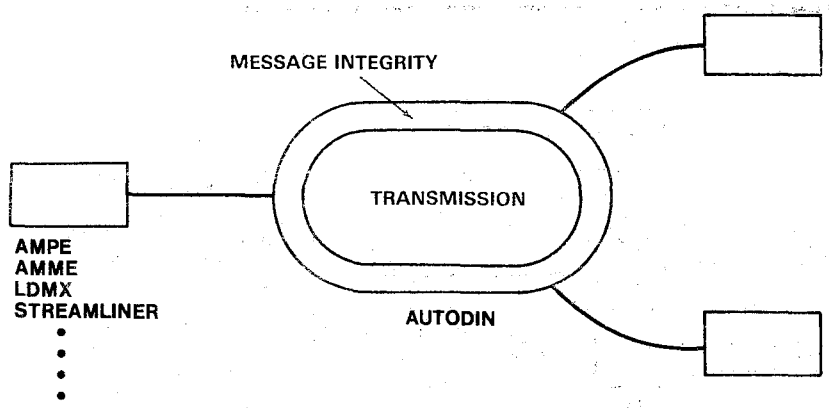
ARCHITECTURAL PHASES




OSD
JCS
REQUIREMENTS
PLANS
WIS
JPM
SERVICES
CINCS
IMPLEMENTATION GUIDANCE AND DIRECTION
IMPLEMENTING ORGANIZATIONS
(INCLUDING AIR FORCE SPO)

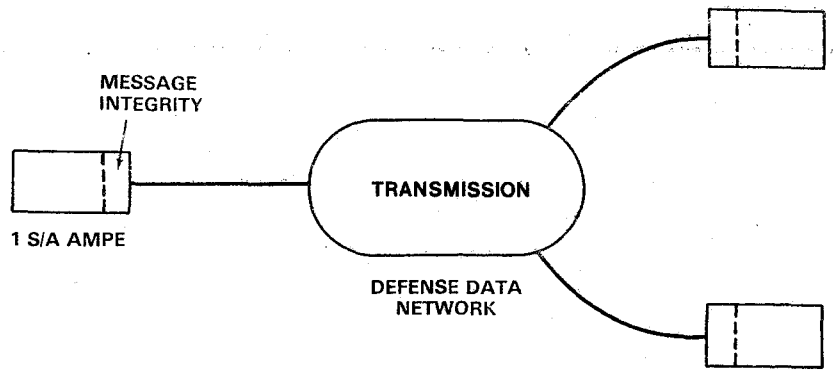
INTER SERVICE/AGENCY AUTOMATED MESSAGE PROCESSING EXCHANGE

TODAY

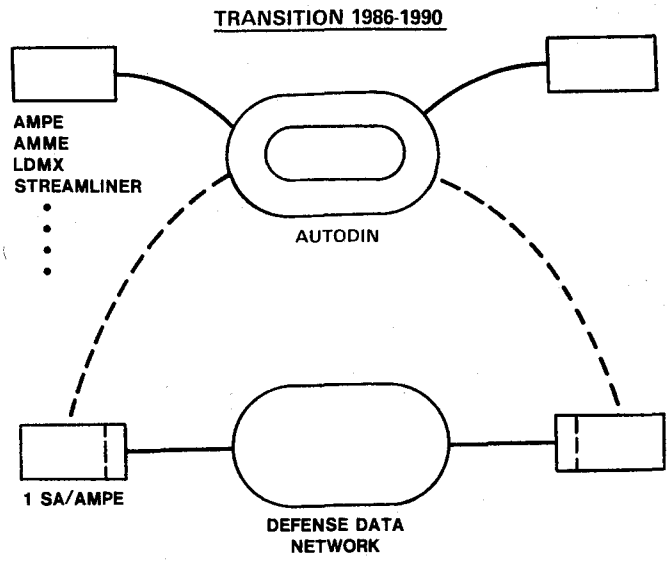


INTER SERVICE/AGENCY AUTOMATED MESSAGE PROCESSING EXCHANGE

1990



INTER SERVICE/AGENCY AUTOMATED MESSAGE PROCESSING EXCHANGE



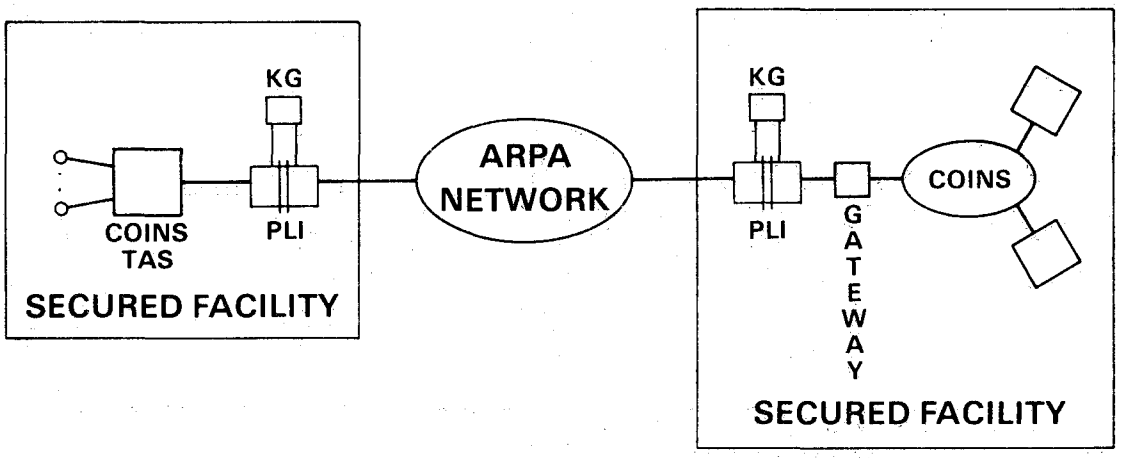
PRIVATE LINE INTERFACE (PLI)

- SINGLE ARPA LIKE NETWORK
- UP TO 32 COMMON SUBSCRIBERS
- FULL RACK IN TEMPEST BOX
- ~$75K PER DEVICE PLUS KG
- APPROVED IN 1977

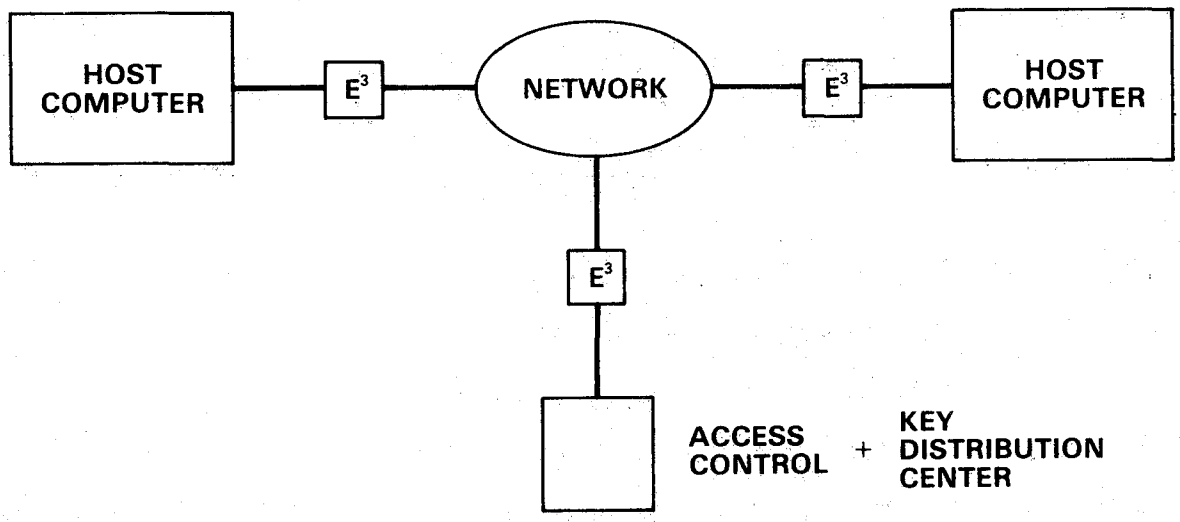
INTERNETWORK PLI

- WORK ACROSS MULTIPLE PACKET SWITCHED NETWORKS
- UP TO 128 COMMON SUBSCRIBERS
- 2 CUBIC FEET, ~ $25K PER DEVICE
- UNDER CONTRACT, AVAILABLE SPRING 1983

COINS — CINCPAC LINK



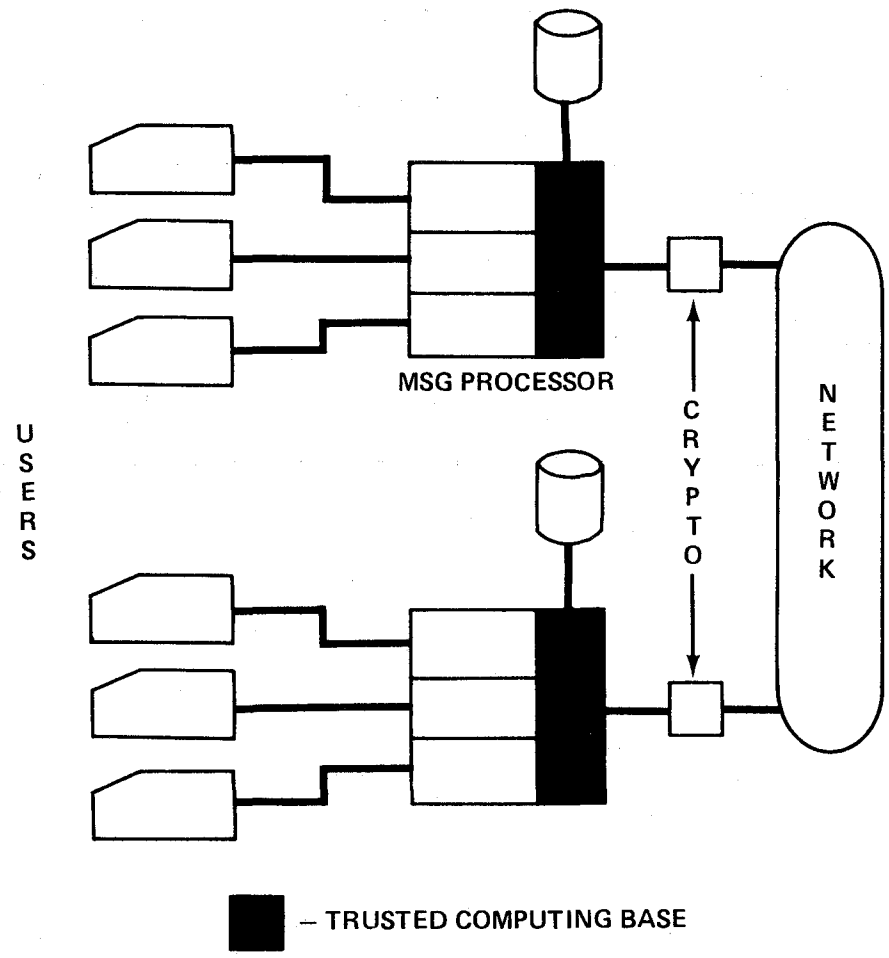
END TO END ENCRYPTION (E^3)



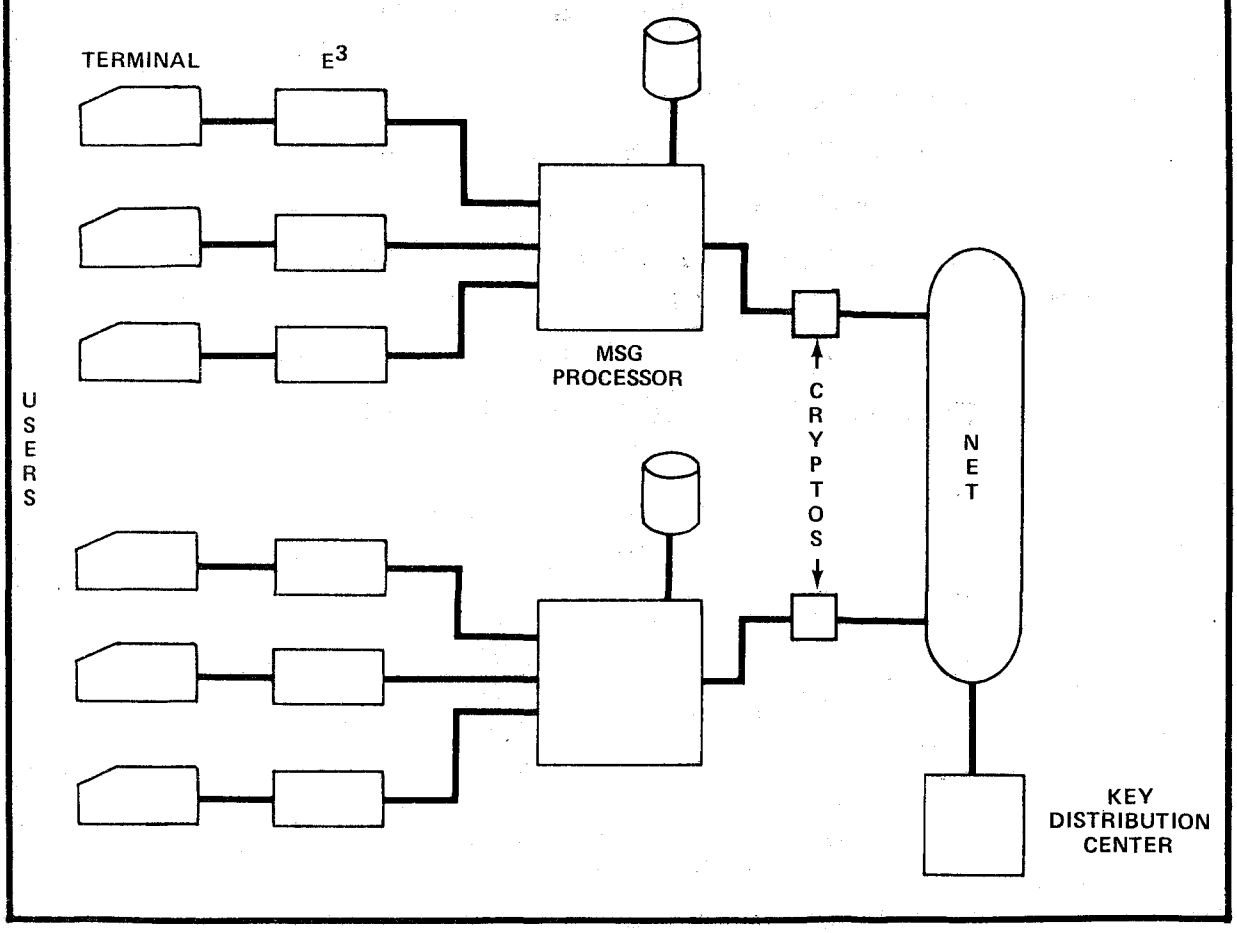
TRUSTED COMPUTER SYSTEMS
VS
END TO END ENCRYPTION

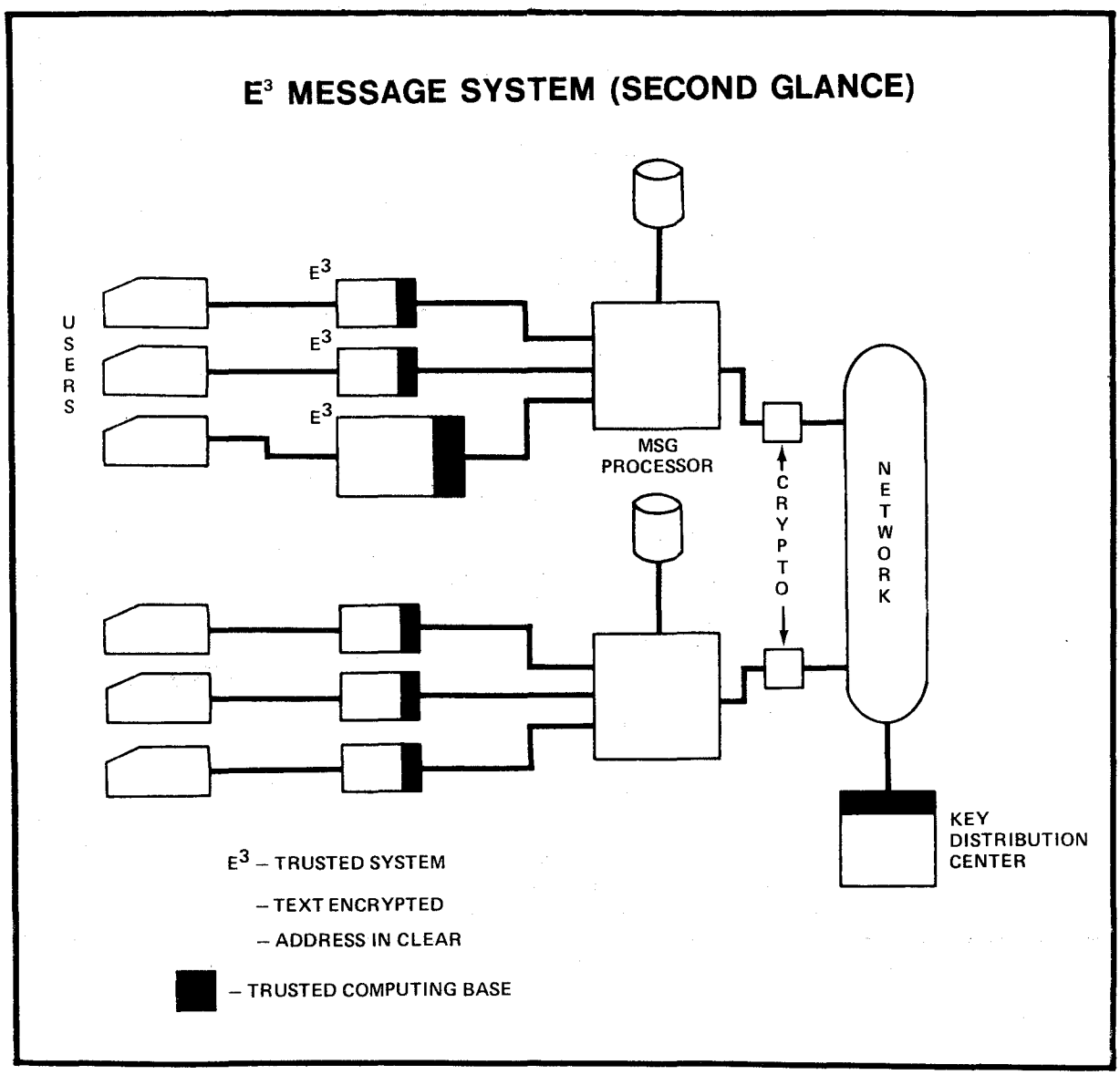
- COMPLEMENTARY
- E^3 — NEEDED FOR COMMUNICATIONS NETWORK
- MAY BE USEFUL IN FILE STORAGE
- NEEDS TRUSTED SYSTEM FOR SEVERAL FUNCTIONS

TRUSTED MESSAGE HANDLING SYSTEM

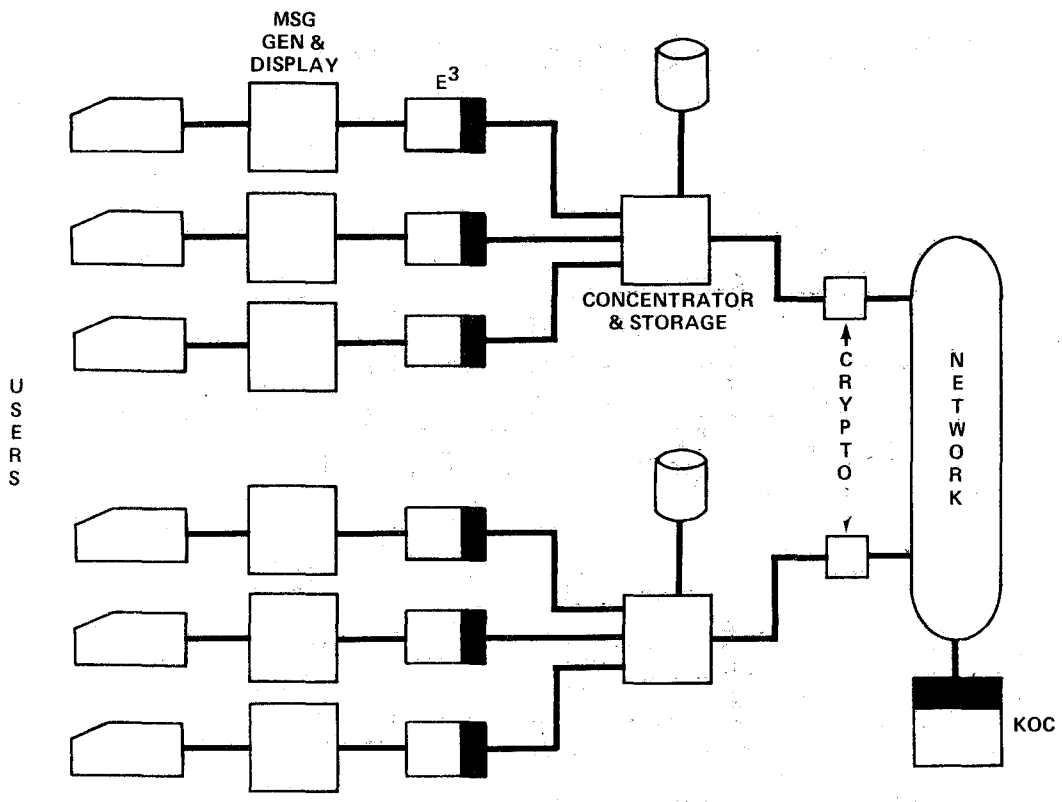


END TO END ENCRYPTION (E^3) MESSAGE SYSTEM (FIRST GLANCE)



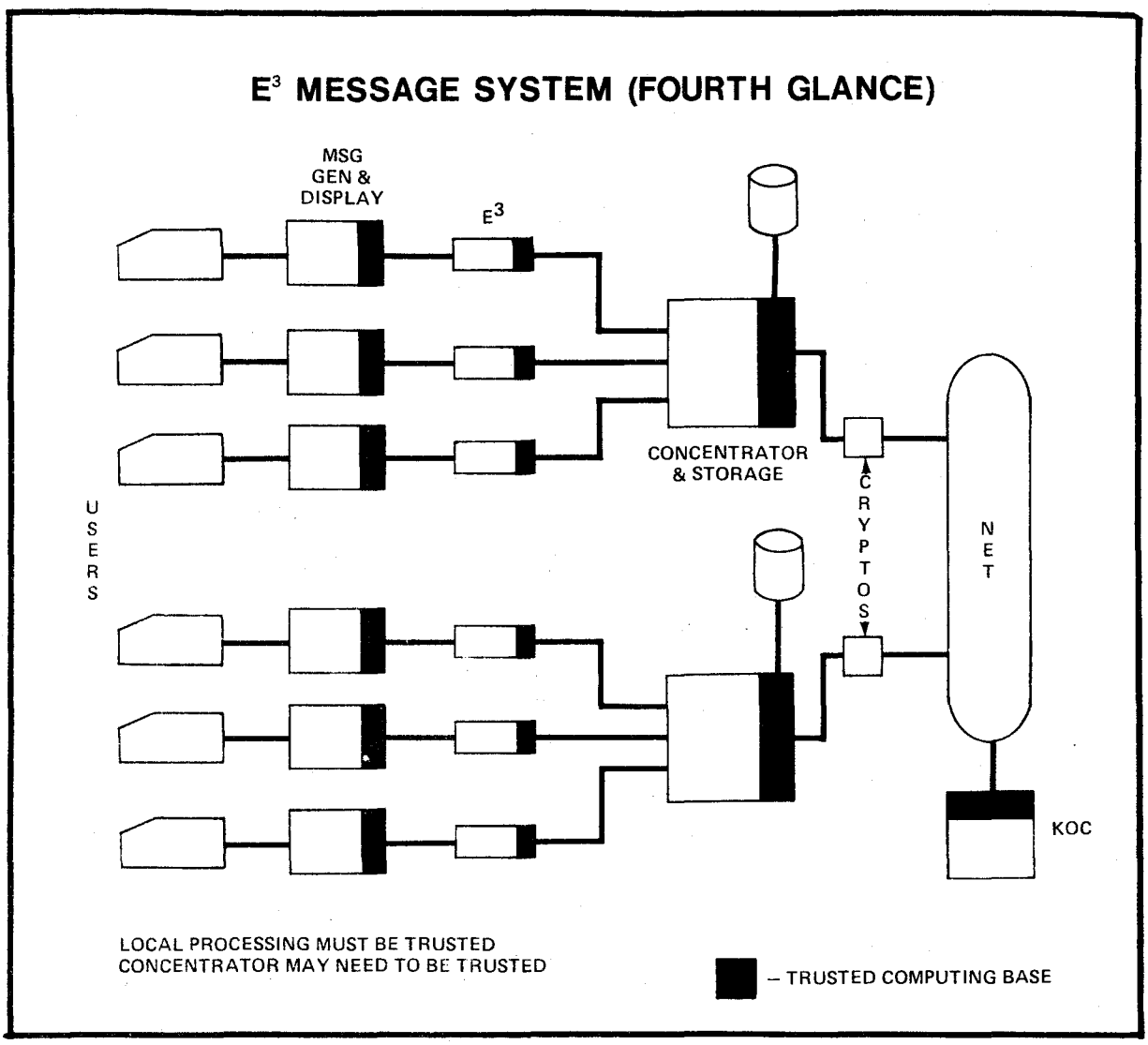

E³ MESSAGE SYSTEM (SECOND GLANCE)
USERS
E3
E3
E3
MSG PROCESSOR
CRYPTO
NETWORK
KEY DISTRIBUTION CENTER
E3 – TRUSTED SYSTEM
– TEXT ENCRYPTED
– ADDRESS IN CLEAR
– TRUSTED COMPUTING BASE

E^3 MESSAGE SYSTEM (THIRD GLANCE)



NEED DEDICATED LOCAL PROCESSING
BECAUSE TEXT ENCRYPTED

■ – TRUSTED COMPUTING BASE


E³ MESSAGE SYSTEM (FOURTH GLANCE)
MSG GEN & DISPLAY
E³
CONCENTRATOR & STORAGE
CRYPTOS
NET
USERS
KOC
LOCAL PROCESSING MUST BE TRUSTED
CONCENTRATOR MAY NEED TO BE TRUSTED
– TRUSTED COMPUTING BASE

COMPUTER SECURITY POLICIES: CHALLENGES AND PROSPECTS

Eugene V. Epperly
Security Specialist
Security Plans & Programs Directorate
ODUSD (Policy)

Gene received a B.A. from Notre Dame, and a Graduate Certificate from The American University (D.C.) in Technology of Management — Computer Systems. He has held positions as Special Agent/Special Agent-In-Charge, Chicago Field Office, U.S. Army Intelligence Corps, 1961–1963; Security Coordinator, IIT Research Institute, 1963–1964; and Security Specialist & Intelligence Operations Specialist (Counterintelligence), Office of Counterintelligence & Security, Defense Intelligence Agency, 1965–1974. He joined ODUSD (Policy) in 1974. Professional Organizations: ASIS, ACM; designated a "Certified Protection Professional (CPP)" by the Professional Certification Board of the American Society for Industrial Security in January 1978, recertified in January 1981.

INTRODUCTION

I'm with the Office of the Deputy Under Secretary of Defense for Policy. One of my responsibilities is the area of Defense computer security policies for the protection of classified information.

In that connection, I will talk about the new Executive Order 12356, which sets forth national security policy for classified information, discuss some of the concepts and ramifications of translating this into the automated environment and briefly cover the results of a survey of national and departmental computer security policies undertaken.

As an initial point of departure and theme, I would like to share a perspective on this whole subject that I've come to accept over time, that I think is relevant to what I'm going to cover. Simply stated, computer security is a support function supporting a support function. As such, it has little beyond pure academic relevance *unless* it is dealt with in the context of the overall organization and mission being supported. For Defense, the mission has to do with maintaining and employing armed forces to support and defend the Constitution, and to insure the security of the U.S., its possessions and areas vital to its interests by timely and effective military action. In this context, automated informations systems are *vital* to us, particularly in light of the relative numbers of military assets we have versus those available to our potential adversaries. And these systems are not going to do us much good if they're not adequately protected when needed.

The ultimate criterion for the whole computer security policy exercise, I submit, is and must be the quality and cost effectiveness of its implementation in the field, where the systems exist and are depended upon to support critical missions and functions.

EXECUTIVE ORDER 12356

Introduction (Figure 1)

This E.O. was signed this past April 2nd [1]. The order's predecessors, going back almost 30 years as shown, have addressed security for that information designated classified national security information, and they have been the primary bases upon which computer security first surfaced within Defense. Review of some of the highlights and features of the executive order should accordingly be of interest to you, especially concerning what is or is not said with regard to the automated environment.

Background and Objective

This order *establishes a uniform system* for classifying, declassifying, and safeguarding national security information. It replaces Executive Order 12065, which was issued in 1978.

The Order's objective is to improve protection for sensitive information relating to our national defense and foreign policy and to prevent excessive classification of documents. The Order will facilitate the public's access to information about the affairs of government when disclosure would not damage our national security, and it expressly prohibits the use of the system to conceal violations of law, prevent embarrassment, or delay the release of information that does not require protection. Basic scientific research information not clearly related to the national security may not be classified, nor may information that the Federal Government does not own or control.

Definitions (Figure 2)—to precisely identify the terms.
— "Information"
— "National Security"
— "National Security Information"
— "Original Classification"

Note in the last definition the Risk Analysis elements that are present. National Security Information has been categorized for decades on the basis of 1) the relative qualitative "value" of the information, in terms of the relative consequences of unauthorized disclosure; 2) the evaluative context is also specified, "National Security," meaning the national defense posture and the conduct of our foreign relations; 3) and the relative level of protection required.

Overview of Key Features (Figure 3)

The Order is divided into six parts, which contain some of the following features:

- The *three* existing *levels of classification* are retained: TOP SECRET, SECRET, and CONFIDENTIAL. TOP SECRET shall be applied to information, the unauthorized disclosure of which reasonably could be expected to cause exceptionally grave damage to the national security; SECRET, serious damage to the national security; and CONFIDENTIAL, damage to the national security. (Section 1.1)

- The Order limits the type of information that can be classified. *To classify* a document, *three requirements* must be met. *First,* the individual classifying the document must be authorized to do so. *Second,* the information must fall within one or more specified categories of information. *Third,* the classifying official must determine that unauthorized disclosure of the information could reasonably be expected to cause at least damage to the national security (the test for the CONFIDENTIAL level). (Sections 1.2; 1.3)

On the last point, *damage to the national security is presumed* in the following cases:
— Foreign government information
— The identity of a confidential foreign source
— Intelligence sources or methods (This item newly specified) (Section 1.3(c))

On the second point, *three new* areas were added to the *categories of* potentially *classifiable information:*
— "(2) the vulnerabilities or capabilities of systems, installations projects, or plans relating to the national security;"
— "(8) cryptology;"
— "(9) a confidential source;"

(Section 1.3(a))

- *Classification/Reasonable Doubt*
  — "...shall be safeguarded as if it were classified pending a determination by an original classification authority, who shall make this determination within thirty (30) days." Similarly, if there is doubt about the classification level, safeguarding shall be at the higher level pending a determination within 30 days. (Section 1.1(c))

- *Reclassification* is now possible by the President or other designated officials for information previously declassified and disclosed *if* it is determined in writing that (1) the protection is required in the interests of national security and (2) the information may reasonably be recovered. (Section 1.6(c))

- *Duration of Classification* — Original classification authorities may continue to establish specific dates or events for *declassification*. However, the Order ties the duration of classification primarily to the continued national security sensitivity of the information. Prior systems linking classification to arbitrary timeframes have not proved successful in significantly reducing the amount of classified information and have jeopardized information that merited continued protection. Under the prior system, only about 5% of the classified documents were actually marked for automatic declassification within the prescribed six-year timeframe. (Section 1.4)

- *Classification Guides* to facilitate the proper and *uniform* derivative classification of information are reemphasized, and associated requirements are added to this EO (Section 2.2). Our implementation is the *"DoD Index of Security Classification Guides,"* DoD 5200.1-I, [2] which is published semiannually. This is complemented by the *"DoD Handbook for Writing Security Classification Guidance,"* DoD 5200.1-H, October 1980 [3].

- *Oversight is critical* to the effective operation of the information security program. The Order requires heads of agencies to monitor this program closely, and to train their employees in its requirements. The Information Security Oversight Office (ISOO) will continue its role of government-wide monitoring, and will report to the President through the National Security Council. The Oversight Office will also receive and investigate any complaints brought to its attention concerning alleged abuses. (Section 5.2; 5.3)

This facet of the program cannot be overemphasized. As I'll suggest later in reviewing national and departmental policies, oversight in some effective guise is absolutely essential for program implementation in the field, where it counts, if it is to have any reasonable probability of success and uniformity. Putting out policy and then going away is simply not going to do the job in a consistent fashion—and it is this area where computer security program implementation per se has been vulnerable.

Defense, both at the OSD level and at the DoD Component level, has long had effective oversight programs in information security in the context of the executive order—a good deal more clearly needs to be done in the area of the automated environment as such, however.

- *Sanctions* — Agencies are authorized to impose penalties for knowing, willing, or negligent disclosure of properly classified information, or for knowing or willful overclassification. These sanctions apply to government employees and government contractors, licensees, and grantees. They may include reprimand, suspension without pay, removal, termination of classification authority, loss or denial of access to classified information, or others provided for by applicable law and agency regulation. (Section 5.4)

- Agencies are also required to develop special *contingency plans* "for the protection of classified information used in or near hostile or potentially hostile areas." (Section 5.3(d)). This is also new in the executive order, although our own implementing program already goes further by requiring planning for protection, removal or destruction of classified material in the case of fire, natural disaster, civil disturbance, or enemy action. Detailed procedures and responsibilities are to be established in this regard. With reference to particularly sensitive material (i.e., TOP SECRET material) where threat analysis reveals such material is not capable of protection from hostile elements in a no-notice emergency situation, then the installation of *"Anti-Compromise Emergency Destruct"* (ACED) equipment is specified. (Section 2, Chap 5, DoD 5200.1-R [4])

Identification and Marking (Figure 4)

There are several relevant changes on *security markings* here.

First, overall, the requirements have been simplified, primarily as a result of the previously mentioned changes regarding duration of classification. Also, the "stamps" for the other information accompanying classification marking can now be the same for both originally classified items and those derivatively classified, with obvious implications for automated implementation.

The second "bullet" denotes new language that specifically provides for marking of *media other than human-readable paper documents.* I note that our implementation already provides a whole section on marking of such "special categories of material," including media often associated with automated information systems; for example, punched cards, fan-fold printouts, microforms, and last but not least, "Removable Automatic Data Processing and Word Processing Storage Media" (Section 3, DoD 5200.1-R). The latter is significant to us—it requires *both* external and internal markings, with the latter "waivable" *only* in the case of existing systems where implementation cannot be effected without extensive system modifications. Even then, alternative procedures must be established to ensure that users and recipients of the media, or the information contained therein, are clearly advised of the applicable classification and the associated markings. These requirements are explicit in our Information Security Program Regulation [4].

This and the following point on "Portion marking," another feature of DoD policy that later showed up in the executive order, brings up an important exploratory effort I feel is needed. This involves the whole area of cost-effective *information tagging* at lower than the file level in automated systems. I believe this is essential as we look to the future, and such an effort can be geared not only to security classification markings at lower than file level, where lower elements have different security classifications, but also to enhance system responsiveness to users in their basic functions. For example, in the intelligence arena, tagging records or other elements with "date-of-information" and "source reliability" indicators, among others, would be very useful in significantly enhancing the utility of the products.

Further related to this theme is new verbiage providing for *marking designation standards*, including abbreviations, to be prescribed by the Information Security Oversight Office in its forthcoming directive, which will provide national-level amplification and guidance with respect to the new executive order.

The last point on *Foreign Government Information* is for the benefit of attendees from allied foreign countries. Their information shall either retain its original classification or be assigned a U.S. classification that will ensure a degree of protection at least equivalent to that required by the furnisher (e.g., UK or NATO Restricted becomes U.S. CONFIDENTIAL).

Access Restriction (Figure 5)

This area is directly relevant to computer security policy. The executive order's provisions are reworded, compared with the predecessor order, but the substance is essentially unchanged—access to classified information requires both: a security clearance determination and a need-to-know determination.

need-to-know

I've not had much problem over the years with new words being coined for old concepts, principles or techniques, but there is one exception here that I want to specifically address. That is, the use in some of the more recent technical literature of the terms, "Mandatory access control policy" and "Discretionary access control policy" to refer respectively to the identified security clearance and need-to-know processes. My concern is specifically with what the term "discretionary" could *connote.* That is, the less careful reader may infer that implementation of the requirement is "discretionary." In discussions with some of the authors, this was not their intent—they meant to denote what's shown; but let me say unequivocally that both of these are requirements, and the implementation of these two requirements is *mandatory*, regardless of the nomenclature one uses.

To reinforce this, I note that a top-to-bottom review of the *DoD Personnel Security Program* was undertaken over the past four months by a *Select Panel* chartered by my boss, the Deputy Under Secretary of Defense for Policy. Although a final course of action on proposed recommendations has not yet been resolved, a significant focus, and an anticipated area of future emphasis, can be expected to be effective, conscientious implementation of the need-to-know principle. In short, it's not going away, and as will be seen, it is one of the key distinguishing characteristics in the range of system security modes embodied in current computer security policy.

Safeguarding (Figure 6)

Finally, translating this generic charge into the ADP environment in a *reasonable, uniform*, coherent and *cost-effective* fashion continues to be our collective challenge. Note that safeguarding in this context embraces both the "life cycle" of the information and its states and stages of being variously stored, processed or otherwise handled.

APPLICATION TO THE AUTOMATED WORLD

Translation of the generic charge of the executive order into valid security standards and criteria for the automated environment is of course no mean task—it has occupied a lot of folks for a long time, beginning in our environment with the Defense Science Board Computer Security Task Force study effort from 1967 to 1970 [5].

There are essentially three general concepts embodied in our computer security policy that approach that task by taking due cognizance of the realities of automated systems and attendant security problems: these are the overall objectives of the policy as set forth in the DoD Directive 5200.28 [6], the nature of the system evaluation and approval process, and the concept of ADP system security modes of operation. As an aside, I must note that 5200.28 must be considered, in a total system sense, the automated supplement to the basic implementation of the executive order, DoD Regulation 5200.1-R [4]. A complete program needs both.

Policy Objective

As a point of departure here is the collective end objective (Figure 7). The ADP system's collective security measures *must,* with reasonable dependability, prevent both:

1. access to classified material by unauthorized persons, *and*
2. unauthorized manipulation of the ADP system.

Although we are protecting information, in this arena the ADP system as such must be protected. The "why" of it has been long known—currently available systems are penetrable and complex. Most significantly, penetration need not be executed at the time unauthorized access to classified information is effected. Rather, a penetration may be effected at one time and remain undetected for long periods of time prior to exploitation (e.g., [7, 8 and 9]).

Basic Security Philosophy; Evaluation and Approval Process

In terms of security concepts, we do not view computer security as fundamentally different from the protection of other information and material. We do not orient on 100% security as feasible in this area—even approaching that level is usually prohibitive in terms of cost or constraint. Our approach is to relatively secure by employing security barriers and measures in complementary combination (i.e., systematized "defense in-depth") so that the cost/risk of penetration exceeds the value or payoff of the penetration object, be it personal or classified information, nuclear material or monetary assets. This "work factor" approach involves identifying vulnerabilities (paths into the "system") and erecting barriers generating a "work factor," in terms of cost/risk, which exceeds the worth of the object(s) to be protected.

The end objective is an "acceptable level of risk determination"—the professional security judgment that the security subsystem generates such a cost/risk work factor in a comprehensive, systematic and cost-effective way. We feel the process through which this determination is most effectively and validly made is the security analysis, test and evaluation process (Figure 8), wherein both vulnerabilities and countermeasures are systematically considered.

The computer security policy problem here is (Figure 9), there are no generally accepted standards, criteria or even valid guidelines for hardware/software security, yet this overall process is *the* basic tenet of our policy. By contrast there are *relatively* clearcut guidelines and minimum requirements in all the other security areas indicated. The end result is that the process cannot now be executed with sufficient confidence in terms of validity or reliability, let alone cost effectiveness.

It is *precisely this problem* to which I see the Computer Security Initiative in general and the DoD Computer Security Evaluation Center in particular responding. Let me outline the policy framework which I feel can effectively accommodate the Initiative Program's technical evaluation concepts as they are evolving—as will be briefed during this seminar.

ADP System Security Modes (Figure 10)

In seeking to accommodate the hardware/software security problem with the need to operate, the need to employ ADP systems to accomplish or support a multitude of defense missions, a set of alternatives evolved which may be viewed simply as alternative paths that involve the sorts of tradeoffs I mentioned at the

beginning (Figure 11). Although not stated as such on the slides, one key variable, in the terms of this seminar, is the relative degree of "trustedness" insofar as the hardware/software security component is concerned. The modes involve basic tradeoffs between conventional security measures on one hand and hardware/software measures on the other. Viewed as alternatives along a continuum, as one moves from left to right, relative hardware/software security responsibility increases, along with relatively increasing risk and uncertainty. In parallel, relative degree of security cost and constraint tends to decrease. The selection of one of these modes for a system is, of course, largely dependent upon the specific system, its functional requirements, its users and its environment, as to which mode is the most cost-beneficial.

As a further specification (Figure 12), let me relate these modes to the two policy requirements for access to classified material. Before an individual may be granted access to classified information: 1. he must have been granted a security clearance; and, 2. his access must be necessary for the performance of his official duties (i.e., he must have a "need-to-know"). In the manual world, both clearance and need-to-know determinations are normally made by humans in a fairly straightforward way. In the automated environment, however, this can vary. Moving again from left to right, clearance and need-to-know are determined prior to system access in the Dedicated Mode. In the System High Mode, clearance is determined before access, but need-to-know is not. The double line indicates a significant change in hardware/software security role—to the right of the lines, it becomes one of preventing outright security violations and compromises. Now let's look at some specifics.

The Dedicated Mode (Figure 13), at the far left, is the most clearly approvable type of system simply because the key security functions I noted are formed by comfortable, well-understood conventional security measures. By definition, everyone with access to such a system has a clearance and a need-to-know for everything then in the system. The major protection burden is assumed by conventional personnel and physical security measures and techniques which isolate the system from unauthorized personnel, pursuant to fairly clear policy requirements. Hardware/software security role is minimized as a result.

The Full Multi-Level Security Mode (Figure 14), is at the other extreme. There are some system users who have *neither* clearance nor need-to-know for material contained in the system at the time of their access. In this case, in direct contrast to the Dedicated Mode, *both* clearance and need-to-know are determined by the ADP system. The *separation* of users, their programs and files *must* be maintained by hardware/software security mechanisms under operating system control, because it's all in the computer and potentially accessible at the same time. In terms of tradeoffs, the direct security costs and associated constraints on system utilization are minimized (e.g., not all users with concurrent access need be cleared to the highest levels; remote terminal areas need not meet the physical security requirements of the central computer facility; cpu (central processing unit) time and system availability are not lost through sanitization procedures, and so on). But at the same time, hardware/software security responsibilities are now maximized. The major burden of key security functions falls upon hardware/software.

The *"System High Mode"* (Figure 15). The basic distinction between Dedicated and System High is the matter of need-to-know. In both cases, all users are cleared to the highest level. In the Dedicated Mode, need-to-know is determined before actual system access is afforded to users; in the System High Mode, it is determined by the ADP system during access. It is established and maintained by hardware/software. This mode is a more flexible, less constraining mode of operating an ADP system than the Dedicated Mode. But, election of this mode requires the development and implementation of hardware/software mechanisms to implement need-to-know.

The Controlled Mode (Figure 16) moves one step further along the continuum and crosses that significant double line. Neither individual clearance nor individual need-to-know is predetermined. But, in contrast to the Multi-Level Security Mode, the important difference is a set of explicit measures to reduce risk and vulnerability and to directly enhance or even bypass hardware/software security measures under operating system control.

Basically, the objective here is to provide a potentially approvable, interim alternative to the more restrictive Dedicated and System High Modes—a transition. But, *one must take explicit steps,* vice the Multi-Level Mode, to reduce relative risk and vulnerability, and, preferably in combination, other steps to augment the system hardware/software security posture. Examples of risk reduction are limits on the range of clearance levels of users who have concurrent access (e.g., users of only two or three clearance levels).

Actions that can concurrently reduce relative vulnerability include restrictions on users capabilities, such as providing only query and response capability.

Application to Initiative Program Concepts and Technical Criteria

I think clearly the most important aspect of the foregoing is the rather clear potential linkage between modes, as a continuum of systems on the basis of relative required "trustedness," and efforts of this Computer Security Initiative Program, dealing with development of "trusted" ADP systems.

As this obsolescent slide shows (Figure 17), there is a clear correlation between the relative levels of protection that are evolving for purposes of evaluation and the continuum of modes. The left-hand column has changed and will be treated later in specifics [10]—the general point I want to make is that there is a clear potential relationship between the Initiative Program's efforts and the provisions of existing policy. Application of those efforts to real world ADP systems through existing policy is therefore neither remote nor obscure.

The notion here is that one might tentatively select a target system security mode on the basis of inherent security capabilities in a system during the initial stages of the risk assessment (Figure 18). There would follow detailed identification and assessment of a host of variables, both technical and non-technical, peculiar to the individual system, any of which, in a tradeoff context, might change the relative security posture of the system "up" or "down" in the right-hand column—that is, with regard to the system security mode ultimately proposed for formal approval by the Designated Approving Authority. Jack Adams has developed a framework for enumerating critical security considerations that can be applied to the middle "interface" column [11].

The significance of this linkage is now limited to those systems processing classified information in DoD and in industry; as I'll suggest in a moment, that significance may be much more profound, depending upon the policy framework that ultimately evolves with regard to the computer security requirements of Transmittal Memorandum No. 1 to OMB Circular A-71 [12].

From the policy interaction, let me turn briefly to the procedural—how the expertise being developed within the Initiative Program might interface with the folks in the field who are currently tasked, and have been for some time, with evaluating and approving real world ADP systems.

As a point of departure, let me again refer to the general process that is a fundamental tenet of our policy (Figure 19). Recall that other than the "hardware/software" area indicated, criteria and requirements are relatively clear. Also, given both resource limitations and the highly technical nature of the task, it appears most likely that formalized establishment of the Initiative Program's expertise will be at least initially centralized.

The technical expertise can be integrated into the test and evaluation process as shown here, by complementing the ongoing Component activities in the technical area. Recall that our policy explicitly delegates ADP system security approval authority to the DoD Components (and DIS for contractor ADP systems). It is not our intention to change that—the final approval must be on a system-by-system basis—that is, keyed to an individual system with its unique environment and functional requirements.

Though this is an old slide (Figure 20), it shows the place of technical advice, indicated in red, in the overall Component evaluation process. It also indicates our intent that the overall synthesis of the diverse parts of the analysis, together with the final decision to approve or not approve, lies with the appropriate Component Designated Approving Authority.

Hardware/Software Security

What we have in DoD 5200.28-M [13] in this area was essentially predicated upon the existence of a relatively secure hardware/software security foundation, a presumption that has been shown through subsequent empirical testing to be at best dubious...

It is this specific area, this missing key piece of the overall puzzle, where we look to the consortium in general and its institutionalization in the DoD *Computer Security Evaluation Center* at NSA to respond with both *evaluated products and* cost-effective *technical standards,* criteria and guidance. Without such, policy implementation in the field will primarily follow those modes where the hardware/software part of the

equation is minimized, a phenomenon clearly demonstrated by our DoD-wide survey of classified systems several years ago, and an approach generally resulting in both high cost and substantial constraint on system utilization.

SURVEY OF FEDERAL COMPUTER SECURITY POLICIES

I'd like to briefly cover this topic to indicate to you what exists on the subject and to relate the findings to my initial theme on policy implementation in the field. I will cover this rather quickly, noting that the complete report is available through the Defense Technical Information Center [14].

Purpose

The purpose of this report was to document the survey of identified national level and Executive Branch department and agency computer security policies as undertaken by the Policy Survey Subcommittee.

Tasking. The subcommittee was asked to review current government computer security policies at both the national and department/agency levels. The purpose of the review was to identify what policy exists, what it addresses, and what responsibilities are assigned. The task, approach and objectives as refined by the subcommittee are summarized in Figure 21.

Approach and Methodology

Target Universe. In view of time and resource limitations, it was decided to limit the survey of Executive Branch departments and agencies and to concurrently maximize survey coverage by focusing on those entities operating the overwhelming preponderance of government ADP systems, as reflected in the GSA *Automatic Data Processing Equipment Inventory in the United States Government.*

We specified a "sample universe" of fifteen departments and agencies (Figure 22), thereby covering 8237 ADP systems in the GSA inventory, or over 88.6% thereof, not including CIA or NSA ADP systems, and including seven of the twelve Cabinet-level departments.

Survey Focus. Given the task of surveying computer security policies, the subcommittee focused on computer security documents as such. Rather than include all policy documents mentioning computer security, it was agreed that documents to be reviewed for this survey must meet the following criteria:

1. They must be authoritative and directive in nature;
2. They must reflect in content the multi-disciplinary, total systems approach axiomatic in current computer security policy (Figure 23).

Total coverage of Executive Branch agencies and departments (over 70) was deemed impractical—the effort focused on the fifteen agencies that represented over 88% of the government ADP Systems reflected in the GSA inventory and included the majority of Cabinet-level departments.

A questionnaire format was developed to extract on a common basis key attributes of document policy coverage, and this was to be completed by subcommittee members in the interests of reliability and consistency. A key objective of the process was to identify national level policies and authorities. Existence of policy/program oversight mechanisms was identified as a secondary but very important focus.

Department/Agency Policies (Figures 24–26)

For the fifteen agencies surveyed, 32 separate computer security policy documents (totalling 1,316 pages) were obtained and reviewed. These were consolidated into 27 policy sets of like scope and applicability. All fifteen agencies have promulgated computer security policies; however, these varied in approach, scope and applicability. Survey results reflected the historical sequence of attention to computer security; 63% of the sets reflected policies implementing national security information protection requirements. Other frequencies cited among the 27 policy sets were: Privacy Act, 41%; Transmittal Memorandum No. 1 to OMB Circular A–71, 30%; Intelligence Special Access Programs, 30%; National COMSEC Directive, 15%; OMB Circular A–108, 11%; and, Atomic Energy Act, 7%. Computer security subdisciplines, frequencies were reflected in the sets as follows: Physical security, 100%; personnal security, 96%; administrative/procedural security, 96%; hardware/software security, 96%; communications security, 89%; and, emanations security, 70%.

"National" Level Policies (Figures 27–29)

A most important facet of the survey was to identify higher level authoritative bases for computer security policies at the department/agency level. Thirteen documents forming 5 policy sets were identified and reviewed. As an operational complement to policy, various program oversight mechanisms were also identified, to include the Legislative Branch.

Comprehensive computer security policy, promulgated by the Office of Management and Budget* and supplemented by further issuances from the Office of Personnel Management (OPM), the General Services Administration (GSA), and the National Bureau of Standards, Department of Commerce (NBS), was revealed. This policy included:

— All federal data and applications processed by computer systems
— Personal, proprietary, and other sensitive data, to include national security data.
— Such data and applications processed by other systems on behalf of federal departments and agencies, as well as by federal computer systems as such.

Supplementing policies in response to OMB tasking included the following:

— OPM has amended the *Federal Personnel Manual*
— GSA has amended the *Federal Property Management Regulations* and the *Federal Procurement Regulations*
— NBS has issued numerous guideline publications and maintains an ongoing program for standards development.

Other national level policies of narrower scope and applicability included implementation of classified information safeguarding requirements (e.g., NATO, Intelligence, and Atomic Energy-related information) and of requirements for personal information subject to the Privacy Act.

A Federal Agency Perspective (Figures 30 & 31)

A section describing the context and flow of computer security policies from higher levels is included to illustrate, in an agency organizational context, policy and oversight approaches taken and possible problems with regard to effective implementation of current and future computer security policy requirements.

A point of the example is to illustrate the manner in which computer security policies and associated requirements converge on an Executive Branch organization and a fashion in which they can be integrated (or *not* be integrated). The overall situation is one which carries the *potential* for the generation of *confusion*, unwarranted *duplication of effort*, and *policy conflict*. The duplication concern is particularly critical inasmuch as computer security is a relatively new area requiring attention, to include resources. And existing resources appear to be quite limited, particularly in the face of the dramatic expansion of requirements represented by the scope of the recently promulgated OMB requirements.

The *general point*, beyond the Defense example, is that *explicit attention* must be given to the impact at the department/agency level of higher level policy actions, particularly the derivative and *cascading effects of any policy confusion, conflict, inconsistencies* and ambiguities from the top down to the bottom line—the ultimate implementation of policies in field data processing installations.

SUMMARY

The survey clearly reflected (Figure 32):

• *Omnibus Policy.* In place, comprehensive computer security policy promulgated by the Office of Management and Budget, Executive Office of the President.

• Pursuant to OMB central agency tasking under this program policy:

— OPM has issued personnel security requirements and guidelines now in the *Federal Personnel Manual* [15,16];

— GSA has amended the *Federal Property Management Regulations* (FPMR Amendment F–42) to add a new section for the protection of ADP and telecommunications systems and a subpart to provide guidelines on environmental and physical security of ADP facilities [17];

— GSA has amended the *Federal Procurement Regulations* (FPR Amendment 210) to require that agencies' computer security requirements be included and certified in agency procurement requests and that acquisition

*Transmittal Memorandum No. 1 to OMB Circular A–71, [12].

specifications include certified government computer security requirements in connection with solicitations, contracts, and contract administration [18]; and,

— National Bureau of Standards, Department of Commerce, has issued numerous information and guidance publications on computer security [19] as well as maintaining an ongoing program for standards development.

• *Other Policies.* There are also documented a number of other, earlier Executive Branch-level computer security policies of narrower scope and applicability, including:

— Department/agency-generated policies in implementation of generic classified information safeguarding requirements imposed by Executive Order 12065 (which preceded EO 12356 [1]).

— Special Access Program classified information, such as:
  - NATO information
  - Intelligence information
  - Restricted Data and associated information

— Policies associated with implementation of the Privacy Act of 1974 in the ADP arena and OMB Circular A-108.

The interrelationships of these policies are suggested by the diagrams at Figures 32 and 33. Figure 32 shows these as separately promulgated from the national level; Figure 33 relates them in a Venn context wherein the OMB policy includes all Federal data/applications processed by computers.

Oversight Results (Figure 34)

Audits and associated reviews found significant problems with the field implementation of computer security programs.

Most recent (at the time) was the January 1979 GAO report which concluded that "programs fell short of being *comprehensive* and *top management support* was lacking." [20] (emphasis added). The report noted that the review was completed prior to the issuance of TM 1 to OMB Circular A-71, but noted that the document ". . . requires action by top agency managers which could contribute greatly to correcting many of the computer data security problems addressed in the GAO report." Further, ". . . it (TM 1 to A-71) sets an appropriate framework for agencies initiatives to correct their data security problems."*

Conclusions (Figure 35)

The Subcommittee considered the situation to suffer significantly from fragmentation across-the-board and from the lack of cost-effective, feasible implementing guidance. The former, particularly, is manifest in the example of national policy flow and impacts at the department/agency level. This suggests a clear need for further efforts to effectively integrate overall computer security policies in a context that specifically considers the flow of data/applications to be protected: 1. between and among federal agencies; and, 2. between federal agencies and private sector contractors as well as the potential counterproductive effects cited above.

The foregoing, in turn, indicates that a deeper level of analysis is required to focus on those aspects of computer security field implementation that are susceptible to benefit from national level attention and effort.

Accordingly, the Subcommittee strongly and unanimously recommended attention be given to the following specific problem areas related to current computer security policies and field implementation thereof:

1. The GAO identified lack of *top management support* in federal departments and agencies to specifically include the need for the education and awareness of top management;

2. Closely interrelated, the lack of *resources*, both research and development resources and operational resources, with specific attention to the problem of *trained manpower* and *funding stability.*

3. The problematic nature of the *hardware/software computer security subdiscipline*, to specifically include the development of secure systems technology, security technical evaluation methodologies, and recommended management and operational mechanism(s) therefor;

*GAO has since issued a new report, essentially a follow-on to the cited 1979 report, with special emphasis on implementation of the OMB policies [21]. It is entitled, "*Federal Information Systems Remain Highly Vulnerable* to Fraudulent, Wasteful, Abusive, and Illegal Practices" (emphasis added).

4. Manifest requirements for means of *more effective integration and coordination* of identified national policy promulgating activities; and,

5. Generation of feasible and cost-effective *implementing* guidance for various *computer security subdisciplines* associated with the implementation of overall computer security policies.

DOD COMPUTER SECURITY POLICY PROSPECTS—CHALLENGES

Fundamental Modifications

I don't see a fundamental modification of the current DoD policy documentation until such time as there is substantive progress in the hardware/software security area, particularly the matter of the linkages I mentioned earlier between the emerging technical criteria and categories on one hand and the security modes on the other. Moreover, this will undoubtedly be evolutionary in the direction of greater specificity and precision.

Nearer-Term Actions

— I expect a revision of the basic directive upon approval of the Computer Security Evaluation Center's charter. Among other things, this revision will reflect the existence of the Center and contain pointers to the Center and its anticipated products insofar as policy implementation is concerned, in much the same fashion as the proposed Evaluation Center's charter contains pointers to DoD computer security policy.

— Updating and amplification for various sections of the manual are long overdue and will be circulated as soon as possible. I'll speak to specifics in a moment.

— Transmittal Memorandum No. 1 to OMB Circular A-71 presents a substantial challenge. It will have to be dealt with in a collaborative fashion, and it's not the responsibility of my office as such. As suggested, however, if we're going to help the field implement it, we have got to integrate our diverse computer security policy requirements in some effective fashion, at the DoD level if not at the national level. In an earlier presentation in this forum, I outlined how we thought this might be done [22]; however, this has been moving very slowly.

— I do have a near-term initiative to briefly describe (Figure 36), already approved in essence for the Defense Industrial Security Program [23]. It is now to be proposed for Defense Components as well. I believe it's timely, reasonable and responsive to current needs. The subject is word processing and related computer-driven information systems, such as electronic mail. The concern driving this initiative stems from our awareness of plans in DoD to implement large-scale, shared word processing networks where in many cases the only security concern being given is to TEMPEST approval for specific equipment components. Our view is that such networks require the same comprehensive, multidisciplinary and systematic approach to security throughout the life cycle as is required for similarly sized, shared ADP systems.

The basic proposal would include such systems within the policy framework of DoD Directive 5200.28 and its manual, to apply such requirements as comprehensive security evaluation and approval and the designation of a specific individual responsible for system security. Concurrently provided would be simplified requirements for stand-alone, single user systems, where protection requirements would approach those of an electronic memory typewriter. The obvious overall intent is to resolve this new and growing security problem within an existing program as the most cost effective approach.

Accompanying the foregoing, I am proposing 3 sections of the manual to be authorized for immediate use, pending final, formal amendment to the manual. These are:

1) An addition to the manual's section on personnel security, focusing on two personnel security requirements necessary for an effective program. First, the requirement for initial and continuing security briefings and indoctrination to system support and user personnel. Secondly, emphasis on supervisor requirements for personnel monitorship in terms of continued eligibility for access to classified information. Both of these are primarily restatements of, and pointers to, relevant requirements set forth in the information security and personnel security regulations [4 & 24].

2) Guidelines and procedures on the adjustment of area controls, primarily to facilitate implementation of the proposed policy in this area.

3) With similar intent, a rewrite and expansion of authorized procedures for media clearance and declassification, applicable to both shared and stand alone systems.

CONCLUSIONS

I realize I've covered a lot of diverse areas in these minutes, and I will try to briefly capsulize.

Technical Support to Policy

On this I will sum up what we on the policy-oriented side of the picture perceive: the primary barrier to both more valid and responsive policy per se, and more importantly, to cost-effective implementation of that policy in the field, remains the relative status of the hardware/software security subdiscipline—the lack of relatively secure system foundations in the first place and the associated lack of standards, criteria and guidelines for both how to get there and how to determine when you have arrived.

To this long-standing need (explicit in the Defense Science Board report in 1970 and the consequent provision in DoD 5200.28 in 1972), the establishment of the DoD *Computer Security Evaluation Center* is considered *directly responsive.* Upon the generation of these products from the Center, we will need close interaction and active collaboration to flesh out the needed *linkages between* the *technical criteria* and related categories that are emerging and the *policy modes,* to permit as easily and simply as possible their direct utilization in the field. This also means due cognizance must continue to be provided for the fact that ultimate system security approval authority properly resides in the Components and specifically involves more than the hardware/software facet exclusively. There are a host of technical and nontechnical variables that are system- and installation-dependent and that have direct impact on the total security posture of a given system. Nonetheless, in view of the exploratory work that's already begun to approach this bridging, it will be an extremely difficult and intellectually challenging task.

Finally, this effort in general, and the Center in particular, should be "user friendly," and ultimately, it *has* to be "user responsive."

Policy at all levels, from the very top to that lowest echelon just above our field computer systems, I feel should have the following attributes:

— SIMPLICITY, especially geared to the person in the field who has to implement our requirements. In most cases, he or she will be a part-timer, one who simply will not be able or inclined to execute time-consuming and complex exercises on behalf of this function. I speak here especially to such notions as risk analysis methodologies—I have come to feel that we must, to the maximum extent that such is practical and valid, generate simplification to such processes, analogous to the type of practical simplification that evolved in the whole area of information security over many decades. If we don't, then we run the real-world risk that nothing effective will happen, and the function is clearly too important to encourage that outcome.

— UNIFORMITY, especially in a relatively new and unprecedented area such as this, is a meaningful goal from an overall organizational perspective, particularly where resource utilization is a factor. Blatant over-protection here and under-protection there is not responsive to the total organizational requirements for comprehensive systems protection.

This need is furthermore reinforced by the organizational facts of life in the Department of Defense. We have on one hand relatively autonomous DoD Components. However, ultimately many operations, especially those keyed to our main reason for being, the field fighting forces, are joint operations. This means automated interfaces between and among the Military Departments, Defense Agencies and the field operating forces, the Unified and Specified Commands and their subordinate commands. When you expand consideration to include other federal agencies and international organs such as NATO, this need magnifies substantially.

— COHERENCY AND CONSISTENCY. These last attributes are pointing both at the Defense and national levels, with special reference to the results of our policy survey. All of those diverse and varied computer security and related policy directives come together both at the federal department/agency level and at the field data processing installation level. The degree to which we can effectively integate these into a coherent and consistent package before we send it out to the field, I believe, will reflect the degree to which we can directly enhance the possibility for cost effective implementation where it really counts.

REFERENCES

1. "National Security Information," Executive Order 12356, April 2, 1982; *The Federal Register*, Vol. 47, No. 66; Tuesday, April 6, 1982.

2. "Index of Security Classification Guides," DoD 5200.1-I, July 1982 (published semiannually).
3. "DoD Handbook for Writing Security Classification Guidance," DoD 5200.1-H, October 1980.
4. "Information Security Program Regulation," Department of Defense Regulation DoD 5200.1-R, August 1982.
5. *Security Controls for Computer Systems: Report of Defense Science Board Task Force on Computer Security*, published by the Rand Corporation for the Office of The Director of Defense Research and Engineering (Rand Report #R-609), February 11, 1970.
6. "Security Requirements for Automatic Data Processing (ADP) Systems," Department of Defense Directive 5200.28, December 18, 1972, as amended (Change 2, April 29, 1978).
7. Branstad, D. K., "Privacy and Protection in Operating Systems," *Operating Systems Review*, VII, 1 (January 1973).
8. Stryker, D. J., "Subversion of a 'Secure' Operating System," Naval Research Laboratory, Washington, D.C. 20375, NRL Memorandum Report 282 (June 1974).
9. Abbott, R. P., et al., "Security Analysis and Enhancement of Computer Operating Systems," National Bureau of Standards, Washington, D.C. 20234, Report NBSIR 76-1041 (April 1976).
10. "Trusted Computer System Evaluation Criteria," (draft), DoD Computer Security Center, May 24, 1982.
11. Adams, J. A., "Computer Security Environmental Considerations," Federal Systems Division, International Business Machines Corporation, Arlington, Va. 22209, (Final Draft) August 15, 1979.
12. "Security of Federal Automated Information Systems," Transmittal Memorandum No. 1 to OMB Circular No. A-71, Office of Management and Budget, Executive Office of the President, Washington, D.C. 20503, July 27, 1978.
13. "ADP Security Manual: Techniques and Procedures for Implementing, Deactivating, Testing and Evaluating Secure Resource-Sharing ADP Systems," Department of Defense Manual DoD 5200.28-M, January 1973, as amended (Change 1, June 25, 1979).
14. Epperly, E. V., et al., "Survey of Federal Computer Security Policies," Report of the Policy Survey Subcommittee, November 1980, Defense Technical Information Center (DTIC) AD# A103676.
15. "Personnel Security Program for Positions Associated with Federal Computer Systems," FPM (Federal Personnel Manual) Letter 732-7, Office of Personnel Management, Washington, D.C. 20415, November 14, 1978 (Subsequently incorporated in the Federal Personnel Manual as Section 9, Subchapter 1, Chapter 732).
16. "Authorities and Guidelines for Investigations of Persons Having Access to Federal Computer Systems, and Information in Those Systems," Federal Personnel Manual Bulletin 732-2, Office of Personnel Management, Washington, D.C. 20451, January 11, 1980.
17. "Security of Federal ADP and Telecommunications Systems," Federal Property Management Regulations Amendment F-42, 41 CFR CH. 101, *The Federal Register*, August 11, 1980.
18. "Special Types and Methods of Procurement; Automatic Data Processing Contracting," Federal Procurement Regulations Amendment 210, 41 CFR Part 1-4, *The Federal Register*, October 6, 1980.
19. *Computer Security Publications*, NBS Publications List 91, National Bureau of Standards, U.S. Department of Commerce, revised April 1981.
20. "Automated Systems Security—Federal Agencies Should Strengthen Safeguards Over Personal and Other Sensitive Data," U.S. General Accounting Office, Washington, D.C., Report LCD-78-123 (January 23, 1979).
21. "Federal Information Systems Remain Highly Vulnerable to Fraudulent, Wasteful, Abusive, and Illegal Practices," U.S. General Accounting Office, Washington, D.C. Report MASAD-82-18 (April 21, 1982).
22. Epperly, E. V., "The Department of Defense Computer Security Policies," Proceedings of the Second Seminar on the DoD Computer Security Initiative Program, January 15-17, 1980 (DTIC AD# A101997).
23. "Security Requirements for ADP Systems," Section XIII, "Industrial Security Manual for Safeguarding Classified Information," Department of Defense Manual DoD 5220.22-M, July 1981.
24. "Personnel Security Program," DoD Regulation 5200.2-R, December 1979. □

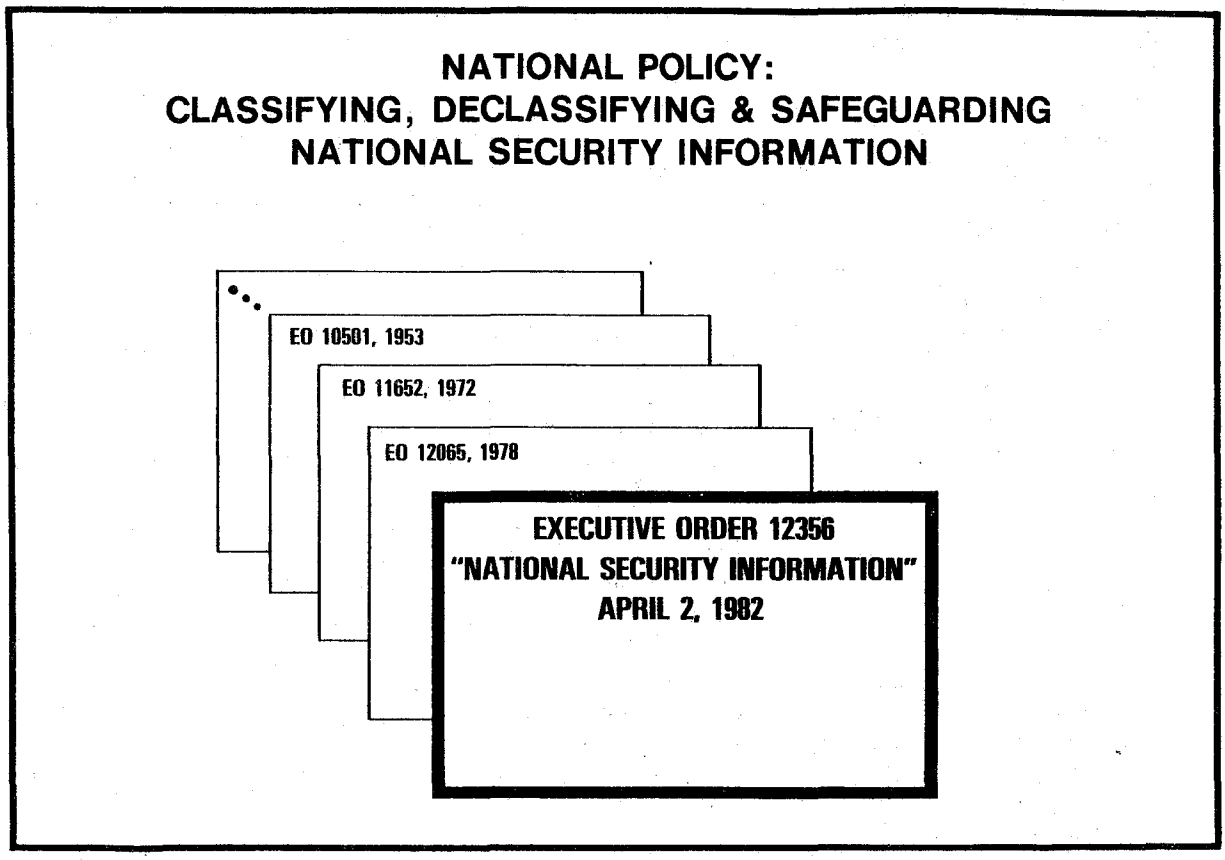

NATIONAL POLICY:
CLASSIFYING, DECLASSIFYING & SAFEGUARDING
NATIONAL SECURITY INFORMATION
EO 10501, 1953
EO 11652, 1972
EO 12065, 1978
EXECUTIVE ORDER 12356
"NATIONAL SECURITY INFORMATION"
APRIL 2, 1982

Figure 1

DEFINITIONS

INFORMATION —	''MEANS ANY INFORMATION OR MATERIAL, REGARDLESS OF ITS PHYSICAL FORM OR CHARACTERISTICS, THAT IS OWNED BY, PRODUCED BY OR FOR, OR IS UNDER THE CONTROL OF THE U.S. GOVERNMENT''
NATIONAL SECURITY —	''MEANS THE NATIONAL DEFENSE OR FOREIGN RELATIONS OF THE U.S.''
NATIONAL SECURITY INFORMATION —	''MEANS INFORMATION THAT HAS BEEN DETERMINED PURSUANT TO THIS ORDER OR ANY PREDECESSOR TO REQUIRE PROTECTION AGAINST UNAUTHORIZED DISCLOSURE AND THAT IS SO DESIGNATED.''
ORIGINAL CLASSIFICATION —	''MEANS AN INITIAL DETERMINATION THAT INFORMATION REQUIRES, IN THE INTEREST OF NATIONAL SECURITY, PROTECTION AGAINST UNAUTHORIZED DISCLOSURE, TOGETHER WITH A CLASSIFICATION DESIGNATION SIGNIFYING THE LEVEL OF PROTECTION REQUIRED.''

Figure 2

E.O. 12356 — KEY FEATURES

- RETAINS 3 CLASSIFICATION LEVELS
  - — "TOP SECRET"
  - — "SECRET"
  - — "CONFIDENTIAL"
- CLASSIFICATION REQUIRES:
  - — AUTHORIZED CLASSIFIER
  - — INFORMATION IN SPECIFIED CATEGORIES (3 ADDED)
  - — UNAUTHORIZED DISCLOSURE — DAMAGE TO THE NATIONAL SECURITY
    (DAMAGE PRESUMED: FOREIGN GOV'T INFO; IDENTITY, CONFIDENTIAL FOREIGN SOURCE; INTEL. SOURCES OR METHODS)
- REASONABLE DOUBT
  - — SAFEGUARD
  - — 30 DAY DETERMINATION
- RECLASSIFICATION
- DURATION OF CLASSIFICATION
  - — CONTINUED NATIONAL SECURITY SENSITIVITY
  - — ESTABLISHED DATES OR EVENTS
- CLASSIFICATION GUIDES — EMPHASIZED & EXPANDED
- PROGRAM OVERSIGHT — NEW REQUIREMENTS FOR ACTIVE OVERSIGHT & SECURITY EDUCATION
- AUTHORIZED SANCTIONS
  - — KNOWING, WILLFUL OR NEGLIGENT DISCLOSURE OF PROPERLY CLASSIFIED INFORMATION, OR
  - — KNOWING OR WILLFUL OVERCLASSIFICATION
  - — GOVERNMENT EMPLOYEES & GOVERNMENT CONTRACTORS, LICENSEES & GRANTEES
- CONTINGENCY PLANS

Figure 3

IDENTIFICATION & MARKING

AT TIME OF ORIGINAL CLASSIFICATION, FOLLOWING INFORMATION SHALL:

- BE SHOWN ON THE FACE OF ALL CLASSIFIED DOCUMENTS

OR

- CLEARLY ASSOCIATED WITH OTHER FORMS OF CLASSIFIED INFORMATION IN A MANNER APPROPRIATE TO THE MEDIUM INVOLVED

1. ONE OF THREE CLASSIFICATION LEVELS
2. IDENTITY OF ORIGINAL CLASSIFICATION AUTHORITY
3. AGENCY & OFFICE OF ORIGIN
4. DATE OR EVENT FOR DECLASSIFICATION OR, ''ORIGINATING AGENCY'S DETERMINATION REQUIRED''

PORTION MARKING

STANDARD DESIGNATIONS, INCLUDING ABBREVIATIONS

FOREIGN GOVERNMENT INFORMATION

Figure 4

ACCESS POLICY — RESTRICTIONS

''A PERSON IS ELIGIBLE FOR ACCESS TO CLASSIFIED INFORMATION

- PROVIDED THAT A DETERMINATION OF TRUSTWORTHINESS HAS BEEN MADE . . . AND

  (FAVORABLE PERSONNEL SECURITY CLEARANCE DETERMINATION)

- PROVIDED THAT SUCH ACCESS IS ESSENTIAL TO THE ACCOMPLISHMENT OF LAWFUL AND AUTHORIZED GOVERNMENT PURPOSE''

  (POSITIVE ''NEED-TO-KNOW'' DETERMINATION)

Figure 5

OUR CHALLENGE
— SAFEGUARDING —

"CONTROLS SHALL BE ESTABLISHED . . . TO ENSURE THAT CLASSIFIED INFORMATION

IS USED,
PROCESSED,
STORED,
REPRODUCED,
TRANSMITTED &
DESTROYED

ONLY UNDER CONDITIONS THAT WILL

- PROVIDE ADEQUATE PROTECTION AND
- PREVENT ACCESS BY UNAUTHORIZED PERSONS"

Figure 6

THE BASIC ADP SYSTEM RELIABILITY AND INTEGRITY FEATURES MUST BE AUGMENTED TO ASSURE THAT SYSTEMS WHICH PROCESS, STORE, OR USE CLASSIFIED DATA AND PRODUCE CLASSIFIED INFORMATION WILL, WITH REASONABLE DEPENDABILITY, PREVENT:

A. DELIBERATE OR INADVERTENT ACCESS TO CLASSIFIED MATERIAL BY UNAUTHORIZED PERSONS, AND

B. UNAUTHORIZED MANIPULATION OF THE COMPUTER AND ITS ASSOCIATED PERIPHERAL DEVICES.

Figure 7

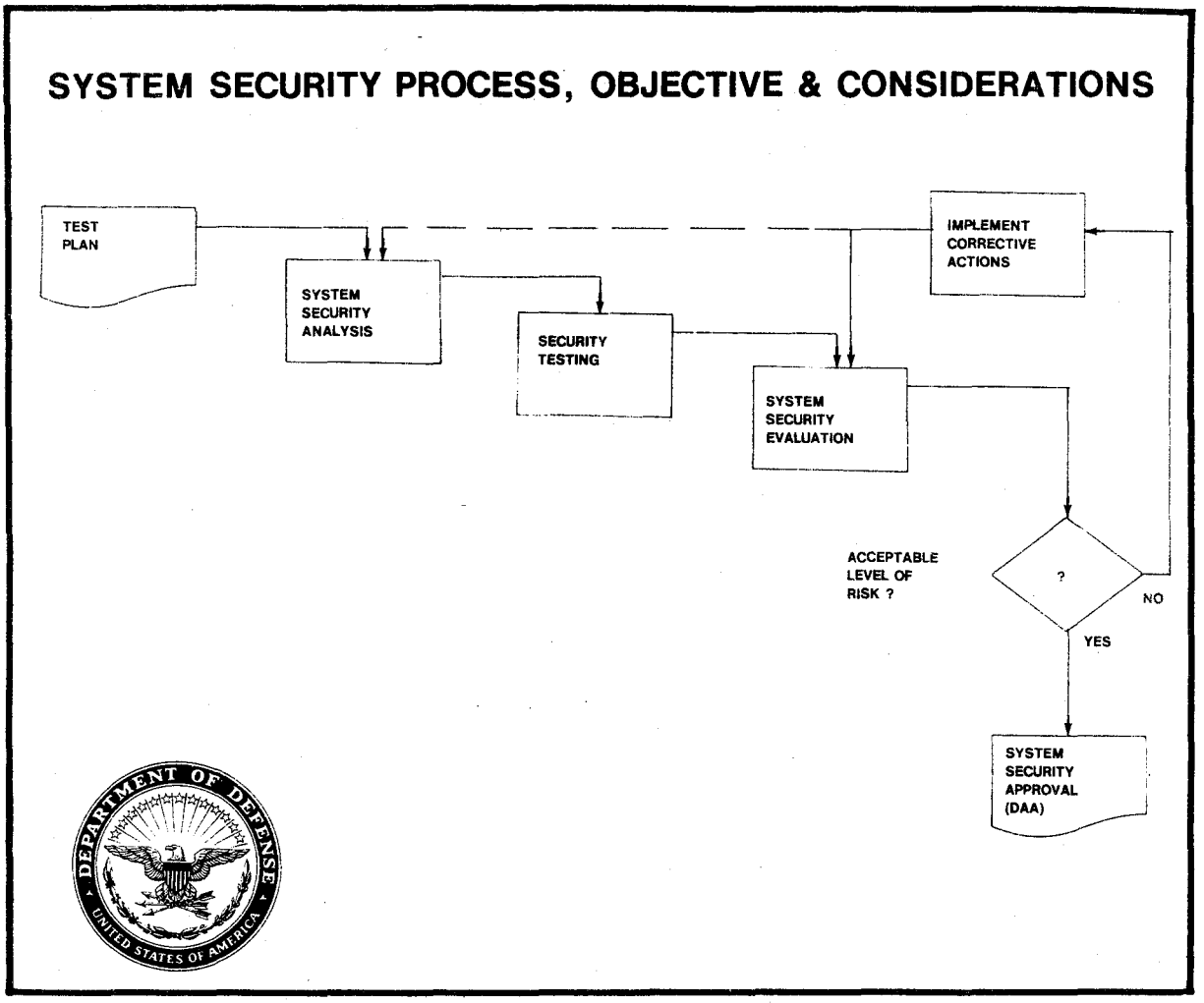

SYSTEM SECURITY PROCESS, OBJECTIVE & CONSIDERATIONS
TEST PLAN
SYSTEM SECURITY ANALYSIS
SECURITY TESTING
SYSTEM SECURITY EVALUATION
IMPLEMENT CORRECTIVE ACTIONS
ACCEPTABLE LEVEL OF RISK ?
?
NO
YES
SYSTEM SECURITY APPROVAL (DAA)
DEPARTMENT OF DEFENSE
UNITED STATES OF AMERICA

Figure 8

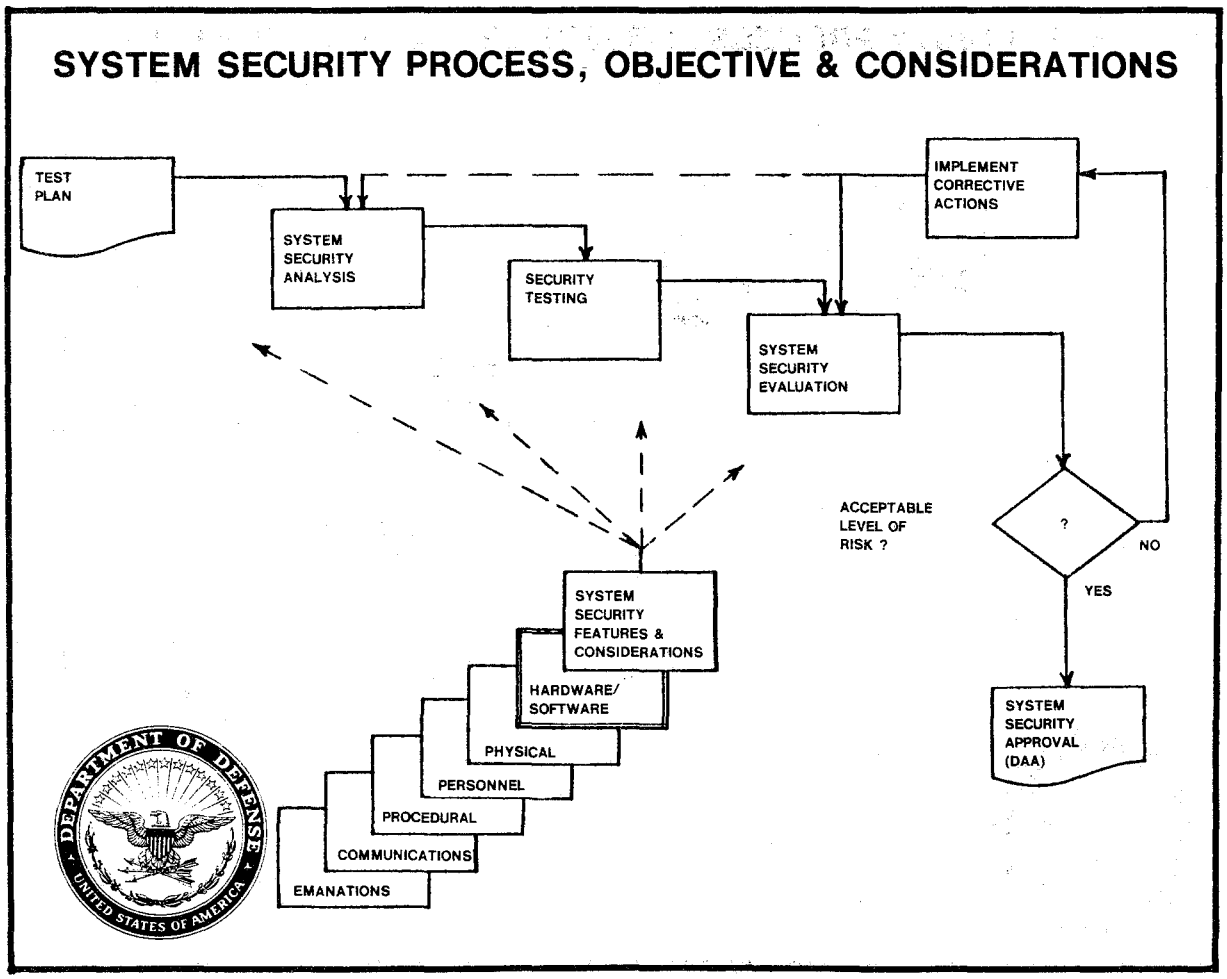

SYSTEM SECURITY PROCESS, OBJECTIVE & CONSIDERATIONS
TEST PLAN
SYSTEM SECURITY ANALYSIS
SECURITY TESTING
SYSTEM SECURITY EVALUATION
IMPLEMENT CORRECTIVE ACTIONS
ACCEPTABLE LEVEL OF RISK ?
?
NO
YES
SYSTEM SECURITY APPROVAL (DAA)
SYSTEM SECURITY FEATURES & CONSIDERATIONS
HARDWARE/ SOFTWARE
PHYSICAL
PERSONNEL
PROCEDURAL
COMMUNICATIONS
EMANATIONS
DEPARTMENT OF DEFENSE
UNITED STATES OF AMERICA

Figure 9

WHAT ARE ADP SYSTEM SECURITY MODES?

ALTERNATIVE APPROACHES TO CONFIGURING ADP SYSTEM SECURITY MEASURES WHERE THE RELATIVE RESPONSIBILITY OF ''AUTOMATED'' SECURITY MEASURES CAN BE VARIED WITH THAT OF ''CONVENTIONAL'' SECURITY MEASURES

Figure 10

SPECTRUM OF ADP SYSTEM SECURITY MODES
—REQUIREMENTS AND TRADEOFFS—

ADP SYSTEM SECURITY MODE:	"DEDICATED"	"SYSTEM HIGH"	"CONTROLLED"	"MULTI LEVEL"

GENERIC TRADEOFFS:

INCREASING HARDWARE/SOFTWARE SECURITY ROLE—INCREASING LEVEL OF RISK AND UNCERTAINTY →

DECREASING CONVENTIONAL SECURITY COST/CONSTRAINT ON ADP SYSTEM UTILIZATION: →

Figure 11

SPECTRUM OF ADP SYSTEM SECURITY MODES
—REQUIREMENTS AND TRADEOFFS—

ADP SYSTEM SECURITY MODE:	"DEDICATED"	"SYSTEM HIGH"	"CONTROLLED"	"MULTI LEVEL"
HARDWARE/SOFTWARE SECURITY ROLE:	NIL	NEED-TO-KNOW	CLEARANCE (AUGMENTED) & NEED-TO-KNOW	CLEARANCE AND NEED-TO-KNOW

GENERIC TRADEOFFS:

INCREASING HARDWARE/SOFTWARE SECURITY ROLE—INCREASING LEVEL OF RISK AND UNCERTAINTY →

DECREASING CONVENTIONAL SECURITY COST/CONSTRAINT ON ADP SYSTEM UTILIZATION: →

Figure 12

SPECTRUM OF ADP SYSTEM SECURITY MODES
—REQUIREMENTS AND TRADEOFFS—

ADP SYSTEM SECURITY MODE:	"DEDICATED"	"SYSTEM HIGH"	"CONTROLLED"	"MULTI LEVEL"
GENERAL SECURITY REQUIREMENTS:				
PHYSICAL AND PERSONNEL:				
CENTRAL COMPUTER FACILITY	HIGH			
REMOTE TERMINAL AREAS	HIGH			
COMMUNICATION LINKS:	HIGH			
HARDWARE/SOFTWARE SECURITY ROLE:	NIL	NEED-TO-KNOW	CLEARANCE (AUGMENTED) & NEED-TO-KNOW	CLEARANCE AND NEED-TO-KNOW

GENERIC TRADEOFFS:

INCREASING HARDWARE/SOFTWARE SECURITY ROLE—INCREASING LEVEL OF RISK AND UNCERTAINTY →

DECREASING CONVENTIONAL SECURITY COST/CONSTRAINT ON ADP SYSTEM UTILIZATION: →

Figure 13

SPECTRUM OF ADP SYSTEM SECURITY MODES
—REQUIREMENTS AND TRADEOFFS—

ADP SYSTEM SECURITY MODE:	"DEDICATED"	"SYSTEM HIGH"	"CONTROLLED"	"MULTI LEVEL"
GENERAL SECURITY REQUIREMENTS:				
PHYSICAL AND PERSONNEL:				
CENTRAL COMPUTER FACILITY	HIGH			HIGH
REMOTE TERMINAL AREAS	HIGH			VARIABLE
COMMUNICATION LINKS:	HIGH			VARIABLE
HARDWARE/SOFTWARE SECURITY ROLE:	NIL	NEED-TO-KNOW	CLEARANCE (AUGMENTED) & NEED-TO-KNOW	CLEARANCE AND NEED-TO-KNOW

GENERIC TRADEOFFS:

INCREASING HARDWARE/SOFTWARE SECURITY ROLE – INCREASING LEVEL OF RISK AND UNCERTAINTY

DECREASING CONVENTIONAL SECURITY COST/CONSTRAINT ON ADP SYSTEM UTILIZATION:

Figure 14

SPECTRUM OF ADP SYSTEM SECURITY MODES
—REQUIREMENTS AND TRADEOFFS—

ADP SYSTEM SECURITY MODE:	"DEDICATED"	"SYSTEM HIGH"	"CONTROLLED"	"MULTI LEVEL"
GENERAL SECURITY REQUIREMENTS:				
PHYSICAL AND PERSONNEL:				
CENTRAL COMPUTER FACILITY	HIGH	HIGH		HIGH
REMOTE TERMINAL AREAS	HIGH	HIGH		VARIABLE
COMMUNICATION LINKS:	HIGH	HIGH		VARIABLE
HARDWARE/SOFTWARE SECURITY ROLE:	NIL	NEED-TO-KNOW	CLEARANCE (AUGMENTED) & NEED-TO-KNOW	CLEARANCE AND NEED-TO-KNOW

GENERIC TRADEOFFS:

INCREASING HARDWARE/SOFTWARE SECURITY ROLE – INCREASING LEVEL OF RISK AND UNCERTAINTY

DECREASING CONVENTIONAL SECURITY COST/CONSTRAINT ON ADP SYSTEM UTILIZATION:

Figure 15

SPECTRUM OF ADP SYSTEM SECURITY MODES
—REQUIREMENTS AND TRADEOFFS—

ADP SYSTEM SECURITY MODE:	"DEDICATED"	"SYSTEM HIGH"	"CONTROLLED"	"MULTI LEVEL"
GENERAL SECURITY REQUIREMENTS:				
PHYSICAL AND PERSONNEL:				
CENTRAL COMPUTER FACILITY	HIGH	HIGH	HIGH	HIGH
REMOTE TERMINAL AREAS	HIGH	HIGH	VARIABLE	VARIABLE
COMMUNICATION LINKS:	HIGH	HIGH	VARIABLE	VARIABLE
HARDWARE/SOFTWARE SECURITY ROLE:	NIL	NEED-TO-KNOW	CLEARANCE (AUGMENTED) & NEED-TO-KNOW	CLEARANCE AND NEED-TO-KNOW

GENERIC TRADEOFFS:

INCREASING HARDWARE/SOFTWARE SECURITY ROLE – INCREASING LEVEL OF RISK AND UNCERTAINTY

DECREASING CONVENTIONAL SECURITY COST/CONSTRAINT ON ADP SYSTEM UTILIZATION:

Figure 16

SECURE SYSTEMS EVALUATION — POTENTIAL POLICY INCORPORATION

RELATIVE TECHNICAL SECURITY POSTURE

Category:	Features:	Examples:
1	DATA SECURITY	Most Current Sys
2	FUNCTIONAL SPECIFICATION REASONABLE PENETRATION RESULTS	"New EXEC 8" "VS"
3	REASONABLE MODERN PROGRAMMING TECHNIQUES LIMITED SYSTEM INTEGRITY MEASURES	MULTICS
4	FORMAL DESIGN SPECIFICATIONS SYSTEM INTEGRITY MEASURES	
5	PROVEN DESIGN SPECIFICATIONS VERIFIABLE IMPLEMENTATION LIMITED COVERT PATH PROVISIONS	KSOS KVM
6	VERIFIED DESIGN AUTOMATED TEST GENERATION EXTENDED COVERT PATH PROVISIONS REASONABLE DENIAL OF SERVICE PROVISIONS	

VULNERABILITY FACTORS (Technical) Example: Prog. Capabilities

INITIAL TARGET MODE

- DEDICATED
- SYSTEM HIGH
- CONTROLLED
- MULTILEVEL

Figure 17

SECURE SYSTEMS EVALUATION — POTENTIAL POLICY INCORPORATION

RELATIVE TECHNICAL SECURITY POSTURE

Category:	Features:	Examples:
1	DATA SECURITY	Most Current Sys
2	FUNCTIONAL SPECIFICATION REASONABLE PENETRATION RESULTS	"New EXEC 8" "VS"
3	REASONABLE MODERN PROGRAMMING TECHNIQUES LIMITED SYSTEM INTEGRITY MEASURES	MULTICS
4	FORMAL DESIGN SPECIFICATIONS SYSTEM INTEGRITY MEASURES	
5	PROVEN DESIGN SPECIFICATIONS VERIFIABLE IMPLEMENTATION LIMITED COVERT PATH PROVISIONS	KSOS KVM
6	VERIFIED DESIGN AUTOMATED TEST GENERATION EXTENDED COVERT PATH PROVISIONS REASONABLE DENIAL OF SERVICE PROVISIONS	

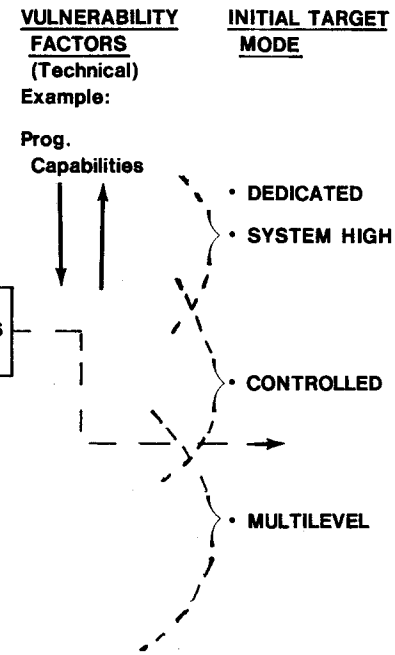


Figure 18

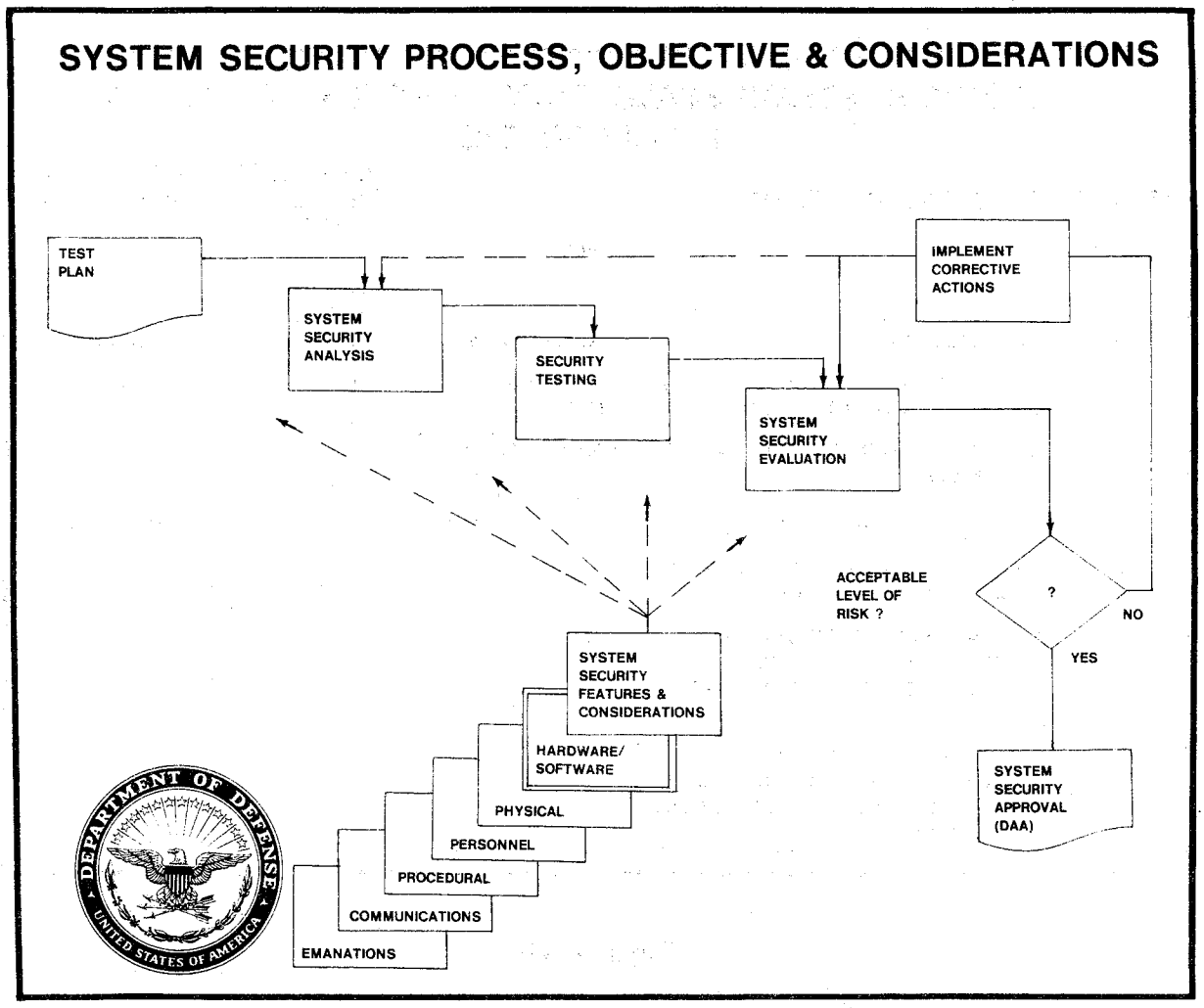

SYSTEM SECURITY PROCESS, OBJECTIVE & CONSIDERATIONS
TEST PLAN
SYSTEM SECURITY ANALYSIS
SECURITY TESTING
IMPLEMENT CORRECTIVE ACTIONS
SYSTEM SECURITY EVALUATION
ACCEPTABLE LEVEL OF RISK ?
?
NO
YES
SYSTEM SECURITY FEATURES & CONSIDERATIONS
HARDWARE/ SOFTWARE
PHYSICAL
PERSONNEL
PROCEDURAL
COMMUNICATIONS
EMANATIONS
SYSTEM SECURITY APPROVAL (DAA)
DEPARTMENT OF DEFENSE
UNITED STATES OF AMERICA

Figure 19

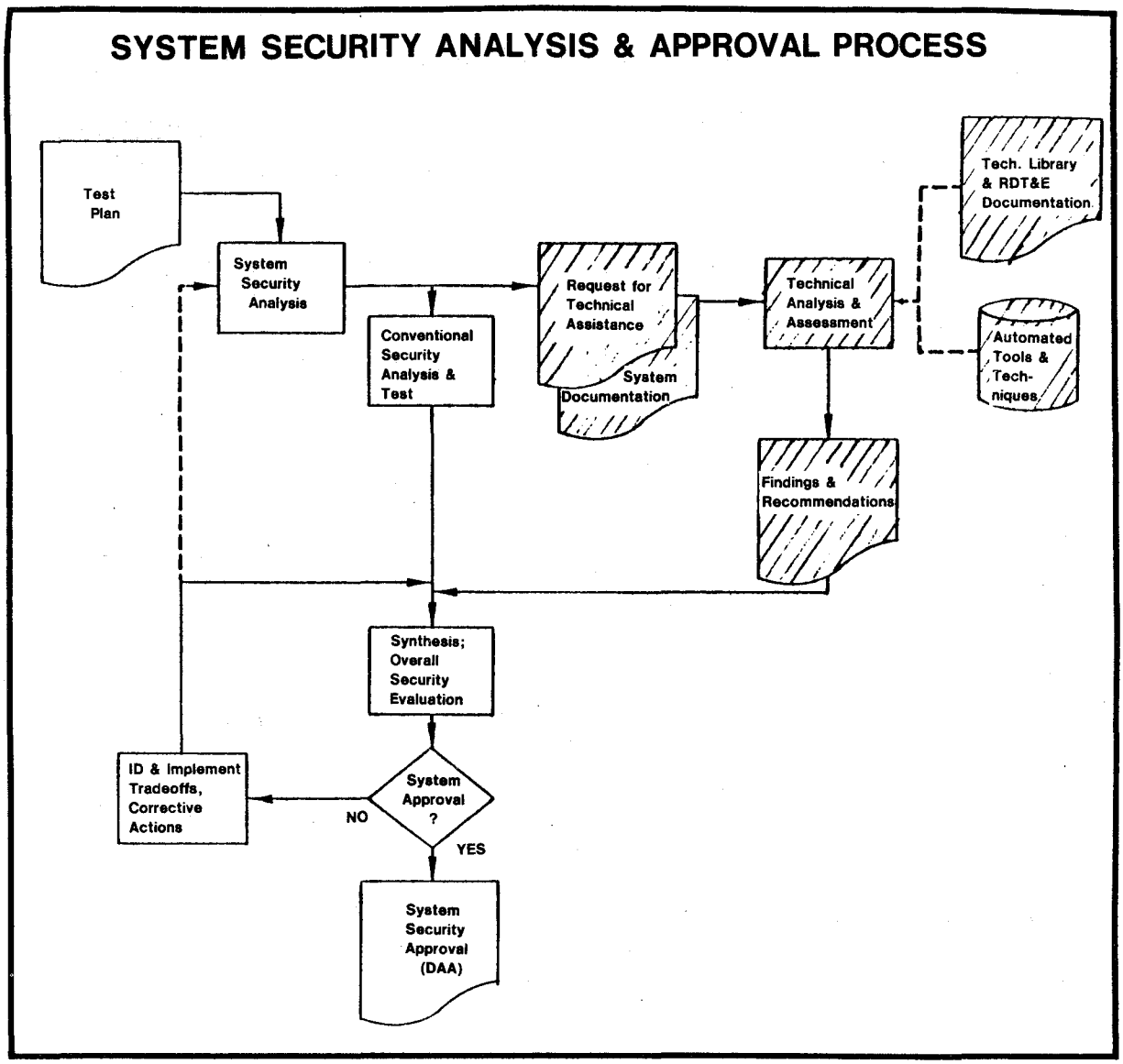

SYSTEM SECURITY ANALYSIS & APPROVAL PROCESS
Test Plan
System Security Analysis
Conventional Security Analysis & Test
Request for Technical Assistance
System Documentation
Technical Analysis & Assessment
Tech. Library & RDT&E Documentation
Automated Tools & Tech-niques
Findings & Recommendations
Synthesis; Overall Security Evaluation
System Approval ?
NO
YES
ID & Implement Tradeoffs, Corrective Actions
System Security Approval (DAA)

Figure 20

POLICY SURVEY SUBCOMMITTEE

TASK: REVIEW FEDERAL GOVERNMENT COMPUTER SECURITY POLICIES
- TO IDENTIFY EXISTING POLICIES, SCOPE, APPLICABILITY & RESPONSIBILITIES
- AT NATIONAL & DEPARTMENTAL LEVELS
- CLASSIFIED AND UNCLASSIFIED INFORMATION

APPROACH: QUESTIONNAIRE SURVEY OF SELECTED NATIONAL & EXECUTIVE BRANCH DEPARTMENT/AGENCY COMPUTER SECURITY DOCUMENTS
- DOCUMENTS ADDRESS COMPUTER SECURITY IN A COMPREHENSIVE SENSE
- QUESTIONNAIRE DESIGNED TO EXTRACT KEY PROGRAM INDICATORS
- DEFINITION OF ''SAMPLE'' UNIVERSE TO FOCUS ON PREPONDERANCE OF ADPE & CABINET-LEVEL DEPARTMENTS

COVERAGE OBJECTIVES:

1. POLICIES
   - NATIONAL LEVEL
   - EXECUTIVE DEPARTMENT/AGENCY LEVEL
2. PROGRAM OVERSIGHT MECHANISMS (SECONDARY)
   - NATIONAL LEVEL
   - DEPARTMENTAL/AGENCY LEVEL

Figure 21

SURVEY FOCUS — EXECUTIVE BRANCH DEPARTMENTS & AGENCIES
GSA AUTOMATIC DATA PROCESSING EQUIPMENT INVENTORY

EXECUTIVE DEPARTMENT/ AGENCY	NUMBER OF ADP SYSTEMS	CUMULATIVE % OF TOTAL
ARMY	1126	
NAVY	1473	
AIR FORCE	1704	
• DEFENSE	4535	49%
• ENERGY	2395	75
NASA	489	80
• TRANSPORTATION	371	84
• TREASURY	174	
• HEW*	137	
• AGRICULTURE	102	
• JUSTICE	33	
NRC	1	88.6
DIA	(INCLUDED IN DOD TOTAL)	
CIA	(NOT INDICATED IN INVENTORY)	
NSA	(NOT INDICATED IN INVENTORY)	
	8237 (OUT OF 9299)	

• CABINET LEVEL DEPARTMENTS
* CONSIDERED ONE DEPARTMENT FOR SURVEY PURPOSES

Figure 22

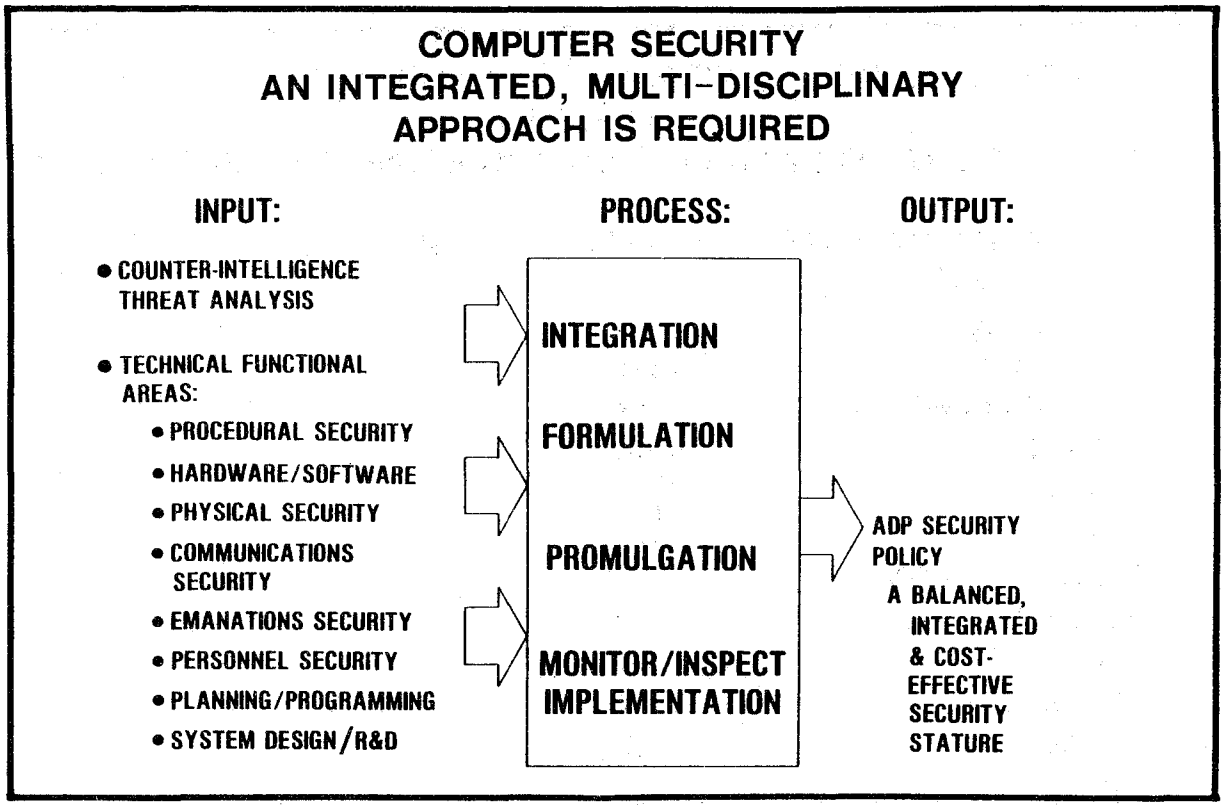


Figure 23

— RESULTS —
FEDERAL DEPARTMENTS AND AGENCIES

- 15 DEPARTMENTS AND AGENCIES
- 27 QUESTIONNAIRES
- 32 DOCUMENTS

QUESTIONNAIRE RESPONSES

• APPLICABILITY	
— "IN-HOUSE"	93%
— "OUT-HOUSE"	85%
• PROTECTION SCOPE	
— ADP SYSTEMS	100%
— AREAS HOUSING SYSTEMS	82%
— OTHER SYSTEMS RESOURCES	63%
— COMPUTER SOFTWARE	89%

Figure 24

— RESULTS —
FEDERAL DEPARTMENTS AND AGENCIES (Continued)

QUESTIONNAIRE RESPONSES

• PROTECTION SCOPE (CONTINUED)	
— LIFE CYCLE COVERAGE?	
• ADP SYSTEMS	85%
• DATA/APPLICATION SYSTEMS	63%
• COMPUTER SECURITY SUBDISCIPLINES INCLUDED	
— PERSONNEL SECURITY	96%
— PHYSICAL SECURITY	100%
— COMMUNICATIONS SECURITY	89%
— EMANATIONS SECURITY	70%
— ADMINISTRATIVE/PROCEDURAL SECURITY	96%
— HARDWARE/SOFTWARE SECURITY	96%

Figure 25

DEPARTMENT AND AGENCY RESULTS

• PROGRAM COMPONENT ELEMENTS:	
— ASSIGNMENT OF RESPONSIBILITY	
• FOR POLICY	96%
• FOR SYSTEMS	93%
— MANAGEMENT CONTROL PROCESS	96%
— DESIGNATED APPROVING AUTHORITIES	78%
— OVERALL SECURITY SPECIFICATIONS/REQUIREMENTS	85%
— SECURITY EVALUATION REQUIRED FOR SYSTEM OPERATION	74%
— AUDIT OR OTHER FOLLOW-UP SECURITY EVALUATION	78%
— RISK ANALYSIS METHODOLOGIES	70%
— SECURITY REQUIREMENTS FOR PROCUREMENT	74%
— REQUIREMENTS FOR CONTINGENCY PLANNING	67%
— PERSONNEL SCREENING	78%
— SPECIFIED WAIVER AUTHORITY	56%
— REQUIREMENT FOR ADP SECURITY BUDGET	15%
• NUMBER OF ADP SYSTEMS COVERED	OVER 8,200
• NUMBER OF PAGES	1,316

Figure 26

— RESULTS —
NATIONAL LEVEL POLICIES

- 13 DOCUMENTS
- 5 QUESTIONNAIRES

QUESTIONNAIRE RESPONSES

• APPLICABILITY	
— ''IN-HOUSE''	100%
— ''OUT-HOUSE''	100%
• PROTECTION SCOPE	
— ADP SYSTEMS	100%
— AREAS HOUSING SYSTEMS	80%
— COMPUTER SOFTWARE	80%
— OTHER SYSTEMS RESOURCES	60%
— LIFE CYCLE COVERAGE?	
• ADP SYSTEMS	80%
• DATA/APPLICATION SYSTEMS	80%

Figure 27

— RESULTS —
NATIONAL LEVEL POLICIES (CONTINUED)

QUESTIONNAIRE RESPONSES

• COMPUTER SECURITY SUBDISCIPLINES INCLUDED	
— PERSONNEL SECURITY	80%
— PHYSICAL SECURITY	100%
— COMMUNICATIONS SECURITY	100%
— EMANATIONS SECURITY	60%
— ADMINISTRATIVE/PROCEDURAL SECURITY	100%
— HARDWARE/SOFTWARE SECURITY	100%

Figure 28

— RESULTS —
NATIONAL LEVEL POLICIES (CONTINUED)

• PROGRAM COMPONENT ELEMENTS:	
— ASSIGNMENT OF RESPONSIBILITY	
• FOR POLICY	100%
• FOR SYSTEMS	80%
— MANAGEMENT CONTROL PROCESS	100%
— DESIGNATED APPROVING AUTHORITIES	80%
— OVERALL SECURITY SPECIFICATIONS/REQUIREMENTS	100%
— SECURITY EVALUATION REQUIRED FOR SYSTEM OPERATION	80%
— AUDIT OR OTHER FOLLOW-UP SECURITY EVALUATION	80%
— RISK ANALYSIS METHODOLOGIES	60%
— SECURITY REQUIREMENTS FOR PROCUREMENT	60%
— REQUIREMENTS FOR CONTINGENCY PLANNING	20%
— PERSONNEL SCREENING	80%
— SPECIFIED WAIVER AUTHORITY	80%
— REQUIREMENT FOR ADP SECURITY BUDGET	20%
• NUMBER OF PAGES	128

Figure 29

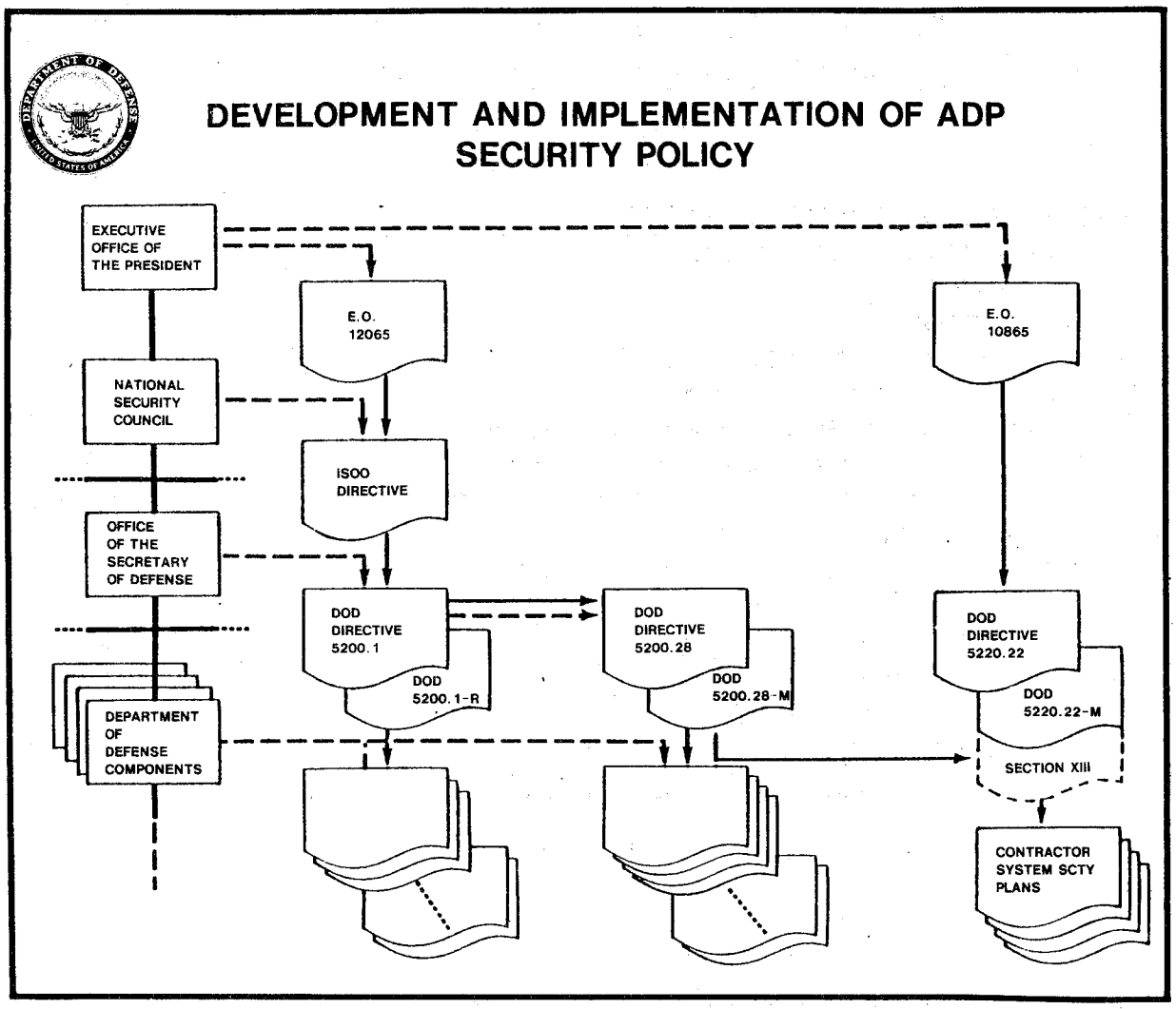

DEVELOPMENT AND IMPLEMENTATION OF ADP SECURITY POLICY
EXECUTIVE OFFICE OF THE PRESIDENT
NATIONAL SECURITY COUNCIL
OFFICE OF THE SECRETARY OF DEFENSE
DEPARTMENT OF DEFENSE COMPONENTS
E.O. 12065
ISOO DIRECTIVE
DOD DIRECTIVE 5200.1
DOD 5200.1-R
DOD DIRECTIVE 5200.28
DOD 5200.28-M
E.O. 10865
DOD DIRECTIVE 5220.22
DOD 5220.22-M
SECTION XIII
CONTRACTOR SYSTEM SCTY PLANS

Figure 30

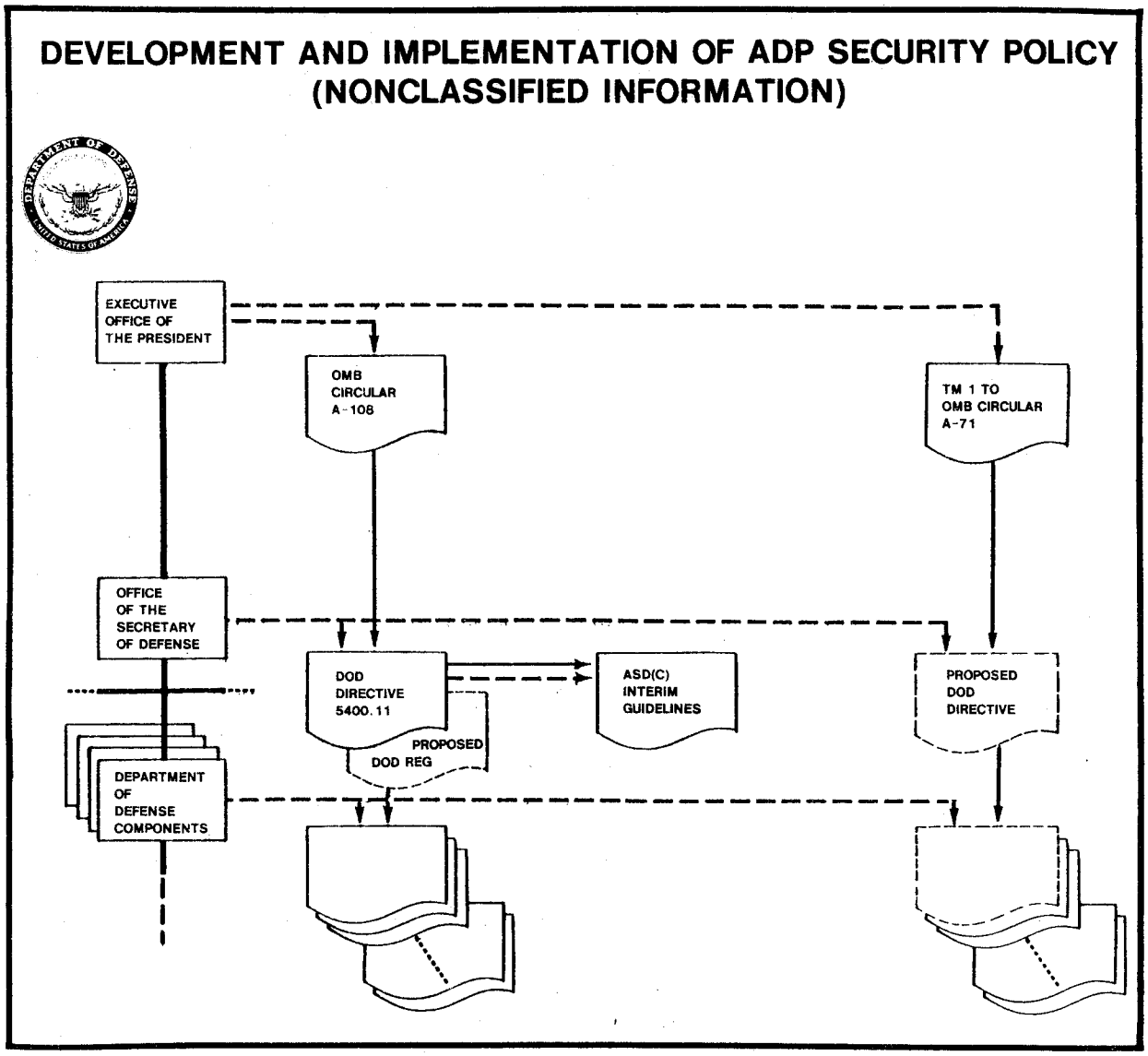
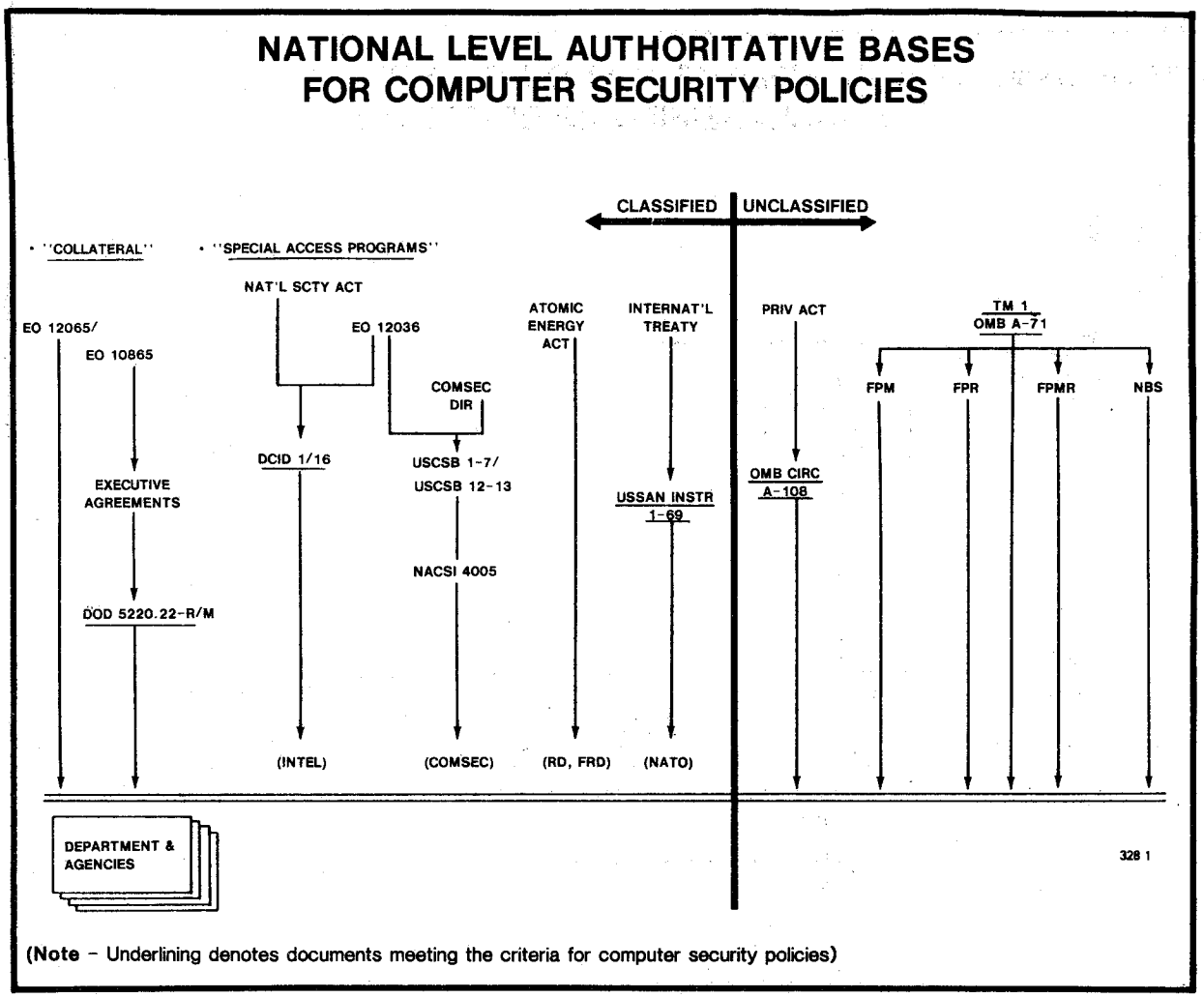

DEVELOPMENT AND IMPLEMENTATION OF ADP SECURITY POLICY
(NONCLASSIFIED INFORMATION)
EXECUTIVE OFFICE OF THE PRESIDENT
OMB CIRCULAR A-108
TM 1 TO OMB CIRCULAR A-71
OFFICE OF THE SECRETARY OF DEFENSE
DOD DIRECTIVE 5400.11
PROPOSED DOD REG
ASD(C) INTERIM GUIDELINES
PROPOSED DOD DIRECTIVE
DEPARTMENT OF DEFENSE COMPONENTS

Figure 31



(Note - Underlining denotes documents meeting the criteria for computer security policies)

Figure 32

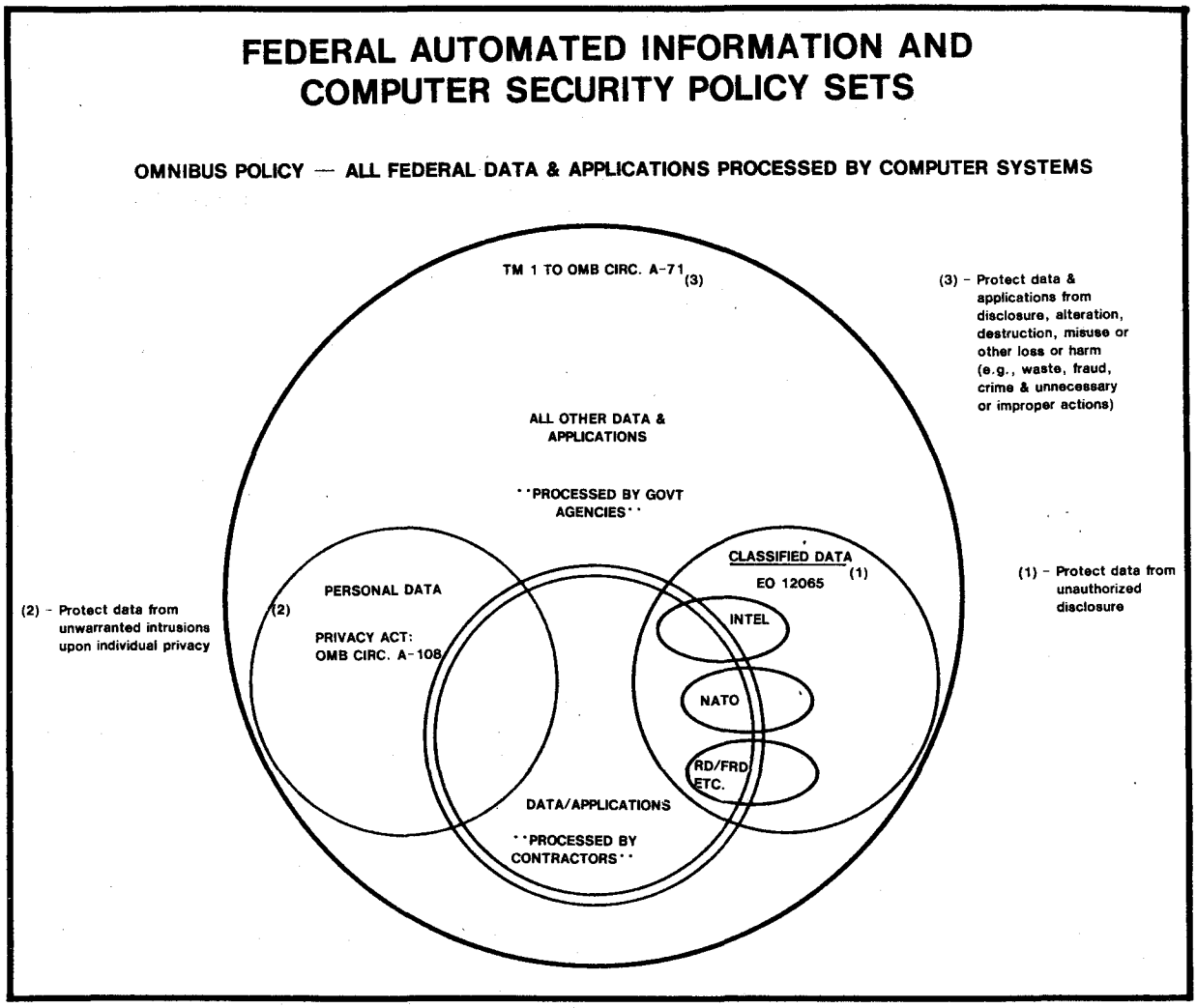

FEDERAL AUTOMATED INFORMATION AND COMPUTER SECURITY POLICY SETS
OMNIBUS POLICY — ALL FEDERAL DATA & APPLICATIONS PROCESSED BY COMPUTER SYSTEMS
TM 1 TO OMB CIRC. A-71 (3)
(3) - Protect data & applications from disclosure, alteration, destruction, misuse or other loss or harm (e.g., waste, fraud, crime & unnecessary or improper actions)
ALL OTHER DATA & APPLICATIONS
"PROCESSED BY GOVT AGENCIES"
CLASSIFIED DATA
EO 12065 (1)
(1) - Protect data from unauthorized disclosure
PERSONAL DATA
(2)
PRIVACY ACT: OMB CIRC. A-108
(2) - Protect data from unwarranted intrusions upon individual privacy
INTEL
NATO
RD/FRD ETC.
DATA/APPLICATIONS
"PROCESSED BY CONTRACTORS"

Figure 33

NATIONAL LEVEL INTEREST

1976 GAO REPORTS:

- ''IMPROVEMENTS NEEDED IN MANAGING AUTOMATED DECISIONMAKING BY COMPUTERS THROUGHOUT THE FEDERAL GOVERNMENT'' (APR 76)
- ''COMPUTER-RELATED CRIMES IN FEDERAL PROGRAMS'' (APR 76)
- ''MANAGERS NEED TO PROVIDE BETTER PROTECTION FOR FEDERAL AUTOMATIC DATA PROCESSING FACILITIES'' (MAY 76)

SENATE COMMITTEE ON GOVERNMENT OPERATIONS:

- ''COMPUTER ABUSES-PROBLEMS ASSOCIATED WITH COMPUTER TECHNOLOGY IN FEDERAL PROGRAMS & PRIVATE INDUSTRY'' (JUN 76)

1977

- ''COMPUTER SECURITY IN FEDERAL PROGRAMS'' (FEB 77)

OMB:

- ''SECURITY OF FEDERAL AUTOMATED INFORMATION SYSTEMS.'' TRANSMITTAL NO. 1 TO OMB CIRCULAR NO. A-71

DRAFT FOR COORDINATION (SEP 77)

1978 FINAL ISSUANCE (JUL 78)

PRESIDENT: INITIATIVE TO ATTACK FRAUD & WASTE

- DOD STEERING GROUP ON OVERSIGHT OF DEFENSE ACTIVITIES

SUBCOMMITTEE ON COMPUTER FRAUD

GAO REPORTS:

1979

- AUTOMATED SYSTEMS SECURITY-FEDERAL AGENCIES SHOULD STRENGTHEN SAFEGUARDS OVER PERSONAL AND OTHER SENSITIVE DATA'' (JAN 79)
- GAO LETTER TO SECDEF (MAR 79)

1982

- ''FEDERAL INFORMATION SYSTEMS REMAIN HIGHLY VULNERABLE TO FRAUDULENT, WASTEFUL, ABUSIVE, AND ILLEGAL PRACTICES'' (APR 82)

Figure 34

AREAS OF RECOMMENDED ATTENTION

1. LACK OF TOP MANAGEMENT SUPPORT
2. LACK OF RESOURCES, R&D AND OPERATIONAL
3. HARDWARE/SOFTWARE SECURITY DEVELOPMENT
4. MORE EFFECTIVE INTEGRATION AND COORDINATION AMONG NATIONAL POLICY PROMULGATING ACTIVITIES
5. MORE IMPLEMENTING GUIDANCE FOR COMPUTER SECURITY SUBDISCIPLINES (IN ADDITION TO 3., ABOVE, COMSEC AND EMSEC GUIDANCE KEYED TO ADP SYSTEMS AND NETWORKS)

Figure 35

WORD PROCESSING & ALLIED AUTOMATED INFORMATION SYSTEMS

- SYSTEM SECURITY NEEDS ANALOGOUS TO ADPE
  - LIFE CYCLE
  - COMPREHENSIVE, MULTIDISCIPLINARY & SYSTEMATIC
- EXPAND SCOPE OF ADP SECURITY POLICY
  - SIMPLIFIED REQUIREMENTS FOR STAND-ALONE, SINGLE-USER SYSTEMS
  - SHARED SYSTEMS
    - SECURITY EVALUATION & APPROVAL
    - DESIGNATION OF RESPONSIBLE PERSON
  - AVOID UNWARRANTED DUPLICATION OF POLICIES, PROCEDURES, PROGRAMS TERMINOLOGY AND ASSOCIATED RESOURCES
- ENHANCEMENTS
  - ADDITIONAL PERSONNEL SECURITY REQUIREMENTS
  - POLICY GUIDANCE ON AREA CONTROLS & ADJUSTMENTS
  - EXPANDED GUIDANCE ON MEDIA CLEARANCE & DECLASSIFICATION

Figure 36

COMPUTER SECURITY CONSIDERATION IN THE COMPUTER LIFE CYCLE

LTC Lawrence A. Noble
Computer Security Policy Analyst
Director of Computer Resources
Headquarters, U.S. Air Force

LTC Noble received a B.S. from the U.S. Military Academy, an M.S. in Mathematics from New York University, and an M.B.A. from Auburn University. He is also a graduate of the Air War College, Air Command and Staff College, and Staff Officer's School. He has had the following assignments: 1963–1965, performed flight safety studies for missile launches from the Air Force Eastern Test Range at Cape Canaveral. 1965–1970, Mathematical Analyst at NSA. 1970–1973, Computer Systems Analyst, participated in development of a Data Base Management System at the Defense Intelligence Agency. 1974–1976, managed computer technology Research and Development projects including computer security R&D at ESD. 1976–1978, Chief of Requirements & Test Division of a program office charged with development of an automated tactical command and control system at the Electronics Systems Division, Air Force Systems Command. 1978–Present, Computer Security Policy Analyst in the Policy and Procedures Group of the Directorate of Computer Resources, as the Air Force focal point for Computer Security Policy.

My job for over 3 years has been to serve as the Air Staff focal point for computer security in the Pentagon. This includes staffing and computer security policy.

I work closely with an organization called the Air Force Computer Security Program Office at Gunter AFB, Alabama.

I would like to discuss an approach that we have taken in recent Air Force computer policy to address computer security requirements in the computer life cycle, while integrating the requirements of national policy, particularly OMB Circular A71.

At the level of Air Force policy, we try to be general enough to allow for many different situations among our field organizations. So, what I am going to discuss is really a conceptual framework.

Because of the decentralization of authoirty for security (I believe Commanders must decide on the proper trade-offs between security, cost and other constraints), and because of the relative newness of this policy, I cannot cite specific case histories. But, I believe the concepts are sound and really what many have already been doing.

It is important to surface the security requirement early. (It has become almost axiomatic that retrofitting security to a system is not the way to go.) So, we want to identify the sensitivity of the proposed system very early.

One must consider whether the system will include classified information, or other sensitive information; for example, personal data subject to the Privacy Act. In addition, one should consider whether the system performs critical functions; for example, functions affecting the safety of human life.

At this point, it is the functional user, the customer, who must provide this information, while it is incumbent on the Data Automator to identify the questions and ensure that they are properly addressed.

We use something called a DAR, "Data Automation Requirement," to document the requirement and a functional description to provide more detail about the system to be built. Both of these are jointly prepared by the potential customer and the data automation people.

I should add that we have also called for identification of any special constraints such as simultaneous sharing of the system by users at different security levels in our DAR format.

At this point, the risk analysis is initiated. We want to begin to consider alternatives and look at trade-offs between security measures and costs. As the system development progresses, the scope of alternative security measures will narrow; however, at this point we are looking at system-wide choices. For example, use of separate computers for different security levels of multi-level systems.

The functional requirements are then approved with the security risks considered. Current policy defines who has authority to make this approval. The Data Project Directive then directs the actions to satisfy the requirements.

In response to this directive, the project manager must then prepare the plan, called the "Data Project Plan," to implement the direction, including necessary actions to address security.

The requirements are reviewed and fine-tuned at the System Requirements Review. Here, the individual who will be responsibile for approving or accrediting the product from a security point of view, the "Designated Approving Authority" or "DAA," should be represented. The intent of course is to avoid forcing the DAA into a binary decision at the end of the development or acquisition after much time and money has been invested, but allow the DAA to influence the resulting product, or at least raise the "red flags" early in the process if he or she believes that security will not be adequate in the final system.

In developing the specifications, the risk analysis should be revisited now that the system is better understood.

At this time, the plans for validating the security measures must be addressed, and the DAA should be involved.

In the subsequent design reviews, DAA representatives should be involved to review security specification and ST&E plans.

Once the system has been developed or acquired, the ST&E may begin. This will lead initially to the certification of various portions of the system.

I view certification as primarily a technical process which addresses how well the delivered system meets its security specifications and documents assumptions and constraints. The certifications of various portions of the system are inputs to the overall security approval by the DAA.

The appropriate manager certifies the facility, the Automated Data System (ADS), that is the application system, and the ADP system (primarily the hardware and systems software).

Following the DA approval of the system, security concerns should not be ignored. The configuration management process should address security relevance of changes in the system. Furthermore, OMB Circular A71 requires periodic audits or evaluations, and risk analyses. The DAA would then be called on to reapprove the system, if appropriate, based on the results of these studies.

In summary, what I have discussed is really a conceptual framework that the Air Force has established as policy for addressing security throughout the life cycle.

We feel that it is important to surface the requirement early and then follow the disciplined practices of life cycle management to ensure that the requirement is adequately addressed, all the while being sure to include the right people in the process.

Of course, the effort expended in any of these steps should be commensurate with what is at stake.

This policy is relatively new and it will take time for it to be fully assimilated, but I believe it will prove out. □

COMPUTER SECURITY IN THE LIFE CYCLE

- IDENTIFY SENSITIVITY OF PROPOSED SYSTEM
  - FUNCTIONAL USER WITH DATA AUTOMATION
  - DOCUMENT IN DATA AUTOMATION REQUIREMENT (DAR) AND FUNCTIONAL DESCRIPTION
- INITIATE RISK ANALYSIS
- APPROVE FUNCTIONAL REQUIREMENTS
  - ADP APPROVAL AUTHORITY
  - DOCUMENT IN DATA PROJECT DIRECTIVE (DPD)
- DEVELOP DATA PROJECT PLAN
  - PROJECT MANAGER
- CONDUCT SYSTEM REQUIREMENTS REVIEW (SRR)
  - DESIGNATED APPROVING AUTHORITY (DAA) REPRESENTATIVES
- DEVELOP SPECIFICATIONS
  - REVIEW AND UPDATE RISK ANALYSIS (SENSITIVITY AND RISK ASSESSMENTS AND ECONOMIC TRADE-OFFS)
    - COMPLETE BEFORE SPECIFICATION APPROVAL
  - PLANS FOR SECURITY TEST AND EVALUATION (ST&E)
  - DAA REVIEW
- DESIGN REVIEWS
- ACQUISITION/DEVELOPMENT
- ST&E
- CERTIFICATION: FACILITY, AUTOMATED DATA SYSTEM (ADS), ADP SYSTEM
- SECURITY APPROVAL
  - DAA
- OPERATIONS
  - CONFIGURATION MANAGEMENT
  - PERIODIC AUDITS/EVALUATIONS
  - REVALIDATE RISK ANALYSIS
  - DAA REAPPROVAL

SUMMARY

- SURFACE THE REQUIREMENT EARLY
- FOLLOW THE LIFE CYCLE PROCESS TO ADDRESS SECURITY REQUIREMENTS
- MAKE SURE THE RIGHT PEOPLE ARE INVOLVED

NON-DISCRETIONARY CONTROLS FOR COMMERCIAL APPLICATIONS*

Steven B. Lipner
Engineering Manager
Computer Security Advanced Development
Digital Equipment Corporation

Steve has been active in computer security R&D since 1970, and was a member of the 1972 Computer Security Technology Planning Study Panel (the Anderson Panel). Before joining DEC he was with the MITRE Corporation, where he managed a number of the early efforts in computer security. Among these were the development of the Bell-LaPadula (lattice) model for secure computer systems and the development of the first prototype security kernel that implemented the lattice model on a PDP-11/45. At DEC, Steve is responsible for advanced development projects dealing with security-enhanced operating systems, security kernels, and network security.

INTRODUCTION

The lattice model of non-discretionary access control in a secure computer system was developed in the early Seventies [BLaP]. The model was motivated by the controls used by the Defense Department and other "national security" agencies to regulate people's access to sensitive information. Since that time, the lattice model has enjoyed reasonable success in several computer systems used to process national security classified information [MME; Multics; SACDIN]. "Reasonable success," in this context, means that human beings accept the systems and are able to use them to accomplish useful work, without the protection provided by the non-discretionary controls unduly interfering with productivity or perceived convenience.

In the late Seventies and early Eighties, the Defense Department, through its DoD Computer Security Initiative and DoD Computer Security Evaluation Center, has attempted to raise general awareness of computer security issues and to urge manufacturers to improve the security of commercial operating system offerings. The DoD notion of improved security [Nibaldi] seems to constitute (1) better system integrity to assure that users and programs can only access information as authorized; and (2) the introduction of the lattice model as a fundamental operating system mechanism for access authorization.

There is little question as to the need for improved system integrity, though different application environments will require different levels of integrity. (The draft DoD evaluation criteria recognize this variation and specify graded levels of achieving them.) The lattice model, however, is derived from national security policies and regulations, leading to some question as to its suitability in non-national security data processing environments. This paper examines some ways in which the lattice model non-discretionary controls might be used in commercial data processing.

The first section of this paper introduces the basic notion and derivation of the lattice model. It also introduces, for purposes of comparison, the results of a previous attempt to apply the lattice model in a commercial environment. The second section examines commercial requirements for information security and derives an alternate application of the lattice model more suited to these requirements. The final section introduces a second component of the lattice model (the integrity model) and revisits the commercial application to show an alternate formulation for achieving equivalent security.

The paper's major conclusions are that the lattice model may in fact be applicable to commercial data processing, but that such application will require ways of looking at security requirements different from those prevalent in the national security community. The joint involvement of people experienced in

commercial applications and of people who understand the lattice security model will probably provide the most productive course towards the development of successful commercial applications of the lattice model.

THE LATTICE MODEL

The DoD Security System

The lattice model was developed from an examination of the Defense Department's (DoD) information security policies and the ways in which they might be enforced in the context of a multi-user computer system. The DoD information security policy gives each document a classification level, L, and a (possibly empty) set of categories, C. The security levels are strictly ordered (i.e., form a chain). The categories tend to have no ordering or precedence, but subsets of categories are ordered by set inclusion. The combination of a classification level and a set of categories is referred to as an access class. Figure 1 identifies the DoD classification levels and their order, and a few example categories.

Top Secret > Secret > Confidential > Unclassified

(a) Classification Levels

Nuclear, NATO, Intelligence

(b) Example Categories

Figure 1. DoD Levels and Categories

The classification levels and categories in the DoD system are also used to level a set of clearances that can be granted to people. The two fundamental security rules of the DoD system are then: (1) An individual can only have access to (read) a document if the individual's clearance level is greater than or equal to the document's classification level, and if the individual is "cleared" for all categories applied to the document; and (2) Only a specially authorized individual may reduce (downgrade) the level or remove categories from a classified document.

The rules identified above were applied more or less directly in the formulation of the lattice security model. Access rights were associated with processes that execute on behalf of users. Programs are treated as objects that are executed by processes—but have no rights themselves.

In a computer, as in the world of people and documents, a process may not read a file unless it (or the person on whose behalf it operates) is cleared for such access. This rule is called the "simple security condition" (Figure 2a). The second rule of the lattice model derives from the second rule of the DoD system—though its manifestation has often been perceived as overly restrictive: A process may not write a file (or other object) unless the access class of the object is greater than or equal to* that of the process.

This second rule does not reflect a restriction on a cleared individual's ability to write an unclassified document—indeed the purpose of a clearance is to certify that an individual is trusted not to compromise or downgrade sensitive information. Rather, the second rule reflects the reality that a process executing an arbitrary program is not trusted to write files with lower access classes than those it may have read and remembered in its process state. This conservative rule (Figure 2b) is designed to enforce the DoD restriction on downgrading the access class of a document and is called the *-property or confinement property.

*One access Class A is greater than or equal to another B if and only if the classification level for A is greater than or equal to that for B, and A's set of categories contains B's. More formally:

$$L_A \geq L_B \ \& \ \{C\}_A \supseteq \{C\}_B$$

A process P may read an object if

$$L_p \geq L_o \ \& \ \{C\}_p \supseteq \{C\}_o$$

(a) Simple Security Condition

A process P may write an object O if

$$L_p \leq L_o \ \& \ \{C\}_p \subseteq \{C\}_o$$

(b) Confinement Property

Figure 2. Lattice Model Security Rules

Systems that apply the lattice model have applied it as a "non-discretionary" access control policy in the sense that neither users nor programs may reduce the access class of information (downgrade it) except by appealing to special operating system software that must be invoked directly by a human being (in some cases a system security officer) and not by an untrusted program. The *-property and the restriction on downgrading assure that no chain of program actions can result in an unauthorized flow of information to a file or other object of reduced access class [BlaP]. Discretionary control mechanisms like access control lists, in contrast, allow the construction of sequences of program actions that can result in observation or modification of information by an unauthorized person. For example, a program operating on behalf of an authorized user might, without the person's knowledge, alter an access control list or copy a file to a new version with a different less restrictive access control list. With a non-discretionary lattice mechanism, no program action can allow access by a user who does not have the access class (clearance) corresponding to that of the original file.

Historically, attempts to apply the lattice model in non-DoD environments have begun by identifying levels and categories that reflect the levels of sensitivity and organizational divisions in the subject environment. For example, Figure 3 presents the levels and categories identified in applying a prototype security-enhanced operating system in a hypothetical corporation. The definitions of the categories are more or less self-explanatory. The levels are simply intended to reflect a notion of increasing information sensitivity.

Security Levels
- Public
- Sensitive
- Confidential

Categories
- Manufacturing
- Personnel
- Engineering
- Accounting

Figure 3. Security Levels and Categories for a Hypothetical Corporation

The application of levels and categories like those in Figure 3 has an unfortunate tendency to fall apart on close inspection. While the company may in fact have organizational divisions like those identified in the categories, there will almost never be a notion of individuals having clearance that corresponds to the security levels. Access to information will normally be granted on a need-to-know basis, depending on individuals' job responsibilities or, worse (from a formal point of view), on a basis of aggregation in which a "little" information is unrestricted, while a "lot" is closely held. The former kind of restriction may be enforced by introducing still more categories; the latter seems to require that an access control system remember large amounts of history, and is thus very difficult to enforce.

There is a more fundamental problem with the lattice model application sketched in Figure 3: It deals only with the reading and compromise of information. In actual practice, commercial institutions may be more interested in unauthorized modification of information than in its compromise. The remainder of this

paper re-examines the commercial information security requirements, then proposes alternate applications of the lattice model to meet those requirements.

CONTROLLING MODIFICATION WITH THE LATTICE MODEL

Revisiting the Requirements

In an attempt to solve the problems identified above, the author initiated an informal search for commercial security policies and requirements. Much of the information gathered centered on who can alter information, rather than who can read it. Furthermore, a great deal of the concern expressed by EDP auditors and EDP security officers had to do with the integrity of financial and finance-related information and with control over the introduction of programs that will operate on that information. Much less concern was expressed about the activities of system users who will write or use programs to process "their own" data of whatever sensitivity.

To make the suggestions above more concrete, the following list of requirements is presented. It is taken from the suggestions of a senior EDP auditor, and is directed toward "production" systems that might operate on financial, material control, or order-entry data:

(1) Users of the application will use production programs and data bases; they will not write their own programs to operate on the production data bases.

(2) Application program developers will do their development and testing in a test environment, and have no access to the production (source or object) programs or data bases. If they need such access, they may be provided with copies of the information they need through a special process.

(3) "Promotion" of programs from development to production status is a controlled event.

(4) The installation system programmers' actions shall be controlled and audited.

(5) Management and audit functions shall have access to the system state and an audit trail of selected activities (both system actions and user interactions with the applications).

The requirements cited above bear little apparent resemblance to the usual statement of DoD security requirements. The following paragraphs will show, however, that the lattice model can be applied to do an effective job of meeting these requirements.

A Lattice Application to Control Modification

In forming a lattice model to meet the requirements identified above, the initial step was to define a set of access classes appropriate to achieving the desired restrictions. Figure 4 shows the levels and categories that will be used in this example. The "system-low" level (SL) corresponds to unclassified in the DoD system, and information at this level is readable by all processes. The application of the remaining level (audit-manager) and categories will be made clear below.

Security Levels
- Audit-Manager (AM)
- System-Low (SL)

Categories
- Production-Date (PD)
- Development (D)
- System-Development (SD)
- Tools (T)

Figure 4. Levels and Categories

The application of the access classes to the system users (and to processes running on their behalf) is outlined in Figure 5. Users (of application programs), application developers, and system programmers each have system low security level and two categories. In each case, the first category is that of the information the population must manipulate (production data in the case of users; application programs and test data in the case of application developers), and the second is that of the programs the population must use. The audit and management population has access to information of any category, and a security level of audit-

manager. Finally, a system control function is defined with system-low level and access to each category, and a "downgrade" privilege to change categories and transform (for example) developing programs into production code—presumably after suitable testing and quality assurance review.

There is one unusual aspect of the assignment of access classes to user populations in Figure 5. Some populations must be required to operate only through processes having all of the categories or levels for which the users are "cleared." In particular, the "check" on system audit and management personnel derives from the fact that they may not login at system low with one or more categories and operate on data or programs as users or programmers, then use their auditing function to conceal changes that they may have made or actions that they may have taken. Furthermore, if a programmer could somehow introduce a "system low" program for use by production users, there would be nothing in this lattice model to prevent that program from corrupting the production data base to which the users have (lattice model) access. The next section introduces an additional model that helps address this program. But, in general, users must be restricted as to which subsets of their full "clearance" they exercise.

User Community	*Access Class*
System Management or Audit	AM; any set of Categories
User	SL; PD & PC
Application Developer	SL; D & T
System Programmer	SL; SD & T
System Control	SL; PD, PC, D, SD, T plus "downgrade" privilege

Figure 5. Users' Access Classes

The assignment of access classes to files (objects) is depicted in Figure 6. The program objects (production code, tools*) each have a single category and are intended to be read-only (unmodified). Objects subject to manipulation (production data; system and application programs under development) have two categories each—that of the object itself and that of the program that must operate on it—so that a process executing the program will be allowed by the confinement property to write the object. Audit trail information is developed with the category(ies) of the activity being audited and is "written up" to the higher audit-manager security level.

Files	*Access Class*
Production Data	SL; PD & PC
Production Code	SL; PC
Developing Code/Test Data	SL; D & T
Software Tools	SL; T
System Programs in Modification	SL; SD & T
System Programs	SL
System and Application Audit Trail	AM; & Appropriate Category(ies)

Figure 6. File Access Class Assignment

The overall effect of the configuration of access classes described above is shown in Figure 7. Their correspondence to the set of requirements stated above is quite close. There are limitations, however. First,

*Programmer support software—compilers, linkers, library managers, etc.

the scheme introduces an apparently all-powerful system control function. In fact, the system control population has only the task of moving tested programs and data from category to category and thus could be limited to a specific set of programs at login (e.g., by the discretionary control mechanism). Furthermore, there is a provision for auditing system control actions. Nonetheless, this facet is worrisome.

A second limitation of the scheme deals with its treatment of system programmers. It is not clear that limiting system program development and modification to an application-like environment with someone else (system control) doing installations is either technically or culturally acceptable. In fact, a system programmer who decided to eliminate the *-property altogether for a set of processes of his choice (or to take some equally drastic covert action) could probably do so. Nevertheless, the scheme presented attempts to limit system programmers to their authorized domain, and this is a desired effect.

On the positive side, Figure 7 does reflect a configuration that meets the stated requirements. Programmers, users, and system programmers are each limited to their own sphere of activity. There is an audit trail for management, and management can observe but not change the state of the system. In summary, this application of the lattice model to a commercial environment seems successful if unconventional. The next section introduces an alternative mechanism for achieving the effects presented in Figure 7.

OBJECTS SUBJECTS	Prod. Data	Prod. Code	Dev. App. Prgm.	Dev. Sys. Prgm.	Tools	Sys. Prg.	Audit Trail
System Mgt. & Audit	R	R	R	R	R	R	RW
Production Users	RW	R				R	W
Application Programmers			RW		R	R	W
System Programmers				RW	R	R	W
System Control	RW	RW	RW	RW	RW	RW	W

Figure 7. Effects of the Commercial Lattice

THE INTEGRITY MODEL AND THE COMMERCIAL LATTICE

The Integrity Model

In addition to the basic (security) lattice model described above, there is another model directed toward the control of modification (rather than disclosure) of information. This model is the "integrity" model [Biba], which may be considered the mathematical "dual" of the security lattice model. The integrity model seems well-suited to meeting commercial security requirements of the sort outlined above.

The integrity model, like the security lattice model, assigns levels and categories to information and to users. Its objective, however, is to prevent the contamination of high-integrity (highly reliable) information by the infusion of lower-integrity data and by processing with lower integrity programs. The analog of the security policy restriction on a process of high security access class writing information of a lower access class (the *-property) is a restriction on a process of high integrity access class reading (or executing) an object of lower integrity access class. This policy (Figure 8a) assures that computations performed by a process of high integrity access class maintain their high integrity, uncorrupted by low integrity information. Like the security *-property, it is conservative policy, preventing all reading by a high-integrity process of low-integrity information, even though the information (or the process' use of it) may be harmless.

A process P may read an object O if

$$L_p \leq L_o \ \& \ \{C\}_p \subseteq \{C\}_o$$

(a) Integrity *-Property

A process P may write an object O if

$$L_p \geq L_o \ \& \ \{C\}_p \supseteq \{C\}_o$$

(b) Simple Integrity Condition

Figure 8. Integrity Lattice Policy

The integrity analog of the simple security condition and its restriction on a process reading information of a higher access class is the "simple integrity condition." This condition prevents a process (that has read or executed information) of low integrity from writing information of high integrity. Unlike the integrity *-property which restricts a high-integrity process on the basis of conservatism (it may make a mistake or be subtly influenced by flawed data), the simple integrity condition restricts a low-integrity process from corrupting information of higher integrity—which it is simply not allowed to access. A formal statement of the simple integrity condition is given in Figure 8b.

While the integrity model has been in existence for several years, literature on its application is sparse. The discussion below integrates the integrity policy into a commercial processing application based on those described in the previous section.

A Commercial Application of the Integrity Model

The integrity model can be used in an attempt to simplify and render more intuitive a lattice policy application that meets the commercial security requirements described above. The formulation described below will meet an additional requirement:

(6) Special-purpose application software shall be provided to effect "data base repair" on the production application programmer or system control population under special circumstances.

A set of security and integrity levels and categories appropriate to this application is shown in Figure 9. The security levels and categories that remain in this example are substantially the same as those in the previous one. The "tools" category is eliminated and replaced by the functions of the integrity lattice model. The integrity levels are provided to insure against modification of system programs (System-Program) and stabilized production and development support software (Operational). Integrity categories are used to separate the development environment from that for production (Development, Production).

Integrity Levels: System-Program > Operational > System-Low

Integrity Categories: Production, Development

Security Levels: Audit Manager > System-Low

Security Categories: Production, Development, System-Development

Figure 9. Commercial Lattice Security and Integrity Access Classes

Figures 10 and 11 show the assignment of security and integrity access classes to users and files (subjects and objects). Security access class assignments are somewhat simpler than those for the previous example; integrity access classes are used to isolate development and production environments and prevent undesired modification. The intermediate integrity level (operational) prevents computations that execute production code or software tools from modifying those objects—since no users are authorized to login at the "operational" or "system-program" integrity levels, files at these levels cannot be modified, except by system control installation. The audit and manager population operates at a low integrity access class, and can thus observe any information but modify none. The audit-manager security level prevents processes that create audit trails from observing those trails thereafter. As before, the system control population has the function of moving files among access classes.

Population	Integrity Level	Integrity Categories	Security Level	Security Categories
System Management or Audit	SL	—	AM	(ALL)
Production Users	SL	Production	SL	Production
Application Programmer	SL	Development	SL	Development
System Programmer	SL	Development	SL	System-Development
System Control*	System Program	Production, Development	SL	Production, Development
Repair	SL	Production	SL	Production

Figure 10. Users' Security and Integrity Access Classes

Object	Integrity Level	Integrity Categories	Security Level	Security Categories
Production Data	SL	Production	SL	Production
Production Code	Operational	Production	SL	Production
Developing Code/Test Data	SL	Development	SL	Development
Software Tools	Operational	Development	SL	—
System Programs in Modification	SL	Development	SL	System Development
System Programs	Sys. Prog.	Production, Development	SL	—
System & App'n. Audit Trail	SL	—	AM	(Any)
Repair Programs	Production	Production	SL	Production

Figure 11. Files, Security and Integrity Access Classes

OBJECTS

SUBJECTS	Production Data	Production Code	Develop. Code & Test Data	Develop. Sys. Prog.	S/W Tools	Sys. Prog.	Re-pair Code	Audit Data
System Mgr.	R	R	R	R	R	R	R	RW
Prod. User	RW	R				R		W
App'n. Prog.			RW		R	R		W
Sys. Program				RW	R	R		W
Sys. Control	RW	RW	RW	RW	RW	RW	RW	W
Repair	RW	R				R	R	W

Figure 12. Effects of Commercial Lattice Model with Integrity

The effects of the new lattice model formulation are depicted in Figure 12. They are similar to those

*Plus "downgrade" privilege

shown in Figure 7. The "repair" function introduced in response to an additional requirement appears identical to the other production programs—and the individuals who execute it have rights indistinguishable (at the non-discretionary control level) from those of production users. This result means that repair code must be protected by a discretionary controls mechanism rather than the non-discretionary controls. The reason for this is that any attempt to introduce a "repair" security or integrity category either prevents the repair process from reading the data to be repaired (integrity category) or from rewriting it (security category). Perhaps the best approach is to use a repair security category, rewrite the data base with repair and production security categories, and then allow "system control" to remove the repair category from the newly-repaired data base. This form of "two-person" control (repair plus system control) may be appropriate to an exceptional operation like data base repair.

The idea of two-person control for the repair function may also apply in the case of an urgent repair where all rules go "out the window." If a critical application or data base is down, an organization may be willing to abandon its nominal security policy and tell the responsible system or application programmer to "fix it!" If the installation has a lattice security policy enforced by its operating system, the programmer may be given the highest security and integrity levels, every category, and downgrade privilege. In such a case, the remaining security rests on the personal integrity of the individual programmer. If an emergency repair is needed, it may better be made by a team of a system programmer, an application programmer and a system control "installer." Such a team can operate within the context of the lattice model controls, and is probably both a more secure and more effective (by virtue of an element of cross-checking) way of accomplishing emergency system repairs.

Another effect of the integrity formulation is to almost eliminate the restriction on users logging in at access classes less than their "full clearance." The exception is again in the area of "repair" processes. An application programmer with development integrity and security categories (the normal case) *and* production integrity and security categories issued to support a repair role could observe and modify production data bases in a way that would normally be unauthorized. Such a programmer could not modify the production or repair code because of the absence of the "operational" integrity level. Furthermore, it is not clear what tools such a programmer would have available to support his efforts since the development tools do not have production integrity category and no software they produce will either. Nonetheless, repair personnel should be required to login either in a repair role (production categories) or a development role (development categories). More generally, it appears that some limitation on a user's selection of login levels and categories is a desirable feature of systems that implement the lattice model.

The integrity lattice formulation, unlike that using only the security model, limits the possibility of introducing a "Trojan Horse" to modify a sensitive data base. A development programmer can write a "system-low" program to manipulate production data (though it will actually appear from the compiler with development categories), but only the system control "promotion" process can give it the integrity level to operate on a production data base. Compromise of production data to a development programmer is prevented by the *-property and the "production" category in both security and integrity lattices.

With the exceptions and caveats noted above, the integrity and security lattice formulations both meet the stated requirements. The integrity formulation offers somewhat better control, but it is not clear that users and security personnel will find the set of levels and categories required intuitively understandable.

REVIEW AND COMMENT

The early paragraphs of this article "threw out" the idea of partitioning a commercial system by levels and categories, based on the claim that there is no notion of clearance in most commercial organizations. Revisiting this assumption, it may be observed that some commercial organizations do have organizational security partitions corresponding to the roles of components (manufacturing vs. engineering) or special activities (a critical new product). These partitions may be reflected by security categories of the lattice model, and overlaid on the sort of lattice formulation described above. Thus a data base and production code might have security categories "Production" and "Project-XYZ-Engineering." The application programming function might likewise be divided into (Project-XYZ-Engineering, Development versus Project-ABC-Engineering, Development) and application programmers either isolated or (in a smaller organization) allowed to login with the security category corresponding to the function they were

supporting.* This application of the security lattice model might even be extended back to the DoD environment and DoD security levels incorporated as well. It is probable that the reintroduction of security levels and categories is most compatible with a formulation that applies the integrity lattice to control writing, and thus reduces the proliferation of security categories.

Whether the formulation of choice is security lattice only or security lattice plus integrity lattice, it appears that the lattice model can help address real commercial computer security requirements. This result may be of interest both to individuals and organizations that have such requirements and to developers and advocates of systems that incorporate the lattice model.

ACKNOWLEDGEMENT

The ideas presented here were heavily influenced by initial discussions with Bill Hill, Bruce Parker, Ashby Woolf and Tom Bailey. Paul Karger, Ed Balkovich, Maurice Wilkes, and Joe Tardo reviewed the draft, and Pat Bright prepared the paper in final form. The effort reported was supported by the Digital Equipment Corporation's Corporate Research and Government Systems Group.

REFERENCES

[Biba] Biba, K. J., "Integrity Considerations for Security Computer Systems," ESD-TR-76-372, Electronic Systems Division, AFSC, Hanscom AFB, MA, April 1977, (AD A039324).

[BlaP] Bell, D. E., and LaPadula, L. J., "Secure Computer Systems." ESD-TR-73-278, Volume I-III, The MITRE Corporation, Bedford, MA, November 1972-June 1974.

[MME] Heitmeyer, Constance L., and Wilson, Stanley H., "Military Message Systems: Current Status and Future Directions," *IEEE Transactions on Communications.* Vol. Com-28, No. 9, Sept. 1980, pp. 1645-1654.

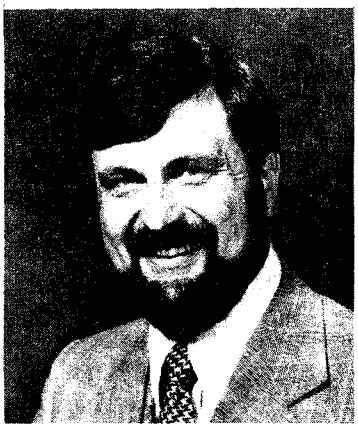
[Multics] Whitmore, J. C., et al., "Design for Multics Security Enhancements." ESD-TR-74-176, Honeywell Information Systems, 1974, (AD A030801).

[Nibaldi] Nibaldi, G. H., "Proposed Technical Evaluation Criteria for Trusted Computer Systems," M79-225, The MITRE Corporation, Bedford, MA, Oct. 25, 1979.

[SACDIN] Chandersekaran, C. S. and Shankar, K. S., "Towards Formally Specifying Communications Switches," *Trends and Applications 1976: Computer Networks,* IEEE, New York, N.Y., Nov. 17, 1976, pp. 104-112. □

*In order to partition files, activities and audit trails, though not to isolate information from the programmers themselves.

COMPUTER SECURITY AND INTEGRITY TECHNOLOGY



Dr. Dennis K. Branstad
Manager
Integrity, Security, and Data I/O Group
Institute for Computer Science and Technology
National Bureau of Standards

Dennis received a Ph.D. in Computer Science, and has taught at the University of Maryland and at George Washington University. Before coming to NBS in 1973, he was active in computer and speech research with the Department of Defense, and has been involved in many computer security activities in the public and private sectors. Dennis was instrumental in the development of the Federal Data Encryption Standard, and is Chairman of the ANSI standards group which is developing standards for using encryption in various security and integrity applications.

INTRODUCTION

Today I would like to spend a few minutes with you discussing the issues of computer security and integrity. I have chosen to differentiate between integrity and security because of the different emphasis given these issues by some organizations. Integrity is the protection of data from unauthorized modification. Security, for the purposes of this discussion, means protection of data from unauthorized disclosure. This is somewhat of an artificial differentiation and the two issues are usually combined in most discussions of computer security. However, we at the Institute for Computer Sciences and Technology have identified a significant set of requirements for integrity in areas where there is little or no desire for secrecy of data.

THE COUNTY LIBRARY CAPER

As an example of such requirements, Mr. Burrows commented on an incident which happened in Montgomery County, Maryland. The incident regarded the Montgomery County Library System. The incident was reported in a local newspaper and I would like to discuss some of the lessons to be learned, especially when we talk about trusted computer systems, reliable systems and systems having integrity. I would like to reiterate some of the details of the incident and challenge the speakers this afternoon and the DoD Computer Security Center people speaking tomorrow to respond to some of these issues and open the scope of trusted systems to include them.

The Montgomery County Library is really a distributed system of libraries which have a nice service feature. You can get a book from any library and return it to any other library in the county. All of the libraries are now connected to a central computer system. About a year ago, as reported in the article that I mentioned, the county purchased a library management system for about one million dollars. The county has not been happy with the reliability of the system. Books have been lost, customers have been delayed and money has been spent in searching for the problem. Because of the failure to meet the reliability specifications of the contract, the county withheld final payment for the system. According to the article, the software company "hinted" that the computer would fail unless the final payment was made. The county computer experts looking for the causes of the reliability problems found code in the system that would cause the system to fail on the 15th of March. The county removed the code but telegrammed the company that if the computer failed again, the company would be held liable. On March 13th, two days before it was programmed to fail, the company sent to the county a program "fix" which removed the offensive code. The liability issue is still under investigation.

TRUSTED COMPUTER SYSTEMS

This incident could be considered sabotage. It is outside the Department of Defense. It is in a library system. No risk analysis would have given a high probability to the possible occurrence of this incident. However, it is a real issue in the area of trusted computer systems.

What are some of the issues of trusted computer systems? How can a user be sure that software can be trusted? How widespread is the need for trusted systems? Trusted software is the theme of the DoD Computer Security Initiative, and as we have heard, is the major emphasis of the new DoD Computer Security Center. We at the National Bureau of Standards are working with the DoD in identifying requirements for integrity and security in the area of trusted systems but outside of the DoD. The incident that I just outlined is obviously one of those cases. The technology being developed in trusted computer systems can and should be applied in many areas if the technology is well defined and cost effective.

Other issues of liability may arise. In the commercial world, if a user decides not to pay a bill, the user could plant a "bomb" of this nature in a computer system (or other product such as an automobile), claim that the vendor did it, and refuse to pay for the product or to even sue for damages. It is difficult to establish liability in the software area. Acceptable liability, contributory negligence and other legal issues may arise in the area of trusted systems as the concept matures.

ICST SECURITY AND INTEGRITY PROGRAM

I would like to spend a few minutes outlining the highlights and near-term goals of the ICST integrity and security technology program. The goals of this program are to provide the necessary standards and guidelines for implementing and using technology which would alleviate a wide range of security and integrity problems. The program is divided into a number of technical areas. You may contact me if you have a special interest in one of these areas.

Personal Identification

The first requirement for both integrity and security is personal identification/authentication. How can a computer system identify the users of the system? Additionally, it may be necessary to identify the designers, the implementers, the maintainers and the operators of the system in order to maintain an audit trail of all the people that contributed to building the system or could have sabotaged the system. Over the past few years, we have established a laboratory for personal identification in ICST. At various times, we have investigated fingerprint readers, handwritten signature readers, hand geometry readers and palmprint readers. These have met with limited commercial success. While the technology is often fairly well developed in these areas, the cost of the devices and the difficulty of use typically make them undesirable in large, distributed computer systems. For example, the prototype palmprint reader is quite interesting, quite successful in our limited experience with it, but is presently quite expensive and not practical for identification of a computer user at a remote terminal. We are also looking at voice verification techniques in our Computer Integrity, Security and Speech Laboratory.

The primary method of personal identification, passwords, is one that we all are living with, will continue to live with and is still the most cost-effective method of personal identification from remote terminals. We have recently completed all the preliminary processing of a proposed standard on password usage. It was published in 1981 for comment. The standard specifies ten factors that must be considered when designing and implementing a password system, and defines minimum security criteria for each of the ten factors that must be met in Federal applications. The standard is scheduled for release in late 1982.

Data Integrity Standard

The subject of data integrity is very important. As I previously mentioned, integrity is the assurance that data has not been modified, either accidentally or intentionally, without authorization. We are planning to develop and implement various methods of assuring data integrity in the Computer Integrity, Security and Speech I/O Laboratory we are establishing within the Institute for Computer Sciences and Technology. Integrity is an especially important area in the financial community. Blake Greenlee will dwell on the requirements for integrity in the financial community following my talk. There are some requirements for secrecy of information but many requirements for protecting data from modification, replacement or replay.

ICST has developed a proposed Data Integrity Standard for providing this protection. This standard can be used to put a seal on data so that it cannot be modified without being detected.

The Data Integrity Standard is based on using cryptographic techniques (the Data Encryption Standard in this case) to compute a number which depends on the contents and order of the data. This number is called an integrity code, or an authentication code, or a cryptographic check field. It is computed using a secret key. The integrity code is attached to the data in some way, for example, appended to the end of the data file or the message. When the integrity of the data is to be verified, for example after a message has been received or a stored data file retrieved for processing, a second integrity code is computed on the received or retrieved data. If the second code is the same as the first code, then the data has not been modified.

Roger Schell spoke yesterday of the need within the Department of Defense to be able to mark data with its classification. He spoke of coloring the "bits" of data either red or blue to denote their classification. The Data Integrity Standard can be used to do this. A "red" key would be used to compute the classification stamp for "red" data and a "blue" key would be used to compute the stamp for "blue" data. Dissemination could be controlled by verifying the color of the stamp on data before it is distributed. Many "colors" of keys could be defined to provide any type of access control granularity that one would desire.

We have implemented the proposed Data Integrity Standard at NBS, both in software and in a hybrid, hardware-software system called the Key Notarization System. The capabilities of the system include identifying users via passwords, encrypting and notarizing passwords, encrypting data, decrypting data, generating and notarizing keys, updating stored keys to be protected under a new key, generating integrity codes and verifying integrity codes. There are over fifty commands in the complete system, but the user is provided an interface which makes the system simple to use. The major emphasis of this work is in providing secure, person-to-person mail (i.e., file transfer), assuring data and program file integrity and automated key management.

Public Key Cryptography

The third major part of our integrity and security technology efforts is the area of public key cryptographic algorithms. These are cryptographic systems in which the enciphering key can be made public so that anyone can encipher (lock) data intended for someone but only the intended recipient can decipher (unlock) the data. We have been looking at these algorithms for several years. We would like to initiate the establishment of a standard in this area. However, one of the factors in standards establishment is the timing of the standard. Technology and user demand must come together so that the best standard can be found. We feel that the time is right in this area to get started.

NBS desired a public key algorithm to be used in conjunction with other security methods to protect electronic mail and electronic money in systems in which pre-arranged Data Encryption Standard keys were not available. In most existing applications in these areas, contractual arrangements have been made which spell out the services to be performed, the financial arrangements for the services, and also the keys to be used for protection. Our present standards work very nicely for that particular type of protection. However, if I would like to get some sensitive information from someone that I had never met or worked with before, and on a one-time basis, I would like to use a public key algorithm to achieve some level of protection. So with examples like this in mind, and based on a letter from the finanacial industry to take the lead in this area, we are preparing to initiate a standard in this area. We have prepared a solicitation for public key cryptographic algorithms which will be published in the FEDERAL REGISTER within the next few weeks (*Note:* it was published on June 30, 1982.)

Open Systems Interconnection (Network) Security

The last area that I would like to discuss today is our security work with the Open Systems Interconnection model of the International Organization of Standardization, commonly called the ISO–OSI. This is a conceptual architecture for the standards required to interconnect information systems. It consists of seven layers of protocol by which an application process in one system can exchange information with another application process in the same or a geographically remote system.

We have been looking at integrity and security in that model. The primary requirements include personal (or process) identification of the user associated with each process and a cryptographic based integrity/

security system. Five categories of protection have been defined (not to be confused with the categories defined by the Computer Security Center of the DoD). Category A is protection of data against disclosure (i.e., secrecy). Category B is protection against modification (i.e., integrity). Category C is protection against both disclosure and modification (i.e., both A and B). Category D is protection against modification, insertion, deletion and replay. Replay is the recording of a valid message (e.g., deposit of $100 into a bank account) and then subsequent replay of the message to an unsuspecting receiver. Category E includes all protection (i.e., categories A through D).

At the present time, the American National Standards Institute Technical Committee on Encryption (X3T1) is preparing the standards necessary to provide these categories of protection at three layers of the seven-layer ISO–OSI model. Various benefits are derived by protecting at each of the three layers (layers 2, 4 and 6 have been selected). The layer 2 standard is complete while the layers 4 and 6 standards are still in development.

SUMMARY

The other areas of interest with the integrity and security technology program are secure voice input/output, software verification, trusted software, network access controls and user authorization. We presently do not have the necessary resources (people and money) to address these areas of needed security adequately. We look to the Department of Defense and the DoD Computer Security Center to provide the needed software verification tools and trusted software specifications. We are happy to work with the people in these organizations in providing their results to a wide audience. We hope that this conference will be successful in continuing to provide this information to you. We hope to hold several conferences in the future as more information is developed and is available. □

THE COMPUTER SECURITY AND RISK MANAGEMENT PROGRAM

Dr. Stuart Katzke
Leader
Security, Management, and Evaluation Group
Institute for Computer Science and Technology
National Bureau of Standards

Stuart received a Ph.D. in Computing and Information Science from Case Western Reserve University, Cleveland, OH. Prior to joining NBS, he was on the faculty of the College of William and Mary, Williamsburg, VA. Dr. Katzke is responsible for developing Federal Information Processing Standards (FIPS) and guidelines for use by Federal agencies in evaluating and managing computer security. His specific areas of responsibility include risk analysis, contingency planning, security evaluation, user access authorization, and security program development. He is co-author of FIPS 73, "Guidelines for Security of Computer Applications," and co-author of the "Executive Guide for ADP Contingency Planning."

INTRODUCTION

This afternoon I would like to define the scope of the computer security and risk management (CSRM) problem and then discuss the Federal Government's responsibility for computer security and risk management; that is from a federal agency or organization point of view. I think you will find that these responsibilities are similar to those of private organizations and of the DoD. Following that, I will discuss ICST's risk management program activities including the work we are doing to provide guidance for managers. Dennis Branstad has already discussed computer security technology activities.

I thought I would ease into the scope of the computer security and risk management problem by giving you a definition of computer security. I have, and I am sure you have, seen numerous definitions of computer security. One that I was going to refer to was on a slide used by Steve Walker this morning. I call it "Definition by Exhaustive Inclusion." He had a slide entitled "computer security" and on that slide was listed "physical security," "administrative security," "data security," etc. If you look each one of those up, of course, you have a definition of computer security.

One definition that we have used in some of our documents is similar to a mathematical theorem: "computer security is the state that exists when the following conditions hold true." Then there is a long set of conditions. If you satisfy those conditions, you have computer security. However, for this audience, I thought we needed a highly technical definition that definitively bounds the computer security area. That definition is: "All the bad things that can happen when you use or depend on a computer."

So, with that in mind, we have a binary definition: if something bad happens, it is a computer security problem; if not, it is not.

SCOPE OF THE COMPUTER SECURITY AND RISK MANAGEMENT PROBLEM

Computer security is concerned with the potential losses related to the use of ADP resources and services. Security proponents are in the unfortunate position of asking management for resources to prevent events that may or may not occur. As you are well aware, management wants to see expenditures related to profits or some other tangible benefits. When a security program is effective, it is often difficult to demonstrate this fact, or worse, the program is cut back.

The emphasis here is on potential loss. There is a joke I recall that presents an analogy with the above situation. I will explain the analogy after the joke. It is an old elephant joke and goes something like this:

> A man is sitting in the center of a big city banging two sticks together. Somebody else is watching him; has been watching him for a long time. Finally, curiosity gets the better of the watcher and he goes over and says, "I don't understand what you are

doing. I have been watching you banging these two sticks together. What for?" The first man says, "Well, I was told that if I just keep banging two sticks together, it keeps the elephants away." The watcher says, "There are no elephants around here." And the other fellow looks back and says, "See, it's working."

Well, if you substitute a computer security program for the banging of the two sticks, and security violations for the elephant, then you have the analogy.

When we talk about potential losses, we are concerned about dollar or material losses and social inconveniences, such as privacy violations, loss of lives, benefit checks that are not received, or underpayments in benefit checks. One doesn't usually think of overpayments as being a social inconvenience. If you are an ethical person and you would like to try to return that money, though, the frustration of having to return money to an agency that is not equipped to accept cash from people might be an inconvenience of sorts. Failure to meet missions or legislative requirements is another type of loss, very often, reflected in the next year's budget. Losses result from disclosure of proprietary or sensitive information. For example, some federal systems maintain proprietary private industry data which, if disclosed, can result in a loss of competitive advantage for some organization. Embarrassment and loss of public confidence are consequences that we in the government are very sensitive to. And, of course, losses result from legal actions.

The reasons for these losses are:

- Unavailability of ADP processing, usually resulting in critical functions not being performed or being performed in an untimely way.
- Incorrect performance of intended functions or the performance of unintended functions. These often result from software flaws due to poor requirements definition or design flaws.
- Accidental or intentional events resulting in destruction, modification, disclosure, or misuse of data, software and hardware. Accidents, errors, or omissions fall into this category, as well as computer related crime, natural hazards, and a variety of other events.

Finally, the scope of the computer security problem deals with the selection and implementation of safeguards. This typically includes prevention of loss by eliminating threats or reducing vulnerabilities. If for some reason you choose not to prevent losses, based on financial or other reasons, you then try to detect security variances. And even if you have an appropriate mix of prevention and detection activities, something always seems to go wrong. Consequently you must be able to recover or have some contingency plans.

GOVERNMENT RESPONSIBILITY FOR CSRM

The federal agencies' responsibilities derive from the following chain of events. First of all, they, or any organization for that matter, have mission requirements and responsibilities. These generally derive from legislative or legal requirements, executive orders, regulations, standards of good practice, and ethical and social issues. To meet responsibilities and satisfy requirements, management, at all levels within an organization, must maintain control over their piece of the organization. This means they must anticipate problems, propose solutions, plan for contingencies, and so forth. The traditional method for doing this is to practice risk management. Risks to an organization occur for many reasons. There can be labor problems due to strikes, financial uncertainties due to funding or market conditions, procured goods not delivered on schedule, etc. But the kinds of risks that this audience is most interested in are those that are related to the use of ADP resources and services. One has to assess those risks, determine what level of risk one can live with, and then implement safeguards and controls. Safeguards and controls generally fall into management and technical categories. Management safeguards are considered more or less near term because they can generally be put into place faster. Technical safeguards tend to be longer term, especially when they require system redesign or development.

Federal agencies, in meeting their responsibility, must develop comprehensive computer security risk and management programs which extend through all levels of their organizations. If no program exists, one must be established; once established, it must be maintained. Generally, maintenace is performed by a computer security program office which assists all levels of management throughout the agency.

ICST's CSRM Program

The Institute for Computer Sciences and Technology primarily provides assistance to federal agencies in meeting their computer security responsibilities. However, during the past years our program has been extending to private industry as well. We find that we are interacting with private industry with increased frequency and that they are making ever increasing use of our products.

Our objectives are to reduce ADP-related security risks to federal agencies, private industry, and private citizens. With respect to federal agencies, we provide technical assistance to them in protecting their ADP resources and services. There are a number of forms this assistance can take. First of all, we can provide direct assistance on a cost reimbursable basis, as well as informal consultation, phone calls, briefings, meetings, and various other services. Second, we assist federal agencies through our publication series. Denny has already mentioned the Federal Information Processing Standards Publications that include guidelines and standards. We also issue other publications, guides, and reports on our standards development and research activities. And, finally, we can conduct research and development in support of our other activities.

I mentioned that we assist both private industry and private citizens. This is more of an indirect type of activity. Because of the large amounts of proprietary and personal data maintained on private citizens and private corporations by the federal agencies, helping the federal agencies protect their ADP systems indirectly protects the interests of these other groups.

The clients for our services and products are federal information managers and users, which include system security officers, data base administrators, auditors, data processing installation managers, among others. Our audience also includes private industry information managers and users, and we welcome their views on our activites. We find that private industry uses our guidance documents, participates in the review and development of our products, and often adopts our standards. We are particularly interested in comments from ADP industry vendors of systems and services because they are often affected by the standards and guidelines we produce.

I will not go into the program areas right now. Denny has discussed some of them and I will come back to others at the end of my presentation.

Often we are asked what criteria we use to determine the program areas we work in. Before you is a list of items that we have to consider in making programmatic decisions. First of all we must consider the demands of our constituency. What is it they need most? Also, we have to look at the benefits, impacts and costs involved in developing various products. Costs include our development and maintenance costs and the user's implementation costs. We are trying to establish a framework for the computer security area so we can improve our evaluations of the costs, benefits, and impacts of various standards and guidelines. Marco Fiorello, who will be speaking in a few minutes, will discuss his efforts on our behalf in that area. Other criteria include resources for the computer security program, the technology, current accepted practices, and legal issues. Also, we have to take into consideration related efforts in voluntary standards development organizations.

Our program activities include:

(1) Identifying our constituencies' requirements (i.e., what are the needs of the federal agencies). We do this through personal contacts, conferences, workshops, meetings and constituency projects (where we provide direct technical assistance to the agency). Senior management officials for federal ADP standards, appointed by agency heads, help us identify standards and guidelines needs.

(2) Identifying best practices and methods that can be used to satisfy our constituencies' requirements. We use technical assessments, conferences, workshops, and other means.

(3) Developing a plan that schedules needed products for the federal agencies to use. When developing products, we try to examine existing practices and methods. Where these are not adequate, we may try to develop new practices and methods that seem feasible, and publish the results in various technical documents.

(4) Reviewing previous products. We are currently reviewing the Data Encryption Standard which has been in effect for five years to determine if it is still up-to-date and needed.

(5) Publicizing our activities by making the general public and federal agencies aware of the work we have done.

Let me mention that we do have a security publications list. If any of you would like one, please contact me.

Let me address some of our relationships with other organizations.

- We have worked with the American National Standards Institute on the data encryption and financial transaction committees.
- With respect to federal agencies, we have been technical consultants to the Department of Energy; we are working with DoD in sponsoring these conferences; we have worked with GAO in co-sponsoring two workshops which initiated some of the work that we have done in the security audit and evaluation area; and we are working with FADPUG by co-sponsoring a conference in early June.
- We also have been expanding our interactions with state and local governments. We would like to find out what, if anything, we have to do to our technical products to make them usable to these organizations. Also, we would like to learn of their activities. Some of the state and local governments have done an excellent job in establishing standards and practices. We are trying to foster technical interaction between them and to provide them with technical assistance. For your information, the three areas which are of most concern to the state and local governments are: computer security, networking, and data base management systems.
- Concerning our interactions with private organizations, we have had a long time association with the American Bankers Association, especially in the encryption and integrity areas. We have invited ADP vendors groups to our workshops to help us review some of our products and documents. Also, we interact with them during their long-range planning activites, especially in trying to predict what federal agency needs might be in the future. Private corporations, as I mentioned, use our documents and participate in our workshops. But, we also try to visit those organizations that have good security programs in order to find out what they are doing right and to learn from their experiences. In one of our visits to Carter, Hawley & Hale, we met John Pricz. We were so impressed by the security program of his organization that we invited him to come and talk to us today.

ICST Security Management Areas

Let me quickly discuss the mangement areas that we are working in, or have worked in.

- We have published a FIPS guideline on risk analysis. It describes a particular methodology which was chosen because it has been successfully used in private industry and forms the basis for many other methodologies.
- Our effort in the evaluation and certification area was started with two ICST/GAO workshops. I won't say anymore about that effort because Zella Ruthberg, who is project manager in that area, will be talking to you about it. But I will mention that one of the future activities in that area is to look at the selection and use of evaluation tools. This would include those tools used by the audit community, as well as those used by Mr. Tippett's group in the Security Evaluation Center.
- In the contingency planning area, we have already published a FIPS guideline and have followed that up with an Executive Guide. The Executive Guide is a brochure that is aimed at high-level management; takes no more than 15 minutes to read (since it is in a question-answer format); and is intended to convince high-level management of the need for contingency planning. One of the issues we are looking at in this area is the selection of an ADP backup strategy. As you know, there are many alternative strategies one can consider. Since some of them are quite new, there is not much of an experience base. These strategies include contingency centers, empty shells, reciprocal agreements, geographically separated processing centers, and shared load between processing centers. We would like to find out what people's experiences have been, particularly in these newer areas, and to develop some guidance to give federal agencies in making their selections.
- I mentioned that part of the federal agencies' responsibilities was to develop or set up a computer security program. We are working on a guideline in that area.
- We are also working on providing guidance in the security variance detection area.
- Some of our future plans include looking at the security of microcomputers. We have established a microcomputer lab. As part of that microcomputer lab, we are going to be looking at the security capabilities of microcomputers, or as you probably guessed, the lack of security features of many of the

microcomputers. We want to look at developing security enhancements to the capabilities of microcomputers and using microcomputers for performing the security functions for other systems or as components of microcomputer-based systems.

In conclusion, I would like to emphasize that we are here to interact with and assist you. I encourage you to take advantage of our products and services. □

PRESENTATION TOPICS

1. Scope of the Computer Security and Risk Management (CSRM) Problem

2. Government Responsibility for CSRM

3. ICST's CSRM Program
   - Management
   - Technical

TECHNICAL DEFINITION OF THE COMPUTER SECURITY PROBLEM:

''ALL THE BAD THINGS THAT CAN HAPPEN WHEN YOU USE OR DEPEND ON A COMPUTER''

SCOPE OF THE CSRM PROBLEM

- Potential Losses Related to the Use of ADP Resources and Services
  - $/Material
  - Social Inconvenience
  - Failure to Meet Mission or Legislative Requirements
  - Disclosure of Proprietary or Sensitive Information
  - Embarrassment/Public Confidence
  - Legal Actions

- Reasons For Losses
  - Unavailability of ADP Processing
  - Incorrect Performance of Intended Functions
  - Performance of Unintended Functions
  - Accidental/Intentional Events Resulting in Destruction, Modification, Disclosure or Misuse of Data, Software and Hardware

- Selection and Implementation of Safeguards
  - Prevention
  - Detection
  - Recovery

GOVERNMENT RESPONSIBILITY FOR CSRM

- Derivation of Responsibility
  - Mission Requirements and Responsibilities
  - Management Control
  - Risk Management
  - ADP Related Risk Management
  - Safeguard and Controls
  - Management (Near Term)
  - Technical (Long Term)

- Meeting the Responsibility
  - Establish a Comprehensive CSRM Program
  - Maintain On-Going Program

ICST's CSRM PROGRAM

- Objectives
  - Reduce ADP Related Security Risks to Federal Agencies, Private Industry, and Private Citizens
  - Provide Assistance to Federal Agencies in Protecting ADP Resources/Services

- Audience
  - Federal Information Managers/Users
  - Private Industry Information Managers/Users
  - ADP Industry Vendors (Systems and Services)

- Program Areas
  - User Identification/Authentication/Authorization
  - Data Communications, Storage and Integrity Protection Using Encryption Techniques
  - Security Variance Detection
  - Risk Analysis
  - Security Program Development
  - Security Evaluation/Certification
  - Contingency Planning
  - Security of Computer Applications
  - Secure Voice I/O
  - Physical Security

- Program Decision Criteria
  - Constituency Demand
  - Benefits
  - Costs
  - NBS Resources
  - Technology
  - Current Accepted Practice
  - Legal Issues
  - ISO/ANSI Related Efforts

- Program Activities
  - Identify Constituency Requirements
  - Identify Best Practices/Methods
  - Develop Products
  - Review Products

- Relationships With Federal/State and Local/National Organizations
  - ANSI Technical Committees
  - Federal Agencies
  - State and Local Governments
  - Private Organizations

MANAGEMENT

- Risk Analysis
  - — Guideline on Methodology (Completed; FIPS 65)

- Evaluation/Certification
  - — Guideline on Evaluation (On-Going)
  - — Guideline on Certification (Current)
  - — Guideline on Selection and Use of Evaluation Tools (Future)

- Contingency Planning
  - — Guideline (Completed; FIPS 87)
  - — Guideline on Selecting an ADP Backup Strategy (On-Going)
  - — Executive Guide on Contingency Planning (Completed)

- Program Development
  - — Guideline (On-Going)

- Variance Detection
  - — Guideline (On-Going)

Note Definitions:

Completed:	Publication Available to General Public
Current:	Work in Progress; Completion Planned For Current Fiscal Year
On-Going:	Work in Progress; Will Not Be Completed in Current Fiscal Year

TECHNICAL CONTROLS

- User Identification and Authorization
  - — Password Use Standard (Current)
  - — Personal Identification Guideline (Future)
  - — User Access Authorization Guideline (On-Going)

- Application Life Cycle Guideline (Completed; FIPS 73)

- Physical Security Guideline (Future)

- Data Communications Protection
  - — Cryptographic (Encryption) Algorithm Standard (Completed; FIPS 46)
  - — Guideline for Implementing and Using DES (Completed; FIPS 74)
  - — Modes of Operation Standard (Completed; FIPS 81)
  - — Encrypted Communications Standard (On-Going) Levels 2, 4, and 6 of ISO Model

- Data Storage Protection
  - — Key Notarization System (KNS) Development (Completed)
  - — Transportable KNS Development (Current)
  - — File Encryption Standard (On-Going)

- Network Access Control
  - — Network User Authorization System (Future)

- Data Integrity Protection
  - — Data Authentication Standard (Drafted)
  - — Financial Message Authentication Standard (Drafted)

- Secure Data Input/Output
  - — Guidelines for Secure Voice Input/Output (Future)

FINANCIAL (BANKING) VIEW OF COMPUTER SECURITY

M. Blake Greenlee
Vice President
Citibank, NY

Blake received a B.S. in Physics and Mathematics from Purdue University, has done advanced work in those subjects at Purdue and the University of Maryland, received his M.B.A. in Finance and Administration from George Washington University, and completed a three-year program at The General Theological Seminary. Prior to joining Citibank, Blake was Technical Director of TTI (a subsidiary of Citibank); with MITRE, where he served as a consultant to JCS, USIA, and the IRS; and with Johns Hopkins Applied Physics Laboratory, where as a program manager he had responsibility for a variety of programs including the Polaris satellite navigation system and the production procurement of Navy Navigation Satellites. Currently, he is responsible for the communications security of Citibank's worldwide network, and chairs the ANSI working groups on Financial Institution Message Authentication and Key Management. He represents the U.S. as a technical expert on those subjects on ISO working groups. Previous assignments at Citibank include development of long-range plans for distributed processing; development of corporate standards and policies for Operations Risk Assessment and The Protection of Telecommunications; and responsibility for the privacy/transborder information flow issue, and for guidelines in auditing OS security for a variety of computers.

INTRODUCTION

Before we start out on a formal talk, I am reminded of a story which I think has an appropriate point or moral for this particular gathering. Once upon a time, a priest went lion hunting. He had been out in the bush for three or four consecutive days and just hadn't seen a thing.

On the fifth day he decided he was going to go back farther into the bush and be more adventurous. As he walked slowly through the bush, it closed in around him and he started to feel uneasy. Then he heard this crunch, crunch, crunch behind him and he looked over his shoulder. It was the biggest lion he had ever seen in his life. He panicked. He dropped his rifle and started to run. And he ran and obviously the lion could run a bit faster.

As he felt the lion's breath on his neck, he looked up toward heaven and said, "Lord, please convert this lion and make him a good Christian." Well, the breathing on the back of his neck stopped instantly. He turned around. There was the lion laying down behind him with his paws folded. As he reached down to pat the lion on the head, the lion looked up toward heaven and said, "Lord, I give thee thanks for this Thy bounty of which I am about to partake."

We have got a lot of very fine talent assembled here and people are talking about new techniques for securing the integrity of both computer systems and communications. But we can't just pray that all will go well; we are going to have to take up those tools and use them. And with that I would like to start the presentation.

SOME SECURITY RELATED ASPECTS OF WHOLESALE BANKING

I would like to talk a little bit about computer and transaction security in what I define as the wholesale banking community. Wholesale as opposed to retail. Retail—that is me going to the cash machine and taking out $50 (if that won't overdraw my account). The wholesale banking business includes the transactions between major banks and major corporations. It has some system characteristics. It tends to be distributed by process and by product.

Typically, we have a different processor for each type of banking product for each type of banking customer. Now, we don't call them data centers anymore because we don't like the word. But if you were to use that word we have somewhere between 500 and 1000 of them in New York City, and at least one or more in 112 other countries. So there is a fair amount of geographical distribution.

Our customer requirements change rapidly. We find that any system that cannot be produced and put on line for the customer in six months time is not worth developing. Put another way, if the project manager walks in with a milestone chart and the date for live operation is greater than six months out, the project is cancelled because the environment will have changed so much in that time that it won't fit customer's needs.

It mentions high transaction values on the slide. I mentioned $50 and a cash machine and that is about the average consumer transaction value. These wholesale transactions are different. Our average transaction value is $2 million. Within the United States, *each night,* we move the equivalent of the GNP. We are only one bank. There are a lot of other big banks in this country.

What happens if we have a system problem. Well, it turns out that there are penalties and liabilities for late delivery. Penalty interest is computed based on the current day's prime and works out at around $450 per million dollars that you have failed to move on time each day. Now, $450 doesn't sound like very much except that if our line to London goes down across the Federal Reserve Bank's cutoff time, the penalty interest is $500,000. That is enough to make you worry about backup and contingency planning.

There is also something called consequential damages. It is not just enough to get the money there or to pay the person the penalty interest if you fail to get it there. If the customer suffers because the money wasn't received on time, you wind up in court reimbursing him for that. That can add up to a lot of money.

We have had a couple of cases where sometimes through failures of our own, more generally through failures in international record carriers, a money transfer hasn't reached a place on time. At one point in time, there was a funds transfer to pay the crew of a ship in the harbor in Hong Kong. It didn't get there on time and the crew walked off. We then owned the cargo. What do you do with a refrigerated freighter load of rotten strawberries in Hong Kong harbor.

That is one kind of consequential damage, but it's even worse if you have to reimburse the shipper for the cost of not being able to do that business in Hong Kong anymore. That is a lot of money.

Why bring this up? One of the main reasons is because among you are represenatives of an organization called DCA. When you perceive there is a valid national emergency, you want to have all of the communication capability you can lay your hands on. I don't blame you, and nobody in the banking system is going to complain. But understand that there are difficulties involved when you want to test that capability. Because if you want to seize the lines of a banking network and it's not a national emergency, we must have both notice and some type of coordination. We, as bankers, want you to keep things in perspective when you want to make the test, because we move enough money in one day as one bank to buy a new intelligence community.

We find that the result of not being centralized (that is, being decentralized) adds a line security risk. We have become heavily dependent on telecommunications.

SECURE OPERATING SYSTEMS? THERE ARE ALTERNATIVES

Now let's talk about some heresy. When I go to line managers within Citibank or any other bank that I know of and I say, "secure operating system," they roll over and laugh and say, "It is an academic fiction, it is costly, it can't be controlled, it would inhibit our development flexibility."

Perhaps we see in their reaction a difference in the perceived anthropology for the employee-selection process. You can prescreen people and get a security clearance run, a full BI if necessary. In some agencies, a polygraph is used. We only use a polygraph when we have a defalcation on our hands and we can't figure which one of 50 employees ran away with the bacon, and then it is voluntary.

However, we do make the assumption that anybody can be tempted. We feel that people are more likely to be tempted when there is the potential for making off with one transaction's worth which is $2 million

than making off with something which is perhaps more important to the nation, some classified data for which our Russian friends will reimburse them for $10,000 or $25,000.

We have tended in decentralizing to consider that a computer is a part of the operating environment. Each operating environment. Each computer generally runs one and only one process; there is no foreground, no background. In New York City we have five 158's being replaced by 3000 series machines. They do only one thing. They take an input which is their output of the check sorting machine and they create a tape which is the sorted check output from one region of our New York customer base. It is a full-time job for the machine. But, boy, does that solve a lot of computer security problems.

So we don't need to compartmentalize, then, within the CPU and we can treat the CPU as an office tool. As every day goes by, the ability for people to take that approach gets easier and easier because the costs are dropping.

However, there is this problem called programmers and their ability to modify programs. Well we don't have any. Now that is an overstatement; we might have 20 in New York City. In 1974, we had a couple of thousand. Our programs are now done on a firm fixed price development basis. When the program is tested and works and the controls are tested and seem to work, the programmers go away. There are no system programmers. The documentation consists of no more than a functional specification, source, and object code (well commented). We find that we rarely get program documentation that is worth the paper it was written on at delivery time. It looks even worse two years later when you want to fix anything. So why pay a lot of money and exert a lot of management effort for a worthless item? (I said that this was going to be about heresy!)

However, when we do use a computer, we use every last scrap of the operating system that adds to the security of that system. We use all of the built-in features that are provided by the manufacturer. Then our EDP auditors take a look. They sit down (I have done it myself on a couple of occasions) with the book shelf and write the procedure for the line EDP auditor that will audit that system daily. What does he or she look at on what logs? Which things are forbidden to be used? How are you sure that for example, nobody compiled anything overnight, or nobody put the compiler on the system, that kind of thing? The human related controls. And we write a procedure for it and we enforce it. So we tend to concentrate one process in a box, program it and then throw away the programmers because they are not ours anymore. We get them out of the building, then nobody can tamper with the system. We put people-oriented controls around the operation. We audit to be sure those controls stay in place.

THE USE OF DUAL CONTROLS

In every system we design, unless dual controls are in place (and that means that the person who enters a financial transaction can't release it to the customer), the system cannot be placed in live operation.

I guess the best way for you to understand how deeply this is ingrained in the banking community is the next time you walk into your bank, walk up to the vault door. There are two separate combination locks. Walk inside and look at your safe deposit box. It is not a matter of being so smart and putting them in, it is a matter of understanding that there is no such thing as a trusted employee or a trusted system. Again, it is in your perception of the human condition.

LINE SECURITY: ENCRYPTION AND AUTHENTICATION

I mentioned line security. The next few slides are ones that I pulled from a presentation we made to our senior management. We were taking a look at alternative ways of protecting the data on our communication lines.

An Example of a Need for Encryption

Encryption has some advantages. You preserve confidentiality and sometimes that is really necessary. It is not necessary all of the time, by the way. Let me give you an example however, of where it is necessary. In London, at Citibank house, we have our foreign exchange traders. They buy and sell foreign currency both for that day and, using contracts, for the future.

Because they live in a wild and wooly environment with people yelling rates and so forth over the telephone and shouting to each other across the room, we have to give them some help in keeping their records straight, other than by using scraps of paper which is what they used to do. So across town in Lewisham, there is a PDP-11/70. All it is, is a simple inventory control system. It just keeps book of what the position of each trader and each currency is and the rates, and then a summary for the department.

Suppose somebody went down to Radio Shack and bought one of those $35 little microphones that are reportedly used for listening to your baby cry in the crib (if you want to listen to your baby cry in the crib), or for entertaining your friends. It turns out that with a very slight modification it can be attached to your communication line. Then, if you put a standard FM receiver about 300 feet away, the wireless mike will transmit to it the knowledge, what our traders are doing.

It is a competitive business. We are handling, as other U.S. banks, the business for U.S. businesses, foreign businesses, U.S. and foreign governments. A lot of money flows over those lines. Well, if somebody got across those lines and started trading against us based on inside information, we would probably wind up in a negative position of about $150 million or $180 million at the end of the first week (that is a guess). And what would happen? The following Monday, the department head would get everybody together and read them the riot act.

And the next week the chagrined but perplexed traders would wind up losing another $150 million or maybe $200 million. At about that time somebody would come in and start looking for a fraud. Perhaps employees would be interrogated. And their morale would suffer. So they would all go to the local pub and not hoist one or two but a moderate eight to ten and show up bleary eyed the next day and somebody would look at them and say, "Is our foreign exchange trading department a bunch of alcoholics and is that why they are doing so poorly?"

The thing that killed Franklin National Bank was their losses in foreign exchange trading; they couldn't take it. And I don't think it had anything at all to do with a line tap. But a passive tap in this kind of an environment can cost you a lot of money very quickly. So here is a place where encryption is clearly needed.

Disadvantages of Encryption for Funds Transfers

We could also use an encryption technology to protect a funds transfer message from changes, and with very careful design prevent insertion of bogus messages. Encryption does not automatically protect against that, by the way. But, there are some real headaches.

First of all, if you get some noise on the line and your encryption devices come unsynchronized, you wind up with a garbled message you can't process. Then you are in that horrible situation I described in the first or second slide: penalty interests, consequential damages, angry customers, boss angry and you are fired.

Some countries will not let us install encryption equipment. Not many but a few. We have concluded that encryption may be the only solution to confidentiality, and local manager users are responsible for making a decision to encrypt or not, but it is just not a viable worldwide solution to our funds transfer security problem.

Citibank's Use of Encryption

However, I should note that we do use encryption equipment and we have since 1974. We do formal telecommunications risk assessments on our lines. Not assessments that take two days to do, but assessments that take 15 minutes to one-half an hour to do. We have DES-based equipment in somewhere between 35 and 40 countries.

We have high grade equipment on something like 250 links overseas, soon to be 500, perhaps 1000. It depends on some negotiations with a government as to whether or not we can get 1000 DES boxes into their country.

The other extreme on equipment is that for some personnel data we went out to the computer hobby store and bought some very inexpensive microprocessors for $400 and implemented the electronic code book method in software. That is great for management data. We have got that in 35 countries.

Protecting Funds Transfers With Authentication

We also looked at authentication, which for us is a means of verifying the identity of the sender of a message and assuring that the integrity of the message is preserved. And you might say why do you need that?

A Fraud Scenario

The scenario is the classic active wire tap scenario except instead of the message being from Citibank to Security Pacific, for example and saying pay to the order of General Motors or some other major client $243 million or whatever it happens to be, suppose somebody gets into the line and modifies the destination or destination account number and instead it winds up in Denny's account. Well, he will be happy. But somebody is going to be very unhappy because the money is not only gone but nobody knows where it went or how it got out of the system.

Furthermore, if the person running the active tape is smart, he will run a two-way active tap and answer query messages himself, telling both ends of the system that everything is just fine. Three weeks later when the customer is bouncing up and down and screaming and we have been saying, "But see, here is a copy of the message I sent you and it says you really got it," everybody will get serious and realize the money is gone and probably for good.

There have been some attempts of that sort in the last couple of years. A central bank of an African country transferred $21 million through a New York bank, not us. They said please pay out 21 million via CHIPS (New York Clearing House Association) to Barkley's International on the west coast.

It was a properly authenticated message. It used the older banker authentication scheme. It is a marvelously complex system. You take 100 random numbers and print them on the card and never change the card. For each transaction you take the next random number and add it algebraically to the dollar amount in thousands and you can concatenate that with the sum of the sequence number and the date. You put that on the same page as the rest of the message. That's security!

I think some of us understand that there is a little different and a little better way to do it. But that was the kind of a system they used. At any rate, Barkley's got the $21 million the next day. They got a phone call asking that they move the money to Lloyd's, and they moved it to Lloyd's. At Lloyd's, where they were just getting their systems on line, somebody looked and scratched his head and said, "Boy that is interesting, it is a brand new account, it has $50 in it and now it has $21 million credit."

They sent a cable back through the system to Barkley's, to the New York bank, to the central bank saying, "We got the $21 million. Do you really want us to put it in that account?" The message immediately came back again, properly authenticated and saying, "We sure do and please add another $21 million."

Well, people do tend to get suspicious after a while. A phone call was made to a friend, one of the bankers who lived in the African country. He went quietly, privately, to the head of the central bank. The head of the central bank scratched his head and said, "You know, we've got that much money. We are good for it. But, we never sent the message, and we never received a query message."

A little bit of quiet checking found the two technicians who had gotten themselves employed by the PTT, and had broken this fabulous authentication code, and were generating their own funds transfer messages and answering all the queries about it. They had confederates in Switzerland, the UK, New York, and Los Angeles.

We need something to protect the lines because the crooks are getting smarter. By the way, the minicomputer used to generate that active tap was a Model 28 Teletype, which says something about line speeds and degree of sophistication in some countries.

Use of Modern Authentication Technology in Banking

If you put in a good authentication system, not one of these antique devices, you can detect changes in a message, including transmission errors. That is important. You can detect the bogus message, you can validate changes and financial liability, and that is a key point for us. I want to know that I have got liability for a message, or the Chase or First Chicago or BFA has liability for it.

However, if there are errors in the message or the authentication equipment fails, I can still read it. It is not as if it was encrypted. And I can then take the business risk to process or not to process. If that transaction says "pay out" to a major U.S. corporation or foreign corporation that is a long-standing client of ours and it doesn't authenticate, we are going to go right ahead and pay out because we know that we will always get the money back and the interest on it.

And as a matter of fact, something in uniform commercial code about unlawful enrichment takes care of us in case we have to go to court. If it is not a familiar client, we pick up the phone and call the account officer who calls the client and says, did you really mean to do that? Or we notice that your transaction, instead of going to a normal bank account says, "Debit your account at the XYZ bank and credit it to the Hong Kong Poppy and Sweet Dream Factory, and pay out the money by putting it in the hole in the tree down by the crick." You get suspicious when these things happen.

Authentication is not prohibited by any country because the PTT or government in each country can read the message and they know what you are doing. And if they don't trust the U.S. or they don't trust your motives, at least everything is in plain sight for them.

Other Uses for the Authentication Technique

The authentication technique that we have developed for the financial industry can be used to do other things. First of all, you can use it to prove your software has not been modified or to protect it during a downline load. And furthermore, we think the proof of modification or lack of it will stand up in court. If voice digitizers get less expensive, we believe you can use it to authenticate the voice message, in digital form, and at least flash a light saying, yep that was from the right source, even though the voices sound like Mickey Mouse.

Present Status

As a note in terms of use of this new standard, I have a galley proof here; it is coming out of the printer's today. Our first installation was last week, between a customer's office in Vienna and a processing site of ours in Frankfurt. As I mentioned, it is a DES algorithm. The system that was put in has automated key management. The sign-on process is encrypted. The equipment was built in France, which says something about acceptance of DES. And I should note that the NBS authentication standard utilizes exactly the same technique that we are using in the financial industry.

The Importance of Dual Control in Key Management

I find that there is a handy way to compare dual control and vaults and authentication and DES for managers. A vault uses a combination lock and the lock design is public knowledge. And a DES algorithm which is a computational process is also public knowledge. You get the security for the lock by the combination; for authentication, a secret number that we all know is called key. The holder of the combination is the person entrusted with the contents of the vault. The holder of the key is the person entrusted with the funds transfers. If I give the combination to a vault to another person, that process of giving implies a sharing of financial liability. If I formally send the key for an authentication process to someone else, it implies that we are sharing financial liability for the transaction as it goes over the telecommunication line.

For dual control, every bank vault that you see will have two combination locks. In authentication systems for dual control we use two keys, both random exclusive within the equipment. You have got to change combinations on a routine basis and keys on a routine basis. If you have a non-random combination (one combination is a birthday and the other one is a phone number), you have got troubles. The same thing applies in terms of being sure your keys are random.

RESPONSES TO QUESTIONS

The price tag for the intelligence community that will result from the use of authentication is not 0 but almost 0. The price for the intelligence community of having DES about is that sooner or later everybody is going to wind up going to DES or an equivalent good algorithm.

We have designed the authentication algorithm so that we can encrypt the envelope, and where we require confidentiality protection we could authenticate and then encrypt as well. We tend to be very careful, and do a risk assessment first to be sure that the safeguard is really worth the cost.

Somebody, the other day, placed an order for 300 microcomputers to be sure we had at least one in every branch for people to start playing with and learning about. In the small part of the office that I am in we have about 15 of them. When those things start talking to each other, there will be audit procedures in place; there will be a way for making sure that the software is not tampered with. Our auditors are taking a look at using authentication to verify from a remote location that firmware or software has not been changed. Beyond that I am not sure, other than adding our normal controls, what we can do. It is a problem. □

SYSTEM CHARACTERISTICS

- DISTRIBUTED
  - By Process/Product
  - Geographic (domestic and international)
- RAPIDLY CHANGING CUSTOMER REQUIREMENTS
- HIGH TRANSACTION VALUES
- PENALTIES/LIABILITY FOR LATE DELIVERY

EFFECTS ON THE DEVELOPMENT OF DATA SECURITY

- PEOPLE ARE INVOLVED—THEREFORE, GOOD CONTROLS AND PROCEDURES ARE NECESSARY
- CENTRALIZED PROCESSING BECOMES STIFLING TO PRODUCT DEVELOPMENT
- DECENTRALIZATION/TELECOMMUNICATIONS ADDS LINE SECURITY RISK

DECENTRALIZATION VS SECURE OPERATING SYSTEMS

- SECURE OPERATING SYSTEMS ARE SEEN AS AN ACADEMIC FICTION, ARE COSTLY, DIFFICULT TO CONTROL, AND THEIR USE INHIBITS PRODUCT DEVELOPMENT FLEXIBILITY

THEREFORE:

- MAKE EACH PROCESS A SECURE, SELF-CONTAINED ENTITY. THIS RESULTS IN:
  - Single Process CPU's
  - No Need To Compartmentalize Within a CPU
  - Fixed Price Development, NO PROGRAMMERS on Staff, in the Processing Area
  - Need to Know Basic OS Control Weaknesses/Add Compensating Controls
  - Need to Protect Links Between Processes

ALTERNATIVE METHODS OF PROTECTING BANK DATA

ENCRYPTION (''SCRAMBLING'')

ADVANTAGES

- PRESERVES CONFIDENTIALITY
- PREVENTS CHANGES IN A FUNDS TRANSFER MESSAGE
- WITH CAREFUL DESIGN, PREVENTS INSERTION OF BOGUS FUNDS TRANSFERS

DISADVANTAGES

- COMMUNICATIONS LINE ''NOISE'' CAUSES A ''GARBLED'' MESSAGE, WHICH CAN'T BE PROCESSED
- SOME COUNTRIES DO NOT ALLOW ENCRYPTION

CONCLUSION

ENCRYPTION:

- IS THE ONLY SOLUTION TO THE CONFIDENTIALITY PROBLEM, BUT
- IS NOT A VIABLE WORLDWIDE SOLUTION TO THE BANK'S FUNDS TRANSFER SECURITY PROBLEM

AUTHENTICATION

- IS A TECHNIQUE TO
  - — VERIFY THE IDENTITY OF THE SENDER OF A MESSAGE
  - — ASSURE THAT THE INTEGRITY OF THE MESSAGE IS PRESERVED

ADVANTAGES

- DETECTS CHANGES IN A MESSAGE (INCLUDING TRANSMISSION ERRORS)
- DETECTS BOGUS MESSAGES
- CAN VALIDATE CHANGES IN FINANCIAL LIABILITY
- MESSAGE CAN BE READ AND PROCESSED
  - — IF THERE ARE ERRORS IN THE MESSAGE
  - — IF EQUIPMENT FAILS
- NOT PROHIBITED BY ANY COUNTRY
- AUTHENTICATION TECHNIQUE CAN ALSO BE USED TO:
  - — PROVE THAT SOFTWARE HAS NOT BEEN MODIFIED
  - — DOWNLINE LOAD NEW SOFTWARE
  - — AUTHENTICATE DIGITIZED VOICE MESSAGES (TAKES PLACE OF VOICE RECOGNITION IN SITUATION WHERE QUALITY IS SEVERELY DEGRADED)

DES AND VAULT LOCKS: AN ANALOGY

A VAULT USES A COMBINATION LOCK	ANSI AUTHENTICATION USES DES ALGORITHM
• LOCK DESIGN IS PUBLIC KNOWLEDGE	• ALGORITHM—COMPUTATION PROCEDURE—IS PUBLIC KNOWLEDGE
• SECURITY IS PROVIDED BY A SECRET NUMBER, THE COMBINATION	• SECURITY IS PROVIDED BY A SECRET NUMBER, THE *KEY*
• AUTHORIZED HOLDER OF COMBINATION IS ENTRUSTED WITH CONTENTS OF VAULT	• AUTHORIZED HOLDER OF KEY IS ENTRUSTED WITH THE FUNDS TRANSFER(S)
• FORMALLY GIVING THE COMBINATION TO ANOTHER IMPLIES A SHARING OF FINANCIAL LIABILITY FOR THE VAULT	• FORMALLY SENDING THE KEY TO ANOTHER BANK OR CUSTOMER IMPLIES A SHARED FINANCIAL LIABILITY FOR THE TRANSACTION
• FOR DUAL CONTROL, TWO LOCKS ARE USED	• FOR DUAL CONTROL, TWO KEYS ARE USED
• COMBINATIONS ARE CHANGED ON ROUTINE BASIS	• KEYS ARE CHANGED ON ROUTINE BASIS
• NON-RANDOM COMBINATION REDUCES SECURITY	• NON-RANDOM KEY GREATLY REDUCES SECURITY

COST-BENEFIT IMPACT ANALYSIS OF COMPUTER SECURITY STANDARDS/GUIDELINES: A BASE CASE FRAMEWORK

Dr. Marco Fiorello
President
Fiorello, Shaw & Associates

Marco received B.S. and M.B.A. degrees in Production Management and Mechanical Engineering, and a Ph.D. in Operations Research, all from the University of California at Berkeley. He has been an analyst at RAND, IBM, and Aerojet General, and was the Director of Management Science and Systems Studies at the Logistics Management Institute. For the past five years he has been teaching a short course in Life Cycle Cost Analysis for the George Washington University School of Engineering, Continuing Education Program, and is preparing a book on the subject. Dr. Fiorello has also taught graduate courses in management information systems at UCLA, and Operations Research, Macro- and Micro-Economics, and the Economics of Information and Decisions at the Graduate School of Engineering, Northrop University, Los Angeles. He is the principal author of "Costs and Benefits of Federal Automated Data Processing Standards: Guidelines for Analysis and Preliminary Estimating Techniques" published in 1978, and used by the Institute for Computer Sciences and Technology for assessing prospective standards and guidelines. Over the past three years, Dr. Fiorello has reviewed and participated in over twelve cost-benefit analyses of Federal Information Processing Standards, including the prospective password use standard. He is currently working on a project to improve the analysis of the costs and benefits of computer-related security standards or guidelines. Fiorello, Shaw and Associates is a private consulting firm that specializes in analysis of business and policy issues, and evaluation of alternative management strategies. Its major work is planning and resource management studies in the areas of defense, energy and information systems.

1.0 INTRODUCTION

The principal objective of cost-benefit impact assessments of Federal Information Processing Standards (FIPS) or guidelines is to determine the nature and magnitude of the impacts, attributable to the prospective standardization, on the government's current way of doing business.

This discussion presents interim results from a project aimed at improving the analysis of the costs and benefits of computer-related security standards or guidelines[1]. As a starting point in the project we have used the preliminary FIPS cost-benefit analysis framework presented in [FIOR-78][2]. As noted in Exhibit 1, a fundamental component of any cost-benefit analysis is the definition of the contemporary setting or base case. Our objective is to develop a comprehensive framework for defining the base case description needed in the cost-benefit analysis of computer-related security FIPS or guidelines. For our purposes, the base case description specifies the status-quo conditions that exist prior to the introduction of the FIPS/Guideline, and is the basis for projecting the likely conditions that are expected in lieu of the introduction of the FIPS/Guideline. All cost-benefit impacts are defined relative to the prevailing base case setting for the FIPS/Guideline.

Establishing a base case description for computer-related security actions is difficult. We have carried out a cost-benefit impact analysis of the Password Use Standard (see Exhibit 2), and several observations from that and other studies are noted in Exhibit 3. There are formidable data constraints, methodology

[1] Based on work by M. Fiorello and P. Eirich of FSA on a project for the Institute for Computer Sciences and Technology; COTR, Dr. Stuart Katzke.

[2] References are noted by 4-alpha and 2-numeric characters within brackets and are listed in Appendix A.

limitations, and institutional disincentives to reveal potential vulnerabilities for—and actual instances of—computer misuse. Our approach to deal with these difficulties is to develop a descriptive model of Computer Security Risk Management (CSRM). The preliminary model consists of three major components: the computer misuse events or process, the results or impact space, and the set of candidate management actions to prevent or control the events. Exhibit 4 illustrates this general model. There are literally hundreds of actions (see [NBS-80a], [NEIL-76], [LAND-81a] and [RUDE-78] for example) that can be implemented singly or in combinations; User Access Authorization (UAA) standardization is one example, which we will use for illustrative purposes in this discussion.

In Exhibit 4, the CSRM base case consists of the events and the impacts; the UAA standardization base case is a subset of specific UAA-related events and impacts. Our focus in this analysis is on the base case description of events and impacts; particularly, on the former. We believe that once we develop an accurate, logical model of the events that cause or result in computer misuse, then we can formulate a testable base case definition and a logical linkage between actions and events.

2.0 A DESCRIPTIVE MODEL PERSPECTIVE OF THE BASE CASE

By a descriptive model we mean a representation of a process in terms of a sequence of identifiable and traceable, steps or events, the end result of which can be measured and evaluated. Accordingly, the model will be elaborated in terms of the events that lead from the inception of a computer crime, abuse, or error to the ultimate impact on an organization. Exhibit 5 portrays these events. A computer abuse or error can occur when a *perpetrator* is *motivated* to *access* computer *resources* and take *actions,* which may or may not be *authorized,* that affect an organizational *resource* and have an *impact* on the organization. All of these elements, which become "events" when introduced into an organizational environment, must be present for an abuse or error to result. In this model we do not deal with fires, floods, and other gross natural disasters, which in general are dealt with adequately in the literature on computer system risk management.

This event sequence provides a "unifying flow" for constructing the model. Such an approach provides a number of advantages:

- the events in the unifying flow can be compared to reality in a straightforward manner, so the reasonableness and applicability of the model can be evaluated by a user.
- the variety of overlapping and/or inconsistent checklists of computer crime elements found in the literature can be placed in context and related one-to-another within this model.
- the effects of safeguards in deterring, preventing, or detecting computer abuse can be localized to specific points in the descriptive flow, allowing the interactions and combined effects of multiple safeguards to be analyzed.
- the effects of standards and guidelines on data processing (DP) practices can similarly be specified in terms of specific events in the unifying flow.
- selected events in the logical computer security model described herein can be matched to specific points in the descriptive model for overall DP operations currently under development [FIOR-81].

These advantages combine to make such a model effective for decisions on standards, guidelines, and safeguards, in that the resulting benefits can be identified and communicated to management, in a precise manner.

In attempting to quantify a base case description in the computer security area, one faces a lack of consistent and reliable data on computer abuse for the variety of reasons often cited in the computer security literature[3]. A descriptive model approach is helpful in this regard as it enables such statistics as are available to be organized and utilized consistently.

2.1 Computer Security and Risk Management Logical Model

The event sequence that forms the unifying flow for the overall computer security and risk management (CSRM) model, shown in Exhibit 5, begins with a *perpetrator* (one or more) who has either a deliberate

[3] For example, reluctance of companies or agencies to admit to having been victimized or provide details for study practice of government agencies in not separately identifying computer fraud cases in their case records, apparent small fraction of such cases that reach the (public) judicial process, etc.

motivation for accomplishing an abuse, or a factor(s) responsible for the inadvertant creation of an error. The importance and role of the remaining model events will differ depending on whether the motivation is deliberate or inadvertant.

The perpetrator must have a time and place to *access* or indirectly influence the target computer system *resource or process.* Some *action,* of a *type* that may be either authorized or unauthorized, and may be part of a coordinated *strategy* of related actions, must be taken by the perpetrator (or a proper action may be omitted) in order to attack or compromise a *target resource* of value, and have an impact on an ADP system, and hence the responsible organization. Each event must exist if an ultimate impact is to occur. Note that an impact that yields a gain for the perpetrator need not imply a loss for the organization, and vice-versa.

2.1.1 Perpetrators of Error/Abuse

One of the first psychological hurdles to be overcome by a novice programmer is the tendency to assume, after diligent fruitless searching for a program bug, that the computer must have made an adding mistake "just this once." How else could the wrong answer from this obviously perfect program be explained? But the old saying, "computers don't make errors—people do," generally applies.

The advanced programmer may encounter operating systems or even hardware design flaws in the course of exercising little known (and not long remembered) features of an ADP system. However, except for bona-fide component failures, most failures can be ultimately traced to human error somewhere along the line.

The point here is that errors, and (especially) instances of fraud and abuse, begin with a person(s) at the source—either as a cause or a contributing factor—and generally the person is not too far removed from the error. The perpetrator, therefore, is the first component of the computer abuse model. Exhibit 6 gives the categories by which perpetrators may be classified. Each perpetrator category will be associated with different patterns of categories/attributes among the remaining events in the model—a "modus operandi" if you will.

The "event" that takes place here is that either:

(i) the perpetrator has an idea that he or she could, and would like to, accomplish a computer fraud or abuse, and/or,

(ii) the perpetrator is in a situation in which an error can be made.

The motivations for such an idea (i), and/or the situational causes for an error (ii) are covered in the next section. Note that the "and" in the "and/or" is not merely for form's sake—many computer frauds and abuses are detected only because the perpetrator has himself gotten into an error situation, or because some implicit assumption made by the perpetrator has been violated. As will be discussed in future work, one means by which safeguards have an effect is by creating, for perpetrators, potential error situations, and then detecting the errors that result.

For fraud and abuse, the perpetrators of greatest concern are data entry/technical operators, officer/managers, and technically knowledgeable outsiders, who accounted for over 65% of the losses in one study of 150 major cases [ALLE-77]. Although in this study several very large cases may have biased the results in favor of data entry, and the fraction of losses where the perpetrator was unknown (24%) was substantial, a variety of other evidence cited in [FIOR-81a] supports the conclusion that frauds committed by data entry/technical operators are the most significant category.

The data available on errors are less explicit, but the evidence presented in [FIOR-81a] leads to the conclusion that administrative personnel, as opposed to programmers and analysts, are responsible for the majority of errors found in DP applications.

2.1.2 Motivations for Error/Abuse

Different motivations apply to inadvertant, as opposed to deliberate, acts (or omissions) that lead to errors and abuse in computer systems. In fact, "motivation" is not really the correct term in the case of errors—here we are referring to any specific factor(s) or environmental situation(s) that pre-dispose errors to occur more frequently than some achievable baseline level.

Our conceptual baseline, for any type of DP activity, will be a hypothetical organization staffed with dedicated personnel of high morale, who perform that activity utilizing the accepted state-of-the-art in DP

equipment, and in accordance with the generally recognized state-of-the-art in DP practices and managerial procedures.

By "state-of-the-art" we do not mean the newest gadget on the market or the latest management approach in the literature. Rather, we mean those equipments and procedures that have been employed successfully and satisfactorily by one or more organizations generally regarded as having a model DP shop.

Even our hypothetical organization will have some residual level of errors and deliberate abuse, and we are primarily interested in those motivations and factors that cause an increase in errors above this level. Exhibit 7 characterizes both motivations for deliberate acts and contributing factors for accidental errors and omissions.

Of deliberate motivations, material gain appears to be, by far, the most significant in terms of causing problems for organizations. Mischief/challenge and curiosity may well be more frequent, although little data exist on this point, but we believe the impacts are relatively trivial in effect compared to actions motivated by material gain. Material gain may be differentiated either as a gain for a perpetrator, relative to a stable situation, or as the resolution of sudden and/or unusual material need, such as obtaining funds for an operation to treat a serious medical problem. Such data as are available concerning motivations for computer abuse do not differentiate between these two forms of material gain. We suspect, however, that the characteristics of computer crimes, as represented by the remaining elements in the logical model, might be found to vary in each instance.

Apart from the contributing factors that lead to accidental errors and omissions, as shown in Exhibit 7, there is the residual level of errors, inherent in human nature and the present state-of-the-art, that will occur even under the best of conditions. Most of these accidental factors are general in nature and apply to a number of different kinds of errors. For example, inadequate documentation could lead to data entry errors as well as software development mistakes.

However, the most serious problem at the present time, and the area of greatest payoff, appears to be getting organizations to acknowledge the importance of the security problem and to take advantage of the state-of-the-art practices that now exist.

2.1.3 Access

Categories for where and when a perpetrator may obtain access to computer systems are shown in Exhibit 8. This is one of the security model elements that applies specifically to the descriptive model for data processing operations [FIOR-81b] mentioned earlier.

The predominant location for the initiation of computer abuse is important to know for the placement of physical safeguards as well as for the selection of safeguards related to mode of access. Knowledge of when abuses occur may either encourage or discourage the application time-dependent safeguards. According to [PARK-76a] the most significant access location vulnerabilities (after translation of Parker's Terminology into the Terminology of Exhibit 8) appears to be in the central computer site and the key entry area.

2.1.4 Process or Resource that is Influenced, Modified, or Used

Exhibit 9 categorizes the system resources and activities that become the "vehicle" through which the abuse or error can occur. This is not a tightly defined element, in that its characteristics are a mixture of:

things — hardware, telecommunications links, media;
activities — data input, system operations;
concepts — design philosophy; and
software — which is not precisely any of the above.

Yet, all of these diverse entities are important to successful computer applications, and any can be subverted to commit an abuse, to perpetrate an error, or to create a situation in which abuses and errors are more easily accomplished and/or more likely to be attempted.

The vehicle of interest is one primarily subverted to commit a crime or abuse, or the one that is most directly the source for an error. For instance, hardware and software are of necessity utilized when a data entry clerk submits a fraudulent transaction, which is processed on the computer in conjunction with legitimate transactions. However, the vehicle of interest here is the data input activity, where the transaction

occurs. Hardware and software are included to represent those instances where the hardware and/or software of a system are attacked directly, and not necessarily in the course of some otherwise normal system activity. However, if some direct attack on software could only be undertaken under the camouflage of some particular system operations activity, then both would be indicated as the vehicle.

Overall, data input is indicated, in several studies [GAO–76a/ALLE–77], as clearly the most significant source for fraud and abuse. Data input, overall system philosophy, algorithm design, and application software development are significant sources of error [MART–73]. In some studies [GAO–79b] data input errors appear to be the most serious problem, while in others [GAO–76b] software errors appear more significant. However, different types of applications were represented in these studies, and we are not aware of any study sufficiently comprehensive to resolve this point.

2.1.5 Action and Type of Authorization

Exhibit 10 lists what actions may be taken by a perpetrator and indicates that the action taken may or may not be of a type within the normal job authority of the perpetrator. For example, the preparation of fraudulent transactions is never authorized, but the *act* of entering specific transactions would be authorized for certain terminal operators, while it would not be authorized for programmers, or for technical operators in departments other than the one in which those particular transactions are normally prepared. A gray area may be found in the case of officers/managers, who may be authorized to enter certain transactions under their cognizance, but who do not normally enter transactions as part of their daily job activity.

As indicated in [GAO–76a/ALLE–77] and other studies cited in [FIOR–81a], the initiation of fraudulent transactions appears to be the action causing most significant computer abuses. Studies such as [GAO–79b] and others suggest that transaction preparation and entry are also the actions causing the most significant error problems. Although we have not seen the authorization issue addressed specifically in a study, our interpretation of the general computer crime literature suggests that most fraud and abuse occurs within the scope of authorized and normal job authority.

2.1.6 Strategy/Target Resource

To realize a material gain and/or cause loss to an organization, computer-related fraud or abuse must obtain, or destroy some resource managed or controlled by the data processing system. Common targets are given in Exhibit 11, and are to be distinguished from the system resources listed earlier as vehicles for abuse. For example, one perpetrator might modify application software to serve as a vehicle for reaching his target, which might be inventory materials. To another perpetrator, that same application software might itself be the target in a strategy to sell pirated software, and compromised telecommunications could be the vehicle.

In complex strategies the target resource may be a system resource whose compromise yields no direct gain, but does provide a vehicle for attempting a further attack on some other target of value. There could be several such iterations between an initial compromise, achieved in the course of general system use, and the ultimate compromise of a target resource.

Inventory materials and financial instruments are the targets accounting for the most sizeable losses from computer fraud and abuse. Depending on one's interpretation of the data in [GAO–76a], namely whether or not to include losses that *could* have occurred *if* several schemes had not been detected before being completed, either target could be named as the most significant.

2.1.7 Impacts

Exhibit 4 characterizes the ways in which errors and abuse can have negative impacts on agency and/or the people it serves. The term errors and omissions implies any type of mis-allocation of resources, due to errors and/or omissions, in processing, whether resulting from over- or under-payments, improper uncollected receivables, loss of materials, incorrect management decisions, or whatever.

Any of the categories shown can result from either deliberate or inadvertant actions. For instance, resource allocation errors, could result as a by-product of fraudulent activity; and errors or omissions could create a situation in which an otherwise impossible fraud or embezzlement could take place.

While precise dollar amounts are difficult to determine, it is generally agreed that errors and omissions, leading to various types of resource mis-allocations, represent the largest financial impacts. Next in line are losses from fraud and abuse.

2.1.8 Summary

We have described the elements and the element categories that comprise the overall logical mode for computer security and risk management, and the generally important categories have been noted. Next we will highlight those combinations of element categories that are most relevant for our illustrative analysis of user access authorization.

There are three parts to the base case formulation for a user access authorization action: a description of the process whereby users access computer systems, estimates of the magnitudes and frequencies of undesirable accesses to federal ADP systems, and a description of the federal ADP inventory. The first part is particularly essential, because, in order to prevent or detect undesirable user access to computers we must first understand the who, why, what, how and related effects of the contemporary process. Once the framework is defined in terms of those dimensions, then estimates of the relative magnitudes and logical combinations of elements within the dimensions can be prepared to complete the base case description, and relate it to the DP system inventory of interest.

2.2. Model Components Relevant to User Access Authorization

2.2.1 Significant Components and Elements

Of the vast combinations of components (or categories) possible within the elements of the overall logical model, only some will be significant for the user access authorization base case. This discussion develops the relationship between the CSRM model elaborated above, and a functional model for user access authorization developed by SDC [SDC-80], [SDC-81], [SDC-82].

"As the purpose of user access authorization is to control access to and use of system resources" [SDC-81a, p. 5-6], we will begin with the element of system resources used as *vehicles* (Exhibit 9). Of the 12 resource categories listed,

- hardware
- system software
- application software and
- data input transactions

are objects explicity controlled under the functional model for standard, and could therefore be restricted in their use as vehicles. Also, general system use would be limited in its potential use as a vehicle since the standard would allow limiting the *type* of actions possible under general system use to be only authorized actions (Exhibit 10):

- add/delete/alter/examine . . .
  - transactions
  - data files
  - application software
  - system software

which have been discussed above and:

- improper use of . . .
  - communications system
  - processing

which may be partly controlled, in that terminals and processes are among the subjects covered under the standard.

User access authorization is effective in controlling attacks on all target resources (Exhibit 11) except possibly accomplishment of non-job-related tasks and transmission of non-job-related communications. Attacks on all those targets could be deterred by a journalling feature, even if these attacks were accomplished solely through the use of actions that might be authorized for a particular perpetrator.

In terms of impacts from computer misuse (Exhibit 8), user access authorization could lead to a reduction of any of the impacts, with the exception of equipment damage. It should be particularly effective against

privacy intrusions, alteration of records, and theft of computerized information, by eliminating unauthorized actions and logging authorized ones. Access authorizations will have some effect on errors/omissions and fraud/embezzlement, depending upon the specific vehicles, actions, and access locations involved in the accident or fraudulent scheme.

The perpetrators (Exhibit 6) against which user access controls would be effective are those that interact directly with the computer system and utilize it in a generally unstructured manner. These personnel include:

- systems programmer
- applications programmer
- office manager
- outsider, technically knowledgeable

Such personnel would make the kinds of errors that user access controls could detect and prevent, such as the inadvertant deletion of a master data file by an inexperienced programmer. Also, these personnel are the ones with sufficient technical knowledge to subvert a computer system with the types of attacks that user access controls are designed to counter. In contrast, terminal operators, for example, tend to attack systems with improper (but innocent-looking) transactions that would appear perfectly legitimate to most access authorization systems. Only an extensive application of journalling and monitoring of transactions would act as a deterrent here. However, by tightly limiting data field and record access to the minimum permissions necessary for job performance, such impacts as privacy intrusions may be curtailed for all categories of perpetrators.

User access authorization is not especially sensitive to differences in motivation (Exhibit 7).

Finally, access (Exhibit 8) is affected strongly by user access authorization since terminals and processes are controlled, and the control may be based on date and time of day.

The greatest potential control of access occurs when terminals (on-site, off-site, or remote batch or the computer console) provide the access path. However, no matter where forms or transactions are prepared, once submitted into the computer system, they are controlled by a system process which is, itself, subject (or potentially subject) to access and dissemination authorization controls.

2.2.2 Combinations of Components

As noted, all of the elements in the computer security and risk management logical model must be represented in the DP system environment before an error or abuse can occur. The above discussion treated each element separately. Next we will illustrate specific combinations relevant to the consideration of user access controls.

Each combination may be viewed as a specific channel travelling along the length of the unifying flow. Some illustrative vignettes are as follows:

1) An application programmer. . .
- is overworked while attempting to meet a critical deadline. . .
- and from his terminal. . .
- in the course of normal program compilation, resaves the compiled object file with insufficient file name specification. . .
- and overwrites the program source code. . .
- which he is authorized to do, (but
- which would probably have been prevented if the source code had employed a different password than the object code).
- This destruction of software. . .
- has the impact of a delay in the performance of the agency mission and/or additional operational cost.

The delay may or may not be lengthy, and the additional operations costs may or may not be significant, depending on the backup/recovery practices in use by the computer center and/or the applications development team.

2) A terminal operator (in the personnel dept.). . .
- seeking to get even with a co-worker who was selected for promotion over him, . . .

- uses his terminal, during regular hours. . .
  - in the course of his normal, authorized system use, . . .
    - to surreptitiously examine the co-worker's personnel records, . . .
      - which intrudes on the co-worker's privacy.

The co-worker could be barred if the terminal operator discovered material detrimental to the co-worker but not generally known within the organization, and in violation of the law if the data viewed improperly (i.e., not required to perform her job) were of a type protected by the Privacy Act. Data field authorization controls might have prevented this, depending on the specific authority possessed by the terminal operator.

3) A financial manager. . .
  - to gain a financial advantage in the commodities market, . . .
    - uses a remote batch terminal on a Saturday, . . .
      - in conjunction with other authorized use as a cover, . . .
        - to examine budgets reflecting planned government commodity purchases, . . .
          - something outside his normal job authority
            - and from the use of this data
              - he achieves his material gain.

The agency may not suffer a corresponding loss, but its reputation could suffer if the abuse became known, and if the manager sold the data to speculators, the government could pay excessive costs for any commodities purchased.

These three illustrations are hypothetical, yet representative of actual instances. [USDA-78] and [USDA-80] are excellent audit studies that document both the variety of access abuses that can occur in the absence of effective access controls, and the effectiveness of a good access control system in preventing those abuses. Future study efforts will attempt to locate or develop comparable data for additional types of DP environments in order to determine the predominant types of access abuse and to estimate their relative impacts and frequencies of occurrence.

3.0 FUTURE ANALYSIS

This discussion reports on work in progress directed toward the development of an effective framework for the impact analysis of computer security and risk management actions. These preliminary results need further expansion along the following points noted in Exhibit 12.

3.1 Refinement of the Descriptive Model Framework

At a minimum the descriptive model of events that results in computer system misuse must provide a rational interpretation of computer misuses, be communicable to managers of ADP facilities, and facilitate the selection and evolution of optimal actions. Each of the elements within a dimension should be independent or sufficiently distinguishable to be the focus of a prospective action. They should also permit a useful basis to interpret relative magnitudes and frequencies.

3.2 Define Scenarios That Logically Combine Dimensions and Elements

The descriptive model presented is largely a logical model of the ways computer system mishaps occur. In any one agency, however, just a few of the potential combinations of elements across the model dimensions are relevant to its functional and data processing setting. These scenarios need to be developed to illustrate the framework and systematically portray the computer security and risk management problems.

3.3 Map the Computer System Mishap Model onto the Computer Processing Descriptive Model

Computer security is one of the components of a data processing facility. In another related effort descriptive models are being prepared for the data processing/operators functions and the application software development function, and descriptive models are planned for other functions such as data resource management, software acquisition, hardware acquisition, etc.

Mapping the framework of the computer system mishap model onto these other descriptive models will tie all the descriptive models together and ultimately create a full-ADP facility descriptive model with component intersections.

3.4 Relate Corrective/Preventative Action to the Computer System Mishap Framework Dimensions and Elements

In the final analysis, managers must select and implement a set of factors to correct or prevent computer system mishaps. The above points are building blocks to facilitate the selection and evaluation of candidate actions in the control of the descriptive framework of the ADP facility. There are literally hundreds of candidate actions and hundreds more of combinations of those actions that a facility manager can apply. The objective is to tie actions to computer mishap causes to impacts, and to facilitate the selection of appropriate (optimal) combinations of actions.

APPENDIX

Selected Computer Security and Risk Management References

Note: The abbreviation *"C&S"* refers to

Dinardo, C.T., ed. *Computers and Security,* from the National Computer Conferences, The Information Technology Series, Vol. III AFIPS Press, Montvale, N.J., 1978.

(ALLE-71) Allen, B., "The Biggest Computer Frauds-Lessons for CPA's," *Journal of Accountancy,* May 1977, pp. 52–62.

(ATTA-76) Attanasio, C.R., Markstein, R.W., and Phillips, R.J., "Penetrating an Operating System: A Study of VM/370 Integrity," *IBM Systems,* 1976.

(BALL-82) Ball, L.D., "Computer Crime," *Technology Review,* April 1982.

(BECK-78) Becker, J., *The Investigation of Computer Crime,* Battelle Memorial Law and Justice Study Center, Seattle, WA, 1978, (prepared for Law Enforcement Assistance Administration, Washington, D.C.).

(BEQU-78) Bequai, A., *Computer Crime,* Lexington Books, Lexington, Mass., 1978.

(BERN-82) Bernhard, R., "Breaching System Security," *IEEE Spectrum,* June 1982.

(BLOO-80) Bloom, R., "Catching the Computer Crook," *Infosystems,* July 1980, pp. 30–35.

(BRYA-82) Bryan, W.L., Siegel, S.G., and Whiteleather, G.L., "Auditing Throughout the Software Life Cycle: A Primer," *Computer IEEE,* March 1982.

(BUSW-81) *Business Week,* "The Spreading Danger of Computer Crime," April 20, 1981.

(CEBE-80) CEBEMA, *Privacy and Security Bibliography,* 1980.

(CJRM-79) *Criminal Justice Resource Manual: Computer Crime,* National Criminal Justice Information and Statistics Service, LEAA, U.S. Department of Justice, 1979.

(CARY-79) Cary, J.M., *An Examination of a Distributed Architecture Data Security System and Performance Overhead Costs,* George Washington University, May 6, 1979.

(COME-80) Comer, M., "Computer Fraud: It takes a Thief. . .," *Business Matters,* December 1980.

(COMP-80) *Computer World,* "Millions of Dollars in Losses, GAO Reports Deficiencies in Federal. . . . .," June 16, 1980.

(COUR-74) Courtney, R.H., "A Systematic Approach to Data Security," *NBS Symposium on Privacy and Security in Computer Systems,* March 1974.

(DISC-80) Discepolo, A.G., *Computer Security Bibliography,* Mitre Technical Report (#MTR 8199), Nov. 1980.

(DONL-80) Donlan, T.B., "Social Security Scam," *Barrons,* August 18, 1980, pp. 9–24.

(EDPA-79) *EDP Analyzer,* "The Security of Managers' Information," Vol. 17, No. 7, July 1979.

(EDPA-80) *EDP Audit,* "Control and Security Newsletter," 1981 Newsletters, Automation and Training Center, Inc., Reston, Virginia.

(EDPA-80a) *EDP Analyzer,* "Risk Assessment for Distributed Systems," Vol. 18, No. 4, April 1980.

(FIOR-78) Fiorello, M., and Jaffin, S., *Costs and Benefits of Federal Automated Data Processing Standards: Guidelines for Analysis and Preliminary Estimating Techniques,* Logistics Management Institute, Task DC801, Washington, D.C., August 1978.

(FIOR-81) Fiorello, M. and Eirich, P., *Impact Assessment of the Proposed Password Use Standard,* FSA, McLean, Virginia, June 1981.

(FIOR-81a) Fiorello, M. and Eirich, P., *Improved FIPS Cost-Benefit Methodology Descriptive Models,* FSA, McLean, Virginia, September 1981.

(GLAS-77) Glaseman, S., Turn, R., Gaines, S., *Problem Areas in Computer Security Assessment,* The RAND Corp., Santa Monica, California, (P-5822), February 1977.

(GOLD-78) Gold, R.B., et al, *Final Report VM/370 Security Retrofit Program Detailed Design and Implementation Phase,* (Rep. #TM-6062/001/00), May 21, 1978.

(GAO-74) *Increased Efficiency Predicted if Information Processing Systems of Social Security Administration are Redesigned,* U.S. General Accounting Office, (B-164031(4)), April 19, 1974.

(GAO-76) *Improvements Needed in Management Automated Decisionmaking by Computers Throughout the Federal Government,* U.S. General Accounting Office, (FGSMD-76-5), April 23, 1976.

(GAO-76a) *Computer-Related Crimes in Federal Programs,* U.S. General Accounting Office, (FGMSD-76-27), April 27, 1976.

(GAO-76b) *Managers Need to Provide Better Protection for Federal Automatic Data Processing Activities,* U.S. General Accounting Office, (FGSMD-76-40), May 10, 1976.

(GAO-77) *New Methods Needed for Checking Payments Made by Computers,* U.S. General Accounting Office, (FGSMD-76-82), November 7, 1977.

(GAO-78) *Procedures to Safeguard Social Security Beneficiary Records Can and Should be Improved,* U.S. General Accounting Office, (HRD-78-116), June 5, 1978.

(GAO-78a) *Federal Agencies Can, and Should, Do More to Combat Fraud in Government Programs,* U.S. General Accounting Office, (GGD-78-62), September 19, 1978.

(GAO-79) *Automated Systems Security—Federal Agencies Should Strengthen Safeguards Over Personal and Other Sensitive Data,* U.S. General Accounting Office, (LCD-78-123), January 23, 1979.

(GAO-79a) *Flaws in Controls Over the Supplemental Security Income Computerized System Cause Millions in Erroneous Payments,* U.S. General Accounting Office, (HRD-79-104), August 9, 1979.

(GAO-80) *The Social Security Administration Needs to Develop a Structured and Planned Approach for Managing and Controlling the Design, Development, and Modification of Its Supplemental Security Income Computerized System,* letter report to the Secretary of HEW, U.S. General Accounting Office, (HRD-80-5), October 16, 1979.

(GAO-80a) *Federal Information Sources and Systems 1980,* (A Directory issued by the Comptroller General), Program Analysis Division, U.S. General Accounting Office, 1980, (GPO# 020-000-00183-6).

(GAO-80b) *GAO Findings on Federal Internal Audit—A Summary,* U.S. General Accounting Office, (FGMSD-80-39), May 27, 1980.

(GAO-80d) *Continuing and Widespread Weaknesses in Internal Controls Result in Losses Through Fraud, Waste, and Abuse,* U.S. General Accounting Office, (FGMSD-80-65), August 28, 1980.

(GAO-80e) *How Extensive is Fraud in Government and Who Commits It?,* U.S. General Accounting Office, 1981.

(GAO-81) General Accounting Office, *Fraud in Government Programs: How Extensive Is It? How Can It Be Controlled?,* Volume II, (#AFMD-81-73), Sept. 30, 1981.

(GAO-81a) *Millions Paid Out in Duplicate and Forged Government Checks,* U.S. General Accounting Office, (AFMD-81-68), October 1, 1981.

(GAO-81b) *Fraud in Government Programs: —How Extensive is it? —How Can it be Controlled?,* Vol. I, (AFMD-81-57), November 6, 1981.

(GAO-81c) *Fraud in Government Programs: —How Extensive is it? —How Can it be Controlled?,* Vol. II, (AFMD-81-73), November 6, 1981.

(GAO-81d) *Fraud in Government Programs: —How Extensive is it? —How Can it be Controlled?,* Vol. III, (AFMD-82-3), November 6, 1981.

(GRAY-81) Gray, M., *Computers in the Federal Government: A Compilation of Statistics—1978, 1979, 1980, 1981,* ICST, NBS.

(GSA-79) *Automatic Data Processing Equipment Inventory,* (as of the end of Fiscal Year 1978), General Services Administration, July 1979.

(HEW-75) *ADP System Security Required by the Privacy Act of 1974,* DHEW Information Standards Publication 3, (HEW TN-75.4); Dept. of Health, Education, and Welfare; July 24, 1975.

(HIGG-81) Higgins, A.J., *A Fuzzy Risk Analysis of a Fictional IRS Computer System,* (GWU-IIST-81-15), August 1981.

(HOFF-77) Hoffman, L.J., *Modern Methods for Computer Security and Privacy,* Prentice Hall, Englewood Cliffs, N.J., 1977.

(HOFF-80) Hoffman, L.J., "Inexact Analysis of Risk," *Proceedings of the IEEE,* 1980.

(KING-80) Kingston, P., "Does the End Justify the Cost?" *Canadian Datasystems,* November 1980.

(KREL-77) Krell, R.A., *Development and Analysis of a Model of the Cost Impact of Privacy Safeguards in a Computer Network,* George Washington University, September 1977.

(KURT-80) Kurtz, H., "Blue Shield Withheld Medicare Data from Auditors, Sources Say," *Washington Star,* December 6, 1980, pp. B-1, B-2.

(LAND-81) Landwehr, C.E., *Best Available Technologies (BATs) for Computer Security,* Naval Research Laboratory, December 21, 1981.

(LAND-81a) Landwehr, C.E., "Formal Models for Computer Security," *Computer Surveys,* Vol. 13, No. 3, September 1981.

(MART-73) Martin, J., *Security, Accuracy, and Privacy in Computer Systems,* Prentice Hall, N.J., 1973.

(SDC-80) *Guidelines for User Access Authorization Schemes (Draft),* System Development Corporation, (TM-WD-8051/023/01), November 1980.

(SDC-81) Morse, H.S., *Study Concerning Applicability of a Proposed FIPS Standard on User Access Authorization Schemes,* System Development Corporation, (TM-WD-8051/025/01), April 10, 1981.

(SDC-82) Morse, H.S., "Announcing the Standard for User Access Authorization," System Development Corporation, March 1982.

(NBS-75) *Computer Security Guidelines for Implementing the Privacy Act of 1974,* ICST, NBS, May 1975.

(NBS-80a) *Guidelines for Security of Computer Applications,* Federal Information Processing Standards Publication 73, ICST, NBS, June 30, 1980.

(NEIT-79) Neitzel, L.D. and Hoffman, L.J., "Fuzzy Cost/Benefit Analysis," *International Symposium on Policy Analysis and Information Systems,* Duke University, June 1979.

(NIBA-79) Nibaldi, G.H., *Proposed Technical Evaluation Criteria for Trusted Computer Systems,* Mitre Corp., (M79-225), October 25, 1979.

(NIEL-76) Nielsen, N., Ruder, B, Brandin, H., "Effective Safeguards for Computer Systems Integrity," *1976 NCC,* Vol. 45, (in *C&S,* p. 23).

(NTIS-82) NTIS, *Computer Information Security and Protection,* August 1979—November 1981 Citations, PB81-803008, NTIS, Springfield, Virginia, January 1982.

(PARK-73) Parker, D., Nycum, S., Gena, S., "Computer Abuse," Stanford Research Institute, 1973, as reported in: "Computer Threats and Abuses—Introduction," (in *C&S,* p. 1).

(PARK-76) Parker, D., "Computer Abuse Perpetrators and Vulnerabilities of Computer Systems," *1976 NCC,* Vol. 45, (in *C&S,* pp. 13-21).

(PARK-76a) Parker, D., *Crime by Computer,* Scribners, N.Y., 1976.

(PERR-80) Perry, W.E., *Distributed Embezzlement,* 1980.

(PETE-67) Peterson, H., and Turn, R., "System Implications of Information Privacy," 1967 FJCC, Vol. 30, (in *C&S,* p. 39).

(RUDE-78) Ruder, B., and Madden, J.; Blanc, R., ed., *An Analysis of Computer Security Safeguards for Detecting and Preventing Intentional Computer Misuse,* NBS Special Publication 500-25, January 1978.

(RUSS-77) Russell, S., Basco, T., Fitzgerald, J., *Systems Auditability and Control Study: Data Processing Control Practices Report,* Stanford Research Institute, 1977.

(SALT-75) Saltzer, J.H., and Schroeder, M.D., "The Protection of Information in Computer Systems," *Proceedings of the IEEE,* Vol. 63, No. 9, September 1975.

(SNYD-81) Snyder, L., *Formal Models of Capability-Based Protection Systems,* The BLUE CHIP Project, Purdue University, 1981.

(STEW-80) Steward, R., *Technology Assessment Special Report: User Access Authorization Schemes,* System Development Corporation, (TM-WD-8051/021/01), August 1980.

(TANG-80) Tangney, J.D., *History of Protection in Computer Systems,* Mitre Corporation, (MTR-3999), July 15, 1980.

(TROT-80) Trotter, E.T., and Tasker, P.S., *Industry Trusted Computer System Evaluation Process,* Mitre Corporation, (MTR-3931), May 1, 1980.

(USDA-77) *Review of ADP Security Procedures and Controls, Washington Computer Center and User Agencies, as of September 28, 1978,* (No. 50530-4-Hd), U.S. Department of Agriculture, October 13, 1978.

(USDA-78) *Review of ADP Security Procedures and Controls, Washington Computer Center and User Agencies, as of July 31, 1980,* (No. 50639-1-Hq), U.S. Department of Agriculture, October 13, 1978.

(USS-77) *Staff Study of Computer Security in Federal Programs,* Committee on Government Operations, U.S. Senate, February 1977.

(WARE-79) Ware, W.H., ed., *Security Controls for Computer Systems,* RAND Corporation, Report R-609-a, October 1979.

(WEIN-81) Weingarten, F., *Impact Analysis of the Strawman Standard,* System Development Corporation, (TM-WD-8051/026/01), May 29, 1981.

(WISS-78) Wissing, R.P., *Optimization of a Computer Security Index Versus Cost,* for the Department of Defense, June 1978.

(WOOD-77) Wood, H., "The Use of Passwords for Controlling Access to Remote Computer Systems and Services," 1977 Ncc, Vol. 46, (in *C&S,* p. 137).

(WOOD-80) Wood, E.B., "Data Base Security: Requirements, Policies, and Models," *IBM Systems,* Vol. 19, No. 2, 1980. □

EXHIBIT I

PRELIMINARY METHODOLOGY FOR COST-BENEFIT ANALYSIS OF FEDERAL INFORMATION PROCESSING STANDARDS/GUIDELINES

OBJECTIVE: — PROVIDE A PRAGMATIC DECISION AID

— TO HELP DETERMINE IF FEDERAL ADP FACILITIES WILL BE BETTER OFF WITH THE PROPOSED FIPS/GUIDELINE

PROCEDURE: — EIGHT STEP FRAMEWORK ESTABLISHING

— — GOALS

— — BASE CASE

— — IMPACT MODEL

— — DATA REQUIREMENTS

— — ESTIMATION & INTERPRETATION

APPLICATIONS: 10 HARDWARE & SOFTWARE FIPS
2 COMPUTER SECURITY FIPS/GUIDELINES

EXHIBIT 2

PROPOSED PASSWORD USE STANDARD IMPACT ASSESSMENT

- POSITIVE NET COST-BENEFIT FINDINGS
  - — — REDUCE COMPUTER-RELATED FRAUD
  - — — REDUCE COMPUTER ERRORS & OMISSIONS
  - — — POSITIVE BUILDING BLOCK FOR SECURITY PROGRAM
- CLEARLY INDICATES NEED FOR USER IDENTIFICATION IN THE BASE CASE DEFINITION

EXHIBIT 3

INSIGHTS INTO IMPACT ASSESSMENTS OF COMPUTER SECURITY FIPS/GUIDELINES

- VERY USEFUL FOR GO/NO-GO DECISIONS
- PROVIDE CONSTRUCTIVE INSIGHTS TO IMPROVE FIPS/GUIDELINES
- DISCOVER PRINCIPAL OBJECTIONS
- RELATE THE PROPOSED ACTION TO AN OVERALL SECURITY PROGRAM OR FAMILY
- SYSTEMATIC BUT NOT QUANTITATIVELY RIGOROUS
  - — QUALITATIVE CONSIDERATIONS ARE RELEVANT AND ESSENTIAL
- EXPLICITLY IDENTIFIED MAJOR UNCERTAINTIES AND CONSTRAINTS
  - — UNDERLYING PROCESSES NOT UNDERSTOOD VERY WELL
  - — DATA ARE SPOTTY AND SOFT
  - — IMPLEMENT STANDARDS IN GROUPS AND NOT AS SINGLE ACTIONS

EXHIBIT 4

COMPUTER SECURITY AND RISK MANAGEMENT DESCRIPTION

IMPACTS
(ISSUES)

EVENTS THAT CAUSE

- ERRORS & OMISSIONS
- FRAUD & EMBEZZLEMENT
- PRIVACY INTRUSIONS
- ALTERATION OF RECORDS
- THEFT OF COMPUTERIZED INFO.
- UNAUTHORIZED USAGE
- DENIAL OF SERVICE
- EQUIPMENT DAMAGE
- NON-PERFORMANCE OF AGENCY MISSION
- INCREASED VULNERABILITY

ACTIONS DESIRED TO PREVENT OR CONTROL

BASE CASE COMPONENTS

EXHIBIT 5

DESCRIPTIVE MODEL OF EVENTS THAT CAUSE PEOPLE-RELATED COMPUTER MISUSE

PERPETRATOR (WHO)

↓

MOTIVATION (WHY)

↓

ACCESS (WHERE & WHEN)

↓

PROCESS/RESOURCES (VEHICLE)
MODIFIED/INFLUENCED/USED

↓

ACTION (WHAT)

↓

NATURE OF ACTION (AUTHORIZED/NON-AUTHORIZED)

↓

STRATEGY/TARGET (RESOURCE)

↓

IMPACTS

EXHIBIT 6

PERPETRATOR

(WHO)

1. DATA ENTRY/TERMINAL OPERATOR
2. CLERK/TELLER
3. SYSTEMS PROGRAMMER
4. APPLICATION PROGRAMMER
5. COMPUTER OPERATOR
6. INVENTORY CONTROL STAFF
7. OFFICER/MANAGER
8. OTHER STAFF
9. OUTSIDER — TECHNICALLY KNOWLEDGEABLE
10. OUTSIDE — NOT TECHNICALLY KNOWLEDGEABLE
11. MAINTENANCE TECHNICIAN
12. UNKNOWN

EXHIBIT 7

MOTIVATION

DELIBERATE	*ACCIDENTAL*
1. MATERIAL GAIN	1. IGNORANCE
2. POWER	2. INCOMPETENCE/APATHY/CARELESSNESS
3. PRESTIGE	3. INADEQUATE CROSS CHECKS FOR DATA VERIFICATION
4. MALFEASANCE	4. INADEQUATE DOCUMENTATION PROCEDURE
5. MALEVOLENCE	5. INADEQUATE TRAINING
6. DUTY	6. INADEQUATE ACCOUNTING/AUDIT CONTROLS IN DP PROCEDURES
7. ALTRUISM	7. OVERWORKED STAFF
8. MISCHIEF/CHALLENGE	8. EXCESSIVE SOFTWARE COMPLEXITY
9. CURIOSITY	9. RESIDUAL HUMAN ERROR, INHERENT DESPITE USE OF STATE-OF-THE-ART TECHNOLOGY AND PRACTICES

EXHIBIT 8

ACCESS

WHERE

1. ON-SITE TERMINAL
   — HARDWARE
   — DIAL-UP
2. OFF-SITE TERMINAL
3. COMPUTER PROGRAM DEVELOPMENT OFFICE AREAS DEVELOPMENT
4. CLERICAL OFFICE AREA (FORMS PREPARATION)
5. KEY-ENTRY AREA (AND RELATED EQUIPMENT)
   — DATA ENTRY (STAFF WORK AREA)
   — GENERAL USE (PROGRAMMERS)
6. CENTRAL COMPUTER SITE
   — COMPUTER CONSOLE
   — JOB SUBMISSION
   — OUTPUT PICKUP
   — OPERATIONS
7. REMOTE-BATCH TERMINAL
   — MANNED
   — UN-MANNED
8. ADJACENT TO MAIN SITE OR COMPUTER COMMUNICATION LINKS

WHEN

1. REGULAR BUSINESS HOURS
2. AFTER WORK HOURS
3. OVERNIGHT
4. WEEKENDS

EXHIBIT 9

PROCESS/RESOURCE — MODIFIED/INFLUENCED/USED

(VEHICLE)

1. HARDWARE
2. SYSTEM SOFTWARE (SYSTEM SECURITY)
3. APPLICATION SOFTWARE
4. OPERATIONS ACTIVITY
5. DATA INPUT
6. ALGORITHM DESIGN
7. SYSTEM PHILOSOPHY (DESIGN, METHODS, ASSUMPTION)
8. OUTPUTS
9. GENERAL SYSTEM USE
10. ACCESS EMANATIONS FROM COMPUTER/COMMUNICATIONS SYSTEM
11. TELECOMMUNICATIONS SYSTEM/LINKS
12. PROCESSING/STORAGE MEDIA

EXHIBIT 10

ACTION

(WHAT)

1. ADD/DELETE/ALTER/EXAMINE
   — — TRANSACTIONS
   — — DATA FILES
   — — APPLICATION SOFTWARE
   — — SYSTEM SOFTWARE
2. IMPROPER USE OF
   — — COMMUNICATIONS SYSTEM
   — — PROCESSING
3. MISAPPROPRIATION OF
   — — OUTPUT
   — — MEDIA
4. COVERT INTERCEPTION OF COMMUNICATIONS
   — — ELECTROMAGNETIC RADIATION
   — — DETECTION
   — — WIRETAPS
5. IMPROPER HARDWARE WIRING/CONNECTIONS
6. INADVERTENT CIRCUITRY/ COMPONENT FAILURE
7. SOFTWARE DESIGN FLAW INCORPORATED/ EXECUTED

(TYPE)

1. WITHIN NORMAL JOB ACTIVITY/ AUTHORITY
2. OUTSIDE NORMAL JOB AUTHORITY
3. AUTHORIZED, BUT ATYPICAL OF NORMAL JOB ACTIVITY

EXHIBIT 11

STRATEGY/TARGET RESOURCE

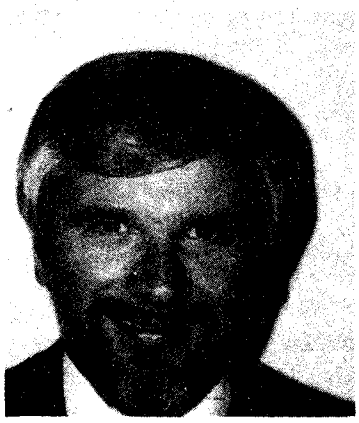
1. SALE/USE/TRANSMITTAL OF
   - DATA
   - SOFTWARE
2. INVENTORY MATERIALS
3. NEGOTIATION OF CHECKS OR FINANCIAL INSTRUMENTS
4. SATISFY CURIOSITY
5. DESTRUCTION OF
   - DATA
   - SOFTWARE
6. ACCOMPLISHMENT OF NON-JOB RELATED TASKS
7. TRANSMISSION OF NON-JOB RELATED COMMUNICATIONS
8. VEHICLE FOR THE PURPOSE OF FACILITATING FURTHER ATTACKS/ COMPROMISES

EXHIBIT 12

ON-GOING RESEARCH

- REFINE DESCRIPTIVE MODEL
  - — ESTABLISH RELATIVE MAGNITUDES
  - — COMPLETE DEFINITION OF ELEMENTS
- DEFINE SCENARIOS THAT LOGICALLY COMBINE DIMENSIONS AND ELEMENTS
- MAP COMPUTER MISUSE MODEL ONTO COMPUTER OPERATIONS MODEL
- RELATE CORRECTIVE/PREVENTATIVE ACTIONS TO COMPUTER MISUSE DIMENSIONS/ELEMENTS
- FINALIZE IMPACT ASSESSMENT FRAMEWORK TO HELP MANAGERS SELECT APPROPRIATE COMBINATIONS OF ACTIONS

COMPUTER SECURITY IN THE RETAIL INDUSTRY



John G. Pricz
Manager
Security Center
Carter, Hawley & Hale

John is a Certified Data Processor, and received a Bachelor's degree in Economics from Rutgers University and a Master's degree from Rider College, Lawrenceville, N.J. He has held various data processing management positions in the areas of systems development, operations, administration and consulting. John is presently Manager, Security, for the Information Systems Division of Carter, Hawley & Hale, where he is responsible for physical and data security.

This presentation describes the basic characteristics of a computer security program in the retail industry. Comparing this program to the requirements in other environments, such as the government sector, may prove helpful in planning, implementing and administering a computer security program.

The following points will be emphasized:

1. The need to adapt computer security concepts and theories to each unique environment.
2. The impact of human factors in a computer security program.

ENVIRONMENTAL FACTORS

A major factor determining the success or failure of any security program to an understanding of the environment affected by the program. As an example of the critical factors to be considered, let me describe the experiences of Carter Hawley Hale Stores, Inc. (CHH) in implementing a computer security program.

CHARACTERISTICS OF THE INDUSTRY

Retail business activities involve capital acquisition, inventory management, payables/receivables accounting, store operations, and employee systems. This highly competitive industry is characterized by significant loss exposure to inventory and cash/credit assets. Widespread use of data processing has created new opportunities for widespread misuse of computers.

Top management is sensitive to the bottom line impact of losses from the dishonest activities of employees and customers. A natural opportunity exists to relate the loss exposure from computer abuse to the acknowledged exposure from inventory theft and fraud.

The day-to-day business activities of most large retailers depend on computer support. If the computer service level is degraded by a security breach, the cash flow and merchandising functions are immediately affected. Again, top management can be taught to understand the relationship of computer security and critical business operations support.

COMPANY PROFILE

CHH is a major North American retailer, operating department stores, high fashion specialty stores, and specialized merchandising operations that market apparel and accessories, home furnishings and books.

The company is represented in many of North America's largest cities, including Los Angeles, San Francisco, New York, Philadelphia, Dallas, Houston, San Diego, Phoenix, Montreal, and Toronto.

Carter Hawley Hale emphasizes fashion and directs its efforts toward upper middle income and higher income customers. The large scale of operations and the customer service implications of CHH's style of merchandising places an emphasis on the integrity and reliability of computer systems.

The privacy of customer data, the need for up-to-date inventory control information, the protection of financial assets, and the support to store-level operations are keys to CHH's continued success and can be enhanced by an effective computer security program.

TECHNICAL ENVIRONMENT

Carter Hawley Hale has a centralized Data Processing Service Center supporting remote input/output facilities located at division sites. An extensive communications network supports these terminals as well as point-of-sale devices at the stores, timesharing terminals, and various on-line/distributed systems. Additionally, some divisions have independent data processing capabilities.

The majority of new systems development is centralized for common systems used throughout the corporation.

This large-scale hardware/software/communications architecture and many unique systems provide a technical environment comparable to many commerical businesses. Due to the diversified systems and applications, security requirements are extensive.

Because the technical environment at CHH is typical of many large data processing installations, it has the same security exposures. The technical management groups proved to be an important part of the team effort to sell the value of a computer security program. Once it became apparent that a security program could help rather than hinder meeting operating objectives, this group helped convince top management and users.

SECURITY PROGRAM

In 1978, a consultant was engaged to review security practices and to assist in preparing a risk analysis for information systems. Specific recommendations addressing immediate and future requirements were presented to management. The program demonstrated the impact of security considerations on policy, organization, personnel practices, systems development, resource management, and controls.

A new organization unit, the Information Systems Security Department, was established to develop and monitor security policies and procedures. Organizationally, it is independent of the operations and development functions. By providing advisory support, training, investigative services, and research, the department helps line managers to carry out their security-related responsibilities. It is also responsible for the administration of the data access control system (passwords), the review of security/controls for new systems development, and the physical security of the Data Center facility.

BUSINESS SYSTEMS

When implementing a computer security program, it is of paramount importance to keep in focus the business objectives of the organization.

Too often, and sometimes justifiably, the technical staff is accused of operating from an ivory tower without an understanding of the real operating world. Most data processing professionals now appreciate how important it is to understand the subtle nuances of the business. Without considering these environmental factors, new systems are doomed to failure through resistance to change, impracticality, or lack of confidence.

Computer security practitioners, take heed.

Translating computer security objectives into management-palatable terminology is also necessary. Removing the technical and security jargon is a formidable challenge, but it is essential to ensure successful communication with management. It is even better to translate objectives into the appropriate business language—for example, commerical or government scenarios.

Here are some other important environmental issues.

- Geographic and demographic characteristics such as time zones and management styles.
- Economic and business conditions.
- Volume and scale of operations differences.
- Systems development philosophy.
- Centralized versus decentralized versus distributed systems characteristics.
- On-line transaction-driven systems versus batch processing requirements.
- Rapidly changing technology.

HUMAN FACTORS

Many "people" issues are addressed during the implementation of a computer security program. These same human factors should be considered when introducing any change affecting an individual's working conditions or the entire organization.

Many textbooks comprehensively address the behavioral considerations of managing change. Any program dealing with security issues and data processing technology is fraught with opportunities for misunderstanding, apprehension, stress, resistance, and negativism. The computer security specialist must have the means to address these problems.

Following are some human factors and organizational considerations:

- Breaking down preconceived notions that computer security is the sole responsibility of the Data Processing staff.
- Tempering the overzealous and motivating the intimidated.
- Establishing the ownership responsibility for information assets.
- Putting teeth into disciplinary and enforcement procedures.

IMPLEMENTATION PLAN

Having been left with a thick volume of the consultant's recommendations, we used the risk analysis to establish priorities for action items.

Organization and policy issues were addressed to provide a mechanism for carrying out the new program. Clearly written guidelines are essential; too much detail is as fatal as not enough.

Although computer security policy should be set forth in general terms and broad guidelines, specific procedures and standards must also be addressed.

Clarifying relationships to existing functions, such as Quality Assurance and Auditing, can ensure their support.

Personnel practices in the areas of recruiting, screening, and training, can be upgraded to lay the foundation for computer security awareness.

Feedback mechanisms such as incident reporting, loss reporting, and statistical analysis are helpful in pointing out the effectiveness of a program and demonstrating results.

It is virtually impossible to institute major changes retroactively. We found that involvement in developing new systems is an effective way to turn the tide. By reviewing Life Cycle documentation and assisting system developers in resolving security and control issues, you can demonstrate the value of designing security into a system rather than retrofitting security features.

The same design principle applies to planning new facilities for data processing. Unless security requirements are established before the blueprints are drawn, physical security safeguards are more expensive and less effective.

RISK ANALYSIS

The value of risk analysis technology for computer security is a controversial issue. Properly managed, risk analysis can help to identify threats to assets. However, if the risk analysis turns into a statistical

nightmare, the results can be confusing. A straightforward approach of using risk analysis to establish relative priorities for action items proved useful to us.

We established three major objectives for evaluating programs—integrity, reliability, and efficiency. Often trade-offs must be established to determine the best overall programs.

COST BENEFITS

Measuring the cost and benefits of a computer security program is an area where we are still seeking answers. Emphasis on preventive computer security measures also complicates measurement.

Why shouldn't the old axiom "If you can't measure it, you can't manage it" apply to computer security? Zero-base budgeting and value-added theory may provide the answers.

CONCLUSION

Our experience indicates that a computer security program can make a positive contribution to the operating objectives of an organization.

The chances for success of any computer security program can be increased by adequately addressing the environmental issues and human factors during implementation.

I hope the review of our experiences is helpful to you in meeting your objectives for computer security. □

COMPUTER SECURITY EVALUATION AND CERTIFICATION

Zella Ruthberg
Staff
Computer Security Management Group
Institute for Computer Science and Technology
National Bureau of Standards

Zella received degrees in Physics/Mathematics from Brooklyn College (M.A.) and the University of Michigan (B.S.). She has worked in the area of computer security for the past six years, and is presently responsible for Federal Information Processing Standards (FIPS) guidelines on computer security evaluation and certification. Zella has organized and co-chaired two national invitational workshops on "Audit and Evaluation of Computer Security." Other projects include providing an NBS peer review to DoD for its efforts to improve its health care delivery through development of standardized computer systems with the DoD TRIMIS program; producing a National Science Foundation-sponsored reference entitled "Software Exchange Directory for University Research Administration"; performing requirements analysis of the Solar Heating and Cooling Data Center at NBS; and editing the Westin report, "Computers, Health Records, and Citizen Rights." She was awarded the NBS Bronze Medal for her activities in the audit/security area in 1979.

INTRODUCTION

Background

I shall be talking about evaluation and certification as we perceive it at the Institute for Computer Sciences and Technology (ICST). First, I would like to give you a little background on this activity. Dennis Branstad mentioned the invitational workshops that I co-chaired on audit and evaluation of computer security. They were co-sponsored by ICST and the General Accounting Office (GAO). The first one, in March of 1977, was exploratory in nature, had ten sessions, and was very productive. The results of the session on standards, for example, were used as a basis by the GAO to produce their audit standards for computer based systems, which went into effect about two years ago. ICST has published a proceedings on the views of all ten sessions at that meeting (NBS Special Publication 500-19). The second workshop, which was more focused, had three sessions on managerial topics and five technical sessions because this way of dividing the problem would give needed emphasis to the managerial as well as technical areas. One of the technical sessions in that workshop produced the basis for Theodore Lee's first paper on operating system evaluation criteria, which he mentioned earlier at this conference. The proceedings of the second workshop are available as NBS Special Publication 500-57. OMB Circular A71, Transmittal Memorandum No. 1 (TM1), July 1978, is the third factor that led to our current evaluation and certification program. TM1 mandated the requirements for a security program in federal agencies plus certification of sensitive applications and their periodic recertification.

The above workshops led to the efforts in ICST to produce a Federal Information Processing Standards (FIPS) guideline on the evaluation of computer security, and the OMB directive reinforced the NBS commitment to initiate its previously planned certification project at ICST.

Definitions

Before I continue, I would like to give a few of our definitions. I would like to define "computer security," "computer security evaluation," "security certification," "computer system," "computer application," and "sensitive application," because all of these terms are very central to an understanding of how we view certification.

In the certification guideline the current definition for *computer security* is: a quality exhibited by a computer system that embodies its protection against internal failures, human errors, attacks, and natural catastrophies that might cause improper disclosure, modification, destruction, or denial of service. We think we have touched all aspects implied by this term, but would like reactions to and comments on this definition.

*Computer security evaluation**, on the other hand, is an independent assessment of how well the controls in the system have protected that system against all of these disasters and catastrophies. *Security certification** includes, in our view, both a technical evaluation as well as the signing of an official statement that approves the security of an entity as weighed against the operational needs and residual security risk. We use the word entity so that we can include software applications or hardware elements in a system.

Computer system is defined as an assembly of elements, including at least computer hardware and usually also software, data, procedures, and people, so related as to behave as an interacting or interdependent unity. A *computer application* is then the use for which a computer system is employed and a *sensitive application* is a computer application which requires a degree of protection because it processes sensitive data or because of the risk or magnitude of loss or harm that could result from improper operation or deliberate manipulation of the application. That last definition comes from OMB Circular A71, TM1.

ICST/NBS CERTIFICATION AND EVALUATION PROJECTS

The ICST/NBS certification project began in 1980 and has continued to the present. I have been the manager and the System Development Corporation (SDC) has been the contractor. The original SDC manager was John Gilligan, the current SDC manager is Daniel Venese, and the SDC principal investigator has been and still is William Neugent.

The certification project consisted of two phases. The first phase was a technology assessment task which took place during 1980 and 1981 with the objective of assessing methods for measuring the level of security. The second phase was the certification guideline task which took place in 1981 and 1982 to the present.

The Technology Assessment Task

The technology assessment information gathering task consisted of reviewing documentation of methods, interviewing developers of many methods, using our own evaluation experience, and examining the influence of the environment, control groupings, data sensitivity, and acceptance criteria.

The major contributions of this technology assessment were the realization that there are three evaluation communities currently in operation: one in the risk analysis area, one in the security review area, and the third in the security audit arena. The conclusions we drew about the reviewed evaluation methods included that there is no widely accepted method. Different methods are useful for different people and different situations, and any method must be tailored, often extensively, for a particular use. The most critical need is for trained and motivated people. Another interesting conclusion was that one can describe a *generic* security evaluation process but that no universal method existed for describing the common elements of these methods. I shall come back to this later in the talk.

The above information on the technology assessment will appear in an NBS Special Publication in FY83.

The Security Evaluation Guidelines

I would like to describe a little of what has been happening with the computer security evaluation guidelines for which I am responsible. Activity in that area has been going on from 1980 to the present with the guideline started in 1981. We expect that guideline to be completed in 1983. The sources for that document are: the two earlier mentioned NBS Special Publication proceedings of our two invitational workshops, the technology assessment that took place for the certification effort, and also a draft document that was reviewed at ICST in January 1981, entitled "Introduction to Computer Security Audit for the

*The revised version of the certification guideline (September 1982) has reverted to the definition for certification in FIPS 39, i.e., certification is a technical security evaluation performed for the purpose of accreditation. Accreditation, also defined in FIPS 39, is therefore implied by this definition of certification, and consists of the management approval for operation of a certified system.

General Auditor." This last report was produced under contract to ICST by William Perry, editor of the EDP Auditor.

The present outline of the "Guidelines for Computer Security Evaluation for Federal Managers" includes an introduction which discusses security activities at the various levels of management, a chapter on the reasons for concern which discusses relevant laws and policy issues as well as many GAO reports, and a computer security evaluation issues chapter which will be discussed in more detail in a few minutes. Those three parts have been written. The rest of the guideline will discuss methods for evaluation, types of evaluation, a computer security evaluation program, and security evaluators.

Computer Security Evaluation Issues.—I would like to return to the computer security evaluation issues chapter in the evaluation guideline. The document attempts to treat the computer security problem in a comprehensive manner and to carry the reader from a view of what is being protected to a view of how well it is being protected. The first section in that chapter treats the protected elements and asks, "What is being protected?" The answer given is that it is sensitive data and sensitive applications. It then cites the need for sensitivity classification schemes and makes some suggestions along those lines. The second section, which I call the control network, answers two questions: "Why is the element being protected?" and "How is it being protected?" This section proposes a particular view of computer security needs, agency control policy, and control implementation. The last question that this chapter deals with is, "How well is that element being protected?" The discussion covers measurement concerns embodied in environment considerations, the need for evaluation criteria, and evaluation evidence forms that one obtains from transaction flow, logging and journaling, testing, documentation, and interviews.

The Control Network (see slide).—The control network view that is expressed in that chapter and shown on the the slide attempts to make the connection between the security needs of the agency and the control techniques that are finally implemented. The agency mission needs, federal computer security policy, and user security needs feed into the totality of computer security needs for the agency in connection with all of its applications. Then, in any agency, there is the unique top management view of their assets and risks—the flavor introduced by the particular people who are managing that particular agency. These two elements, computer security needs and the top management view of assets and risks, feed into the agency control policy. The agency control policy is then expressed in terms of control objectives which are very broad general statements of what should be done in the areas of control. Those are then interpreted in terms of mulitple control technique objectives, which the audit community calls standards, and which would provide the criteria against which one might evaluate the security of a system. These control technique objectives are then translated, in the control implementation, into various control techniques. There are many documents published on the various control techniques that one might employ.

This is a very idealized view of what should be the connections between these elements. The dashed lines in the slide indicate what most often actually happens in an agency. Federal computer security policy is still a very undefined area for most agencies. Agency mission needs may feed directly into the agency control policy and, if there is not an agency control policy, agency managers may unfortunately go directly from computer security needs to the control implementation.

The Certification Guideline

I would like to touch upon the evaluation-relevant areas of the draft "FIPS Guidelines for Computer Security Certification" which was finished in April of this year. The principal author is William Neugent, and I have been the technical manager and director of that activity.

Major Functional Roles.—The persons and their major functional roles that we see for the certification activity are the Senior Executive Officer who issues a directive for certification, the Certifying Official who is responsible for accepting residual risks, the Certification Program Manager who defines the agency-wide security certification program, the Application Certification Manager who manages a specific certification, and the Security Evaluator who performs the technical evaluation. I would like to reiterate our view of certification of sensitive application as encompassing both the technical evaluation, and the acceptance for operation by a Certifying Official—some manager at a high enough level who accepts the residual risk and directs the organization to correct deficiencies.

Major Elements.—The elements of the certification, in our view, are the following: 1) the planning effort for the particular certification; 2) the use of information-gathering techniques to attain information for that certification, i.e., the evaluation information; 3) the actual evaluation, at least a basic high level one, and possibly also a detailed evaluation, depending upon the situation; 4) a documentation of that evaluation; 5) the certification decision; and 6) the considerations for recertification, that is, the conditions for changes under which one recertifies a system. The first four steps are really the evaluation activity.

As I said earlier, we found that all methods of evaluation could be generically described. The steps are: planning, gathering information, analyzing that information, reporting the findings and then asking the question, "Is the information sufficient for the evaluation being performed?" If it is not enough for the particular situation that you have, that is, the particular sensitivity of the application, then one iterates and perhaps does a deeper level of evaluation. If it is sufficient, then one is finished.

The specifics of the *planning activity* for certification consists of an initiation step in which the Application Certification Manager would contact the application sponsor and the Certifying Official and together decide what needs to be done. They would base that on an analysis which looks into the policies, the security requirements, the emphaisis needed, the evidence that might be accumulated and the level of detail. This initiation effort would also need information on the resources needed including the people, the time, the administrative support, and technical tools. All of this would feed into the application certification plan.

We have a sample outline for an application certification plan. It includes the responsibilities of the evaluation team, the support required from other offices, the evaluation products needed, a possible certification statement, and the tasks that are needed to support the technical evaluation.

In the certification guideline we recommend three approaches to *information gathering*: going to the application management to get precise descriptions of what is happening in that system, documentation review (if documentation exists), and interviews with key people. If the information required is for a very sensitive system, there should be at least two sources of the information for a particular application.

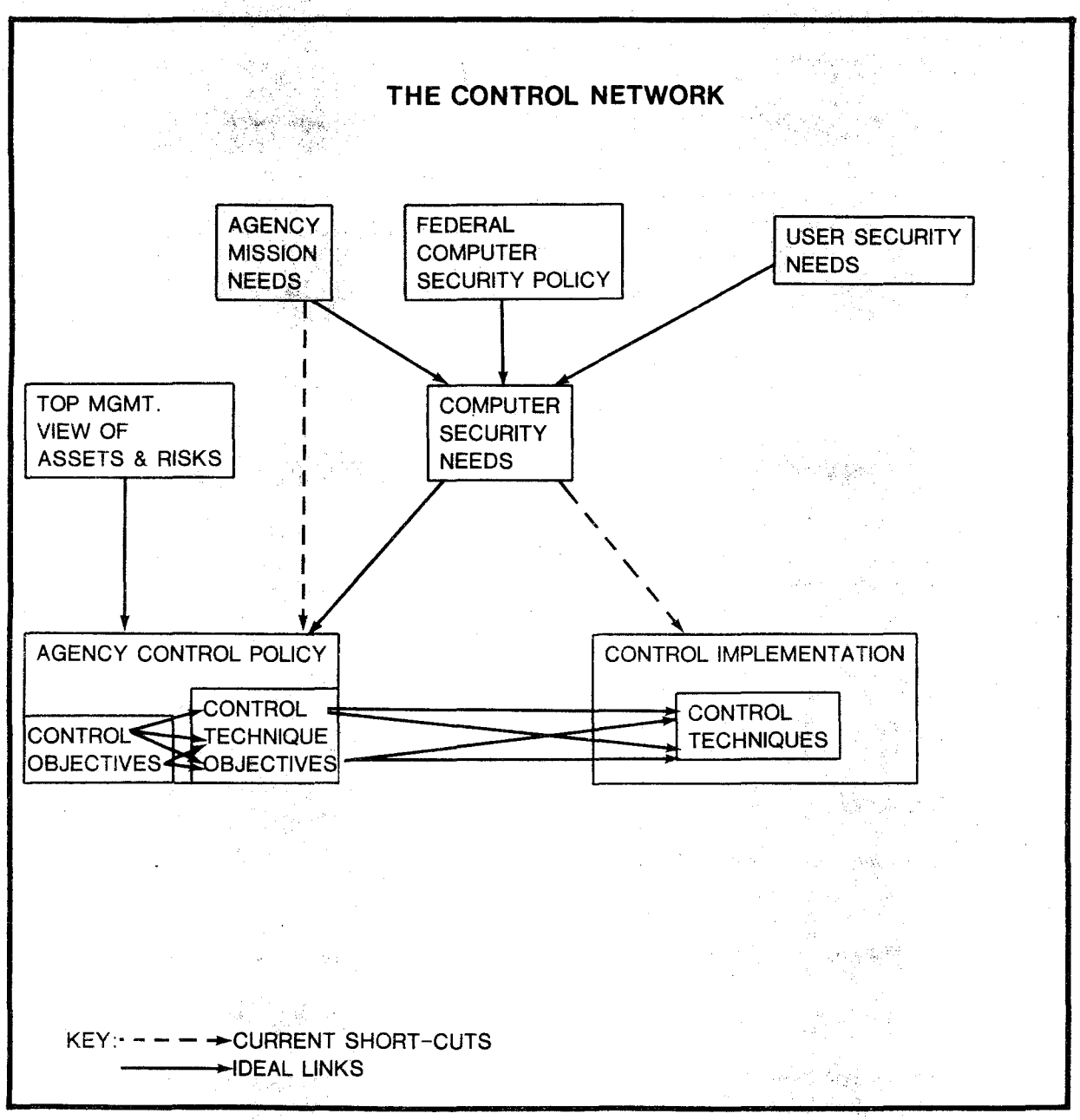
The document describes two levels of evaluation. *Basic evaluation* looks at the entire application but is high level in the sense that it terminates before going into great depth. This type evaluation is sufficient at times for less sensitive systems. It involves security requirements evaluation and asks the question, "Do safeguards satisfy security requirements?" It then looks at a functional level control evaluation asking the question, "Does the control posture meet security requirements?" Then it looks for control existence and asks, "Have controls been implemented?" Finally, it looks at the implementation methodology and asks, "Were the methods used to implement the security functions acceptable?" If the quality of the implementation is questionable, a deeper review may be needed. The factors to look for in a good implementation are good documentation, defining security requirements, project control, good programming practice, and use of well-trained people.

The *detailed evaluation* which we recommend for more sensitive systems or for systems in which primary safeguards are within the computer, looks at: 1) proper control function asking, "Do the controls function properly?" 2) proper control performance asking, "Do the controls satisfy performance criteria such as availability, survivability, accuracy, response time and thruput?" and 3) penetration resistance asking, "How readily can controls be broken or circumvented?" Since detailed evaluation requires a great deal more resources, we recommend detailed focusing on particular areas that the evaluation team feels needs doing and suggest two possible strategies for the detailed focusing. One could be based on the security components, the assets, exposures, threats and control that exist in that particular application. The other one might be based on a situational analysis such as an attack scenario or an analysis of transaction flows. This latter type of detailed focusing provides a detailed, well-understood, example to complement the completed high level review of the basic evaluation.

After all the evaluation activity has occurred, a *security evaluation report* is written and we have a sample outline for such a report. It gives the major findings, the statement of the general control posture, what the vulnerabilities are, the recommended corrective actions and, finally, a proposed certification statement as an attachment to the report. An additional attachment to the report might contain all the detailed evaluation reports. This report would then be sent to the Certifying Official within the agency.

Sensitivity Classification Schemes.—The last item I would like to mention is an illustrative set of sensitivity categories for applications. Such classification schemes are needed by federal agencies. This one is adapted from the paper on policy presented earlier today by Gene Epperly. One might adopt levels of sensitivity such as 1) critical sensitive, 2) non-critical sensitive and 3) non-sensitive in federal agencies. These sensitivity levels might have the characteristics, respectively, of 1) mission critical for the whole agency, or life critical, or an automated decision system with potential for loss greater than 10 million dollars per year; 2) mission critical a for a major element, or having Privacy Law implications, or Freedom of Information Act exemptions, or being used for automated decisions with the potential for loss of 1 to 10 million dollars per year; and 3) all others.

The rest of the certification document discusses the criteria for the certification decision, suggests a form for the certification statement, and gives a complete discussion of recertification and its relation to change control. A correlation between the levels of change to levels of recertification is made. I shall end here, however, since this conference is centrally concerned with security evaluation and I have presented the relevant parts of the certification guidelines. □


THE CONTROL NETWORK
AGENCY MISSION NEEDS
FEDERAL COMPUTER SECURITY POLICY
USER SECURITY NEEDS
TOP MGMT. VIEW OF ASSETS & RISKS
COMPUTER SECURITY NEEDS
AGENCY CONTROL POLICY
CONTROL IMPLEMENTATION
CONTROL OBJECTIVES
CONTROL TECHNIQUE OBJECTIVES
CONTROL TECHNIQUES
KEY: CURRENT SHORT-CUTS
IDEAL LINKS

SAMPLE OUTLINE OF APPLICATION CERTIFICATION PLAN

1. EXECUTIVE SUMMARY
2. INTRODUCTION
   - 2.1 BACKGROUND
   - 2.2 SCOPE
3. RESPONSIBILITIES
   - 3.1 EVALUATION TEAM
   - 3.2 OTHER OFFICES
4. SCHEDULE
5. SUPPORT REQUIRED
   - 5.1 ADMINISTRATIVE
   - 5.2 TECHNICAL
6. EVALUATION PRODUCTS
7. TASKS

APPENDICES

A. CERTIFICATION STATEMENT (S)

B. TOOLS TO SUPPORT TECHNICAL EVALUATION

BASIC EVALUATION

- SECURITY REQUIREMENTS EVALUATION
- FUNCTIONAL LEVEL CONTROL EVALUATION
- CONTROL EXISTENCE DETERMINATION
- IMPLEMENTATION METHODOLOGY REVIEW

DETAILED EVALUATION

- PROPER CONTROL FUNCTION
- PROPER CONTROL PERFORMANCE
- PENETRATION RESISTANCE TO
  - — BREAKING CONTROLS
  - — CIRCUMVENTING CONTROLS
- DETAILED FOCUSING

SAMPLE OUTLINE FOR SECURITY EVALUATION REPORT

1. INTRODUCTION AND SUMMARY
2. BACKGROUND
3. MAJOR FINDINGS
   - 3.1 GENERAL CONTROL POSTURE
   - 3.2 VULNERABILITIES
4. RECOMMENDED CORRECTIVE ACTIONS
5. CERTIFICATION PROCESS

ATTACHMENT A: PROPOSED CERTIFICATION STATEMENT

ATTACHMENT B: DETAILED EVALUATION REPORTS

ILLUSTRATIVE SENSITIVITY CATEGORIES FOR APPLICATIONS (ADAPTED FROM PAPER BY E. V. EPPERLY)

- CRITICAL SENSITIVE
  - — MISSION CRITICAL — WHOLE AGENCY
  - — LIFE CRITICAL
  - — AUTOMATED DECISION WITH POTENTIAL FOR LOSS GREATER THAN $10 MILLION/YEAR
- NON-CRITICAL SENSITIVE
  - — MISSION CRITICAL — MAJOR ELEMENT
  - — PRIVACY LAW
  - — FOIA EXEMPTIONS
  - — AUTOMATED DECISION WITH POTENTIAL FOR LOSS OF $1 MILLION — $10 MILLION/YEAR
- NON-SENSITIVE
  - — ALL OTHER APPLICATIONS

BIZARRE BAZAAR: AN APPROACH TO SECURITY TECHNOLOGY TRANSFER

Clark Weissman
Chief Technologist
System Development Corporation

Clark received a B.S. in Aeronautical Engineering (EE Minor) from M.I.T., and has done graduate work in Computer Science at Rutgers University, USC and UCLA. Clark is one of the nation's foremost experts in computer security, having spent over 22 years in the research, design, and development of security systems for major projects such as the ADEPT-50 Time-Sharing System and the U.S. Air Force's Strategic Air Command 465L program. He is a consultant to the U.S. Government on security assurance testing and certification, and he is an active participant in efforts to formulate Federal Government computer security standards. In addition to his numerous significant assignments and accomplishments at SDC, Clark participated in the 1967 Ware Study on Computer Security, the 1970 AFIPS Security Study, the 1971–1972 Security Study Panel for the U.S. Air Force's Electronic Systems Division, and the 1972 ACM/NBS Security Congress; was Keynote Speaker at the 1972 U.S. Navy Carderock ADP Security Symposium; and was a Co-Chairman of the 1975 MITRE Corporation/NBS Privacy Mandate Symposium/Workshop. He was a National Lecturer during 1967–1968 for the Association of Computer Machinery (ACM); Editor of Operating Systems for ACM Communications 1973–1975; presented two papers voted outstanding presentations at AFIPS Joint Computer Conferences in 1964 on Time-Sharing Systems and in 1969 on Computer Security in Time-Sharing Systems. Clark is in Who's Who in the West, and Science and Technology.

INTRODUCTION

When Mel Klein and Dan Edwards first asked me to speak to you about the future of computer security, I accepted with the intent of pitching some technical views. However, my remarks are more programmatic than technocratic. They are an expression of my long frustration with this business; the excitement of significant technical accomplishments and painfully slow market development. Each year I defend the small SDC R&D budget by noting the "rising tide of Computer Security interest" demonstrated in the growing literature, numbers of conferences—the fifth of these at NBS—and now the DoD Computer Security Center.

I know my frustrations are shared by many of you in the audience. In particular, my long-time colleague Jim Anderson, who will share this session later, has made the profound mistake of letting his frustrations reach print first, in a recent paper entitled, "Accelerating Computer Security Innovations."[1] The paper chastises twelve years of serious security R&D which has yielded only ". . . two one-shot 'brassboard' systems and one commercially supported product. . ." I presume he means KSOS, KVM, and Multics, respectively. But the thrust of the paper is a challenge for us to do things differently. He proposes to stimulate computer manufacturers to build DoD mandatory multi-level security into their commercially supported operating systems with a DoD VHSIC-like program, where the DoD underwrites the investment proportional to the manufacturer's security investment. He challenges his critics to propose alternatives. I like his suggestion; it has helped me focus my ideas, if not differently, then from another perspective. I've titled this perspective, "Bizarre Bazaar" to focus our attention on the need for technology transfer in the 1980s from R&D to real applications. The Bazaar I see as a special community of computer security groups from industry, university, and government tied together in close cooperation by common objectives, tools, and advanced computer networks, sharing secure product development, tool-building, security evaluations and consulting services, and secure end-user applications.

Technology transfer is difficult; it requires the laboratory to push AND the informed program office to pull the security technology into the market. It requires cooperation between buyer, seller and manufacturer. What better place for that cooperation than a Security Bazaar, where the newest technology employs the

oldest marketing technique? A place where current security ideas, tools, and products are on permanent, hands-on display to prospective buyers, critics, competitors, supporters and the curious. A place where technical peer review is reestablished and claims and accomplishments can be resolved by in-depth access to the technology. A place where product evaluation and informed judgement can occur. A place where buyers can try-before-buy and reduce their anxiety over technical risks. A place where we can finally break the cliche, "If you had one I'd buy it—If you buy it I'll make it."

My market is bizarre in that it does not exist at a fairground, or exhibition hall, but employs the very technology it sells. A recursive market? It is a distributed market employing the ARPAnet with security products, tools and services available remotely from network servers. Servers, like Bazaar stalls, would belong to the vendors and be displayed and supported by them. The government would operate the Bazaar, control the net and administer its access. Buyers could fund experiments, demonstrations, and even applications. The Bizarre Bazaar is itself not so bizarre an idea. Much of the technology for the network now exists, most of the security R&D staffs and the DoD services and agencies are on the net now (there is even a Security Forum newsletter on-line), and the cost of adding hosts is rapidly falling with the newer microprocessor-based IMPs and TIPs. This latter point is important as we need to open the net to wider access by the manufacturers and service vendors to encourage their participation and investment in security "wares."

The bulk of my remarks raise a number of important issues for the security community to ponder in security technology transfer: technology and transfer. The issues cover: rethinking technical approaches, overcoming barriers to technology transfer, and the role of the Bazaar in future computer security approaches.

RETHINKING TECHNICAL SECURITY APPROACHES

I have spoken about the "security triad" for a number of years—formal policy models, enforcement mechanisms, and technical accreditation—which must be employed to achieve credible computer security systems. The triad is the basis for the Ted Lee NBS Report[2], Nibaldi security rating scheme[3], and now the proposed Computer Security Center's security evaluation criteria. Let me comment on a number of open issues within the triad framework.

A Plethora of Security Policies

In 1967, alas 15 years ago, I specified the formal security policy for the ADEPT 50 time sharing system, a policy-model called the "High Water Mark," in which the security policy characterized the DoD mandatory security system.[4] From that policy, subsequently called the "Security Condition," has flowed many new models which address other threat environments. After Lampson exposed the difficulty of information confinement in an environment of hostile programs, i.e., Trojan Horses[5], Bell and LaPadula defined the *-property policy as a constraint on the behavior of "untrusted" programs.[6] It constrained such programs from full employment of the Security Condition, and dynamic changes of objects and their security labels, i.e., the Tranquility Principle. It is the basic policy for DoD multi-level security kernels. Dorothy Denning later demonstrated the use of Information Flow models which could detect *-property violations easily at compile-time by analysis of source text specifications and HOL code.[7] That made the *-property even more popular.

The issue I wish to raise is that we have gotten stuck on the *-property and security kernels and we have avoided other policies that are needed. Though it is our first line of defense, there are applications which must violate the *-property for useful work to be obtained. These applications act as Reference Monitors and their policies are formalized for trustworthiness. In fact, each of the kernel programs has spent more effort on policies and methods of trusting the non-kernel "trusted processes" than on the kernel itself. We see examples in the following applications: sanitizers and message filters such as the ACCAT and FORSCOM Guards; clear memory utilities for Periods Processing; label checkers for message routing or text processing in proposed secure local networks; network protocols; all forms of switching and control, from message de-muxing in COS/NFE to line switching in the AFWL Job Stream Separator (JSS); encryption and key management components as in the BCR box, IPLI and other programs; trusted display, edit and release in VIPID-like secure terminals; all forms of trusted utilities such as loaders, editors, language tools; and the

almost untouched area of data base management, including secure message handling policies as explored in the military message experiment (MME) and more recent work of NRL.[8]

Violations of the Tranquility Principle are required for most useful work. Dynamic creation and destruction of objects, or changing the properties and characteristics of objects, is a trusted task for all security kernels, but also for other systems including DBMS and most of the examples noted previously. We need to explore and encourage these policies, and to catalogue them in a library of formal specifications and HOL code for re-use by others.

Over-Emphasis on Operating Systems

Our work on the enforcement element of the security triad has focused too long solely on just the operating system. When economy-of-scale drove us to large central processors, that strategy was valid. But technology has progressed in the past decade to warrant changing our strategy. Economy-of-scale now favors the small personal machine, the special-purpose machine, and the work station on a network. We must continue our emphasis on trusted systems, but trusted distributed systems in a trusted network of trusted components. The problem that emerges is production of trusted software. The hard part of building a secure O/S is the security-vs-performance trade required to design the trusted resource sharing software on current hardware. That hard problem is made unsolvable by demanding that the solution be compatible with the existing product line. If an existing O/S operates at system high or at a dedicated security level, or if there is little or no trusted sharing required, as in a dedicated application or personal machine, the security solution becomes tractable as a trusted network.

The use of static objects and reduced dynamic resource allocation in the UCLA[9], COS/NFE[10], and the SDC Communications kernels[11] accounts for their successful balance between security strength and system performance. The direct SDC experience with these three kernels, and others of a similar "limited O/S" flavor, such as the AFWL JSS, a trusted local net Bus Interface Unit (BIU), and a secure Release Terminal, contrasts to the large O/S for the secure Kernelized VM/370 (KVM)[12], and provides a unique opportunity to confirm the technical speculation: simpler things work better! (And they are more secure.)

Concurrent with these findings is the matching simplification of security requirements by DoD users, that limited trusted security mechanisms can offer sizable and salable cost-benefits. For example, the "Controlled Mode" of AF Regulation 300-8 can be technically satisfied with simpler enforcement mechanisms than full MLS O/S, such as a secure "label checking" BIU of the kind recently suggested in a MITRE paper by Gasser and Sidhu.[13] A trusted security "filter," which acts like a security "impedance" matching device between high and low security level systems, is a practical application of the "secure subsystem" strategy proposed long ago in the 1972 Anderson Report.[14] A number of such "guards" are now under development. One last example—"red-black" data separation in encryption systems—can be satisfied by less than full MLS O/S capabilities. Trusted software and proper hardware to support the "separation" policies are technically within our reach and doable.

Secure data base management is the next big security frontier. The critical requirement is for shared data bases of item-level granularity, and mixed data-item security levels and need-to-know compartments. MLS DBMS issues of aggregation—when an aggregation of unclassified items becomes classified—and statistical inference—when multiple unclassified queries yield classified data—are of growing importance. The trusted enforcement mechanisms do not exist, and their architectures can only be poorly outlined. Much work remains to describe the trust policies and their mechanizations. That work has begun. The Navy has grappled with a special case DBMS, the electronic message and its archive, for a number of years and shows promise of rational solutions. RADC has initiated work with SDC to seek design aids for building specialized, ML DBMS for embedded weapon systems, including special DBMS hardware options. Finally, the Air Force Studies Board and the National Academy of Sciences have summoned a 1982 Summer Study of the problem.

Work on secure O/S must continue as large shared processors are required for many applications. My message is for us to get on with other solutions as well. Many of these limited enforcement mechanisms are all the mechanism needed. They also are near-term achievable. We need as many such "success" cases as we can get, to sharpen our technical skills and build market credibility.

Under-Emphasis on Accreditation Tools

It takes three legs for a tripod to stand. So too for our security programs. Accreditation is the third, and the most misunderstood leg of the security triad. Accreditation is a management decision, by the Designated Approving Authority, that a given computer system in a given application offers sufficient protection for the classified assets in its control. Management makes an informed decision based on the assets at risk, the threat environment, and the security strength of the enforcement system. The assets are National Security data that are visible by their security classification labels. The threat is that the assets will be compromised. Specific threat methods are themselves classified, resulting in a general DoD view of asset risk as "binary"—secure or not secure. There is no figure of merit, or measure of how secure, as in actuarial tables for life insurance. Some years ago, I suggested to Gene Epperly that we might introduce a metric of classified data "worth" by rating the "gap" between security levels, like the electrical potential across a circuit component. We could then build enforcement mechanisms of varying strengths. We are doing just that with security modes, i.e., Dedicated, System High, Controlled, and MLS. In the absence of a true metric of data worth, we are forced to measure the strength of the protection mechanism. The NSA Computer Security Center is charged with that task for commercial systems used by the DoD. Some special systems may also be studied. The issue is how? We have heard present Center plans on that process these past few days. That sensible approach is derived from work by NBS, C^3I, Security Consortium, Mitre, and NSA.

An initial observation is that the evaluation process is directed at too narrow a set of enforcement mechanisms—MLS O/S. My earlier remarks about this over-emphasis on the O/S is relevant here as well. It has been said by some anonymous sage that, "If the only tool you have is a hammer, everything looks like a nail." If the only system we use is an O/S, the only security defense is a kernel. I think we have shown a more mature trusted computing base than that. I would rather see the focus on TRUSTED SOFTWARE evaluation, and an evaluated products list that contains a diversity of trusted systems from a variety of vendors of micros, mainframes, peripherals, communications, software systems and applications.

My next observation has to do with the cost in human resources to do the job of evaluation. I have estimated that just one system, AUTODIN II, consumed many dozens of man-years for government security evaluation. That level of effort is too high, considering the number of systems and products to be evaluated. The evaluated products list (EPL) is one way to address the problem, i.e., distribute the cost over a large number of users of standard products, e.g., MLS O/S. However, that approach forms a bottleneck on the number and types of secure products that can be certified. Another way is to lower the cost to the government for evaluations. There are a number of ways to do that:

- build simpler security products,
- distribute the evaluation process over a broad industry base, i.e., peer review,
- increase the machine-intensive aspects of certification,
- increase the quality of "trust evidence" from the manufacturer.

Simpler products were described earlier. There has been insufficient peer review of security products, particularly in-depth examination of trust evidence. The DoD should find ways to establish this proven technique. Machine-enforced rigor in preparing trust evidence is at the heart of program verification efforts examined more fully below. The Bizarre Bazaar approach caters to all these approaches, and accommodates the EPL as well.

Current verification technology is concerned with either invariant analysis or flow analysis. Under invariant analysis, the security properties are expressed in terms of acceptable or unacceptable states of a system described by its state variables. Verification involves showing that none of the possible system transforms ever cause the system to enter an unacceptable state. A variation involves a set of "guarded" states which may never be entered unless the guard condition, i.e., the predicate, is satisfied. The predicates are proven to be invariants associated with the secure states of the system. Flow analysis demonstrates that information flowing in the system never flows to state variables which are not authorized for such information. The strength of these verification techniques lies in their ability to show that a system design, as represented by its formal specification and its code, satisfies a clearly stated set of security properties based upon a security policy model.

In our R&D program at SDC, we have worked on about a dozen different efforts to apply formal methods, resulting in thousands of lines of formal specifications and thousands of pages of formal proof evidence. What have we learned?

- Using proof tools is reasonably straightforward.
- Building good verification tools is deceptively hard.
- Most of us really don't know how to design systems.

It has been suggested that formal verification is really a "competency test" (some suggest "incompetency"), of our designers, not our designs. Comparing our experience with others shows general agreement on this point. We don't naturally think in terms of "state machines" and the level of specificity they require. The proof tools keep the process honest, and do uncover significant design flaws. However, much of the security strength comes from formal specification writing rather than from the final proofs themselves, because the tools force a more rigorous, often better design.

The final observation from all the above is the need for the DoD to place more emphasis on the tools of secure system development. As I will discuss more fully next, we have over-emphasized secure products and under-emphasized the product-development "process." We encourage secure products for the EPL. We even have a Nibaldi scale for measuring the security strength of these products based in large measure on the trust of the development process. Yet we have no serious tool set for such development! We are in immediate need of these tools:

- formal specification language processors,
- specification verification tools,
- verification condition generators (VCG) for different HOLs,
- integrated specification-HOL VCG,
- interactive, extendable and automatic theorem-proving tools,
- specification, HOL, and proof source library configuration tools,
- cross-compilers for the HOLs for a variety of secure machines,
- good documentation and continuing support for the tools.

For accreditation, management seeks certification evidence that the security strengths match the application risks. The computer enforcement mechanisms implement a reference monitor, and the evidence must show that it:

- is always invoked to enforce the security policy,
- is tamper-resistant by use of a secure system architecture,
- employs appropriate DoD and discretionary security policy.

Security risk assessment, architectural analysis, hardware adequacy analysis, formal verfication proofs, requirement-specification-code correspondence data, penetration studies, and operational experience form the data base of evidence from the certification analysis.

OVERCOMING BARRIERS TO TECHNOLOGY TRANSFER

Research by Dr. Edward Roberts of the MIT Sloan School for the past 20 years has shown that technology transfer is difficult because technology is a local phenomenon, influenced by engineering trades between market demands and constraints that are often different between the sending and receiving organizations.[15] Science, unlike technology, is universal and global. For computer security problems, we have employed the best computer science has to offer in seeking solutions, but we have stalled in technology transfer of laboratory solutions into commercial products for reasons similar to those studied by Dr. Roberts and his associates. A review of some of the prime technology transfer barriers will help support my case for Bizarre Bazaar.

Absence of Strategic Technology Planning

Over the life cycle of a product, the early years are characterized by great innovation in product function and design, and the later years by creative efforts focused on production or "process" innovation, i.e., improving production tools and methods. The middle years see a shakeout of different products until a "dominant design" emerges. Our focus on the commercial O/S for security has come on the scene late in the life cycle of such a product, where the manufacturer is well into production innovations. Yet computer

security demands product design changes for kernels and trusted system components. Furthermore, our production ideas of formal verification are alien to those that made the O/S product successful. The end result is a set of barriers to our security technology. How might DoD overcome these barriers? It must get into the product life cycle earlier by better technology planning that couples more closely with industry planning.

An old Jewish proverb says, "Man plans and God laughs." However, lady luck smiles favorably on those who plan, and all successful businesses plan. Profit is a prime mover of business, so plans reflect actions toward improved earnings. Investment decisions induce technology portfolios to spread the technology and market risks. The DoD is not a major part of the market for most commercial computer products and has little impact on product plans. The hope of the past ten years has been for the privacy and commercial security markets to develop vigorously and allow the DoD to piggyback its requirements on products from that market. It hasn't happened. Those markets are developing quite slowly, moving in directions of more physical and procedural security, and employing untrusted applications software controls. DoD now has to find those products, and/or those companies for which it is a major market force. Focusing solely on large commercial mainframe O/S looks like a losing strategy. However, O/S security in concert with special-purpose trusted security products, of the variety suggested earlier, can produce favorable results.

It would be presumptuous for me to tell DoD how to plan, even if I were capable, but the techniques that are employed in industry do not appear to be in use in the DoD Security Initiative. The DoD Computer Security Consortium has market clout and can leverage that muscle into good security products if it gets its house in order. The formation of the Computer Security Center is a good sign, but I don't see in the structure the planning functions needed to closely couple DoD goals and plans with the strategic planning activities of industry. For example, where are the Technology Planning Units (TPU) articulated? TPUs are the intersections between DoD market requirements and the existing technologies: the clusters of common interest between the R&D and Consortium communities. For each TPU there can be developed a set of Critical Success Factors (CSF), which Dr. John Rockart of the MIT Sloan School describes as ". . . factors that must be manifest for a company to achieve its overall objectives." CSF can be measured, and tracked over time as Technical Progress Functions of price and performance. Strategies of investment can ride these "learning" curves; riding constant price over the years yields a product with increased functionally, whereas riding the constant function curve yields a standard product of lower cost. The first is a Production R&D investment strategy, the other a Process R&D investment strategy. Various combinations are also considered. DoD is both the market and the investor for much of the security business, and yet it is silent on these critical technology planning factors. If industry is to be called upon as a mature partner in production of secure products for the DoD, this level of planning must exist in DoD and admit contributions from industry in a coopertive manner. And industry can and must cooperate in both Product and Process innovation!

Industry Disincentives

Advanced technology carries high risk. Much of our industrial society is based on risk avoidance, which shuns high technology programs. Computer security technology not only carries that burden, it can't even hold out the promise to industry of large production contracts downstream as an inducement to take risks. Furthermore, the security triad demands both product and process innovation with little incentive other than an implied promise that a good grade on the EPL will result in future sales. There are only negative incentives for process inventions, since those inventions must belong to the government for product maintenance and improvement. There's a double "whammy": if you don't make products but make only process tools, you lose; and if you do make products, someone else gets the easy, low technology-risk O&M contract. Where's the incentive for industry to invest its critical human and financial resources? And those resources are huge. At a recent congressional hearing, Richard DeLauer, Under Secretary of Defense for Research and Engineering, noted that some 250 defense contractors received $1.2 billion dollars in 1979 IR&D reimbursements under PL 91-441. That funding level does not include R&D invested for commercial products. If DoD contractors had spent only 1% of their IR&D for computer security over the decade, it would have resulted in more funds than a decade of direct DoD contract and in-house support! Industry has the muscle, DoD has the problem. To attract the attention of industry moguls and their technology planners, one needs incentives, not disincentives; "honey not vinegar."

I believe "visibility" is the key. DoD must expend more attention to marketing its needs in ways industry can hear. It must build, through the Security Consortium, a five-year security plan of technology and weapon system procurements for which security is the driving force. Those plans should list expected budgets for the procurements. There should be open meetings and briefings on the plan, with yearly updates. Since the data is all in open budgets anyway, no data compromise is effected, but industry is informed in an attention-getting manner. The Bizarre Bazaar is another visible way to reach industry through its technical community and its information "gatekeepers," in a manner similar to the mail, newsletters, reports, and administrative information which flows over the ARPAnet. As a testbed of ideas and products a security culture will expand from the Bazaar.

Improving DoD Acquisition Policies

The Armed Services Procurement Regulations, ASPRs, define eight contract types, which parallel the life cycle of product development. These correspnd to the DoD R&D Program Elements 6.1 to 6.5.

1. Research	(6.1)
2. Exploratory Development	(6.2)
3. Advanced Development	(6.3)
4. Concept Formulation	
5. Contract Definition	
6. Engineering Development	(6.4)
7. Operational System Development	
8. Management and Support	(6.5)

Projects in 6.1–6.3 budgets tend to be generic technology programs of military relevance, whereas, 6.4 and 6.5 funds focus on specific DoD mission programs.

Most computer security funding in the past has come from generic R&D budgets, and the technology has not migrated into the mission programs successfully. DoD and industry mis-management have contributed to this technology transfer failure, particularly the misue of scarce resources in these generic R&D programs. I am particularly troubled by competitive "flyoffs" for 6.3 funds, as with KSOS. Prototypes are not operational mission systems, and they should not be procured as such. It was premature and divided, what small R&D community there was, against itself. It spread critical talent too thin, and cut off needed technical cooperation by industry and government. A better idea then and now is a greater number of sole-source contracts for different architectures—KSOS vs PSOS vs KVM etc.—a competition of ideas and prototypes in an open intellectual bazaar that encourages in-depth peer review. It may prove less expensive and more expeditious.

"Sole-source" is not a dirty word, and can be applied where appropriate, as in a security program where the technical resources are unique. The governing policies and regulations, OMB A–109 and DoDD 5000.1, are currently in revision to reflect the 1982 DoD Authorization Act. The new act continues to encourage competitive procurements, but it states that unsolicited proposals are possible when justified by the unique capability of the contractor. The new act is also favorable to R&D programs. For example, it raises from $100,000 to $5 million the threshold for R&D determinations and findings (D&Fs) at the Secretarial level. It also liberalizes patent rights for contractors on R&D programs.

Deputy Secretary of Defense Frank Carlucci has started a series of initiatives to improve the acquisition of weapon systems.[16] Though most of the 32 initiatives enhance the procurement for large production contracts, they are relevant to our interests in security technology transfer to operational systems. Specific items of note are: multiyear procurements, encouraging contractor capital investments with risk rewards and liberal data rights policies, funds budgeted for technology risks, adequate front-end funding for test and evaluation (T&E), deemphasis on low cost in contractor selection, incentives for improved reliability and support.

These new procurement changes are positive actions by the DoD to improve business and cooperation. It can be the beginning of the "honeymoon" for restoring incentives to industry to invest in secure systems.

SUMMARY: ROLE FOR THE BAZAAR

In concluding my remarks, I think it only fair to subject my proposition to my own criticism. Does the Bazaar advance security technology solutions and their transfer to useable products and processes?

Technology Issues: Policies

The Bazaar will be an on-line demonstration and testbed for a spectrum of trusted processor policies. These will be available for analysis and review. A library or data base of such materials could be assembled from the requirement, specification and HOL texts.

Technology Issues: Enforcement Mechanisms

Bizarre Bazaar works best as a testbed for these mechanisms. Systems which wish to operate at a dedicated classified level could explore isolation devices such as label-checkers, one-way links, filters and encryption boxes. Special-purpose O/S can be used as controllers for these boxes, and as the basis for local net secure BIU's, gateways, and secure NFE's. MLS O/S like Multics, KVM, KSOS/11, KSOS/SCOMP, COS/NFE, SACDIN/NFE, etc. can be available as service hosts in the Bazaar. These would require appropriate network boxes to secure the links as well. Use of these MLS systems could range from testbed for prototype applications, to overhead facilities for classified contracts for tool use and for administrative data management. The later application could explore new ideas and products in secure DBMS, initially as encrypted files, and hosts for dedicated backend data base machines. Later, as MLS DBMS appear, they will provide the MLS O/S support required. The specific interest in secure message systems should see wide use of the testbed facilities of the Bazaar. The secure message DBMS work couples well with potential products in trusted terminals and work stations for message and text processing, release stations, and sanitizer/filter applications.

Technology Issues: Accreditation Tools

Evidence generation and scrutiny are advanced by the Bazaar. Data bases can be created for the requirements, specifications, code, proofs, and analysis reports for the security evaluators and for peer review. These same materials will aid in increasing verification and state machine design literacy by making them widely available in a convenient manner. Evidence will be generated as the projects advance, so these data bases form a snapshot progress report, and become a vehicle for peer review and design analysis. Initially, there can be verification servers available on the net for all the popular verification systems. These can grow with their ability to satisfy the user community. They certainly will assist the Computer Security Center in its efforts to evolve an integrated formal development tool suite that will eventually service the Bazaar. These tools will be available to all on the net, including system manufacturers for development of new security products. In this manner, the technology transfer efforts do begin early in the product life cycle. I cannot stress enough the need for in-depth peer review of the certification evidence. That process has been all but absent, except for some contract-specific reviews. The Bazaar can take peer review further by permitting repeatability of the evidence generation by another independent organization; a hallmark of scientific scrutiny.

Technology Transfer: Planning

Visibility is the principal benefit of the Bazaar. It allows people and ideas to circulate rapidly. Data on learning curves can be compiled as a Consortium planning data base, and strategic objective can be discussed and communicated.

Technology Transfer: Disincentives

Overcoming barriers for contractor and DoD personnel to form more cooperative relationships is the best asset of the Bazaar. For industry, the "honey" can be an on-line data base of Consortium security programs, past and future, including releasable contract data such as SOW, contractor, budget or award price that will show visibility of the market to encourage industry to play. The "best available technology" (BATs) paper by Carl Landwehr is a good start for that kind of material.[17] It should be augmented by financial data, and with data on future procurements.

Lavish support on those contractors willing to take the risks inherent in the security business. That was the message in Jim Anderson's proposal as well. The construction of the Bazaar would be a start in that direction. That can prime the security IR&D pump for those same security contractors. IR&D funds can be used to support the company's "booth" at the Bazaar—host servers for Ina Jo* Formal Specification

*Trademark of System Development Corporation.

Language, COS/NFE, Multics, KSOS, GNOSIS, etc. I challenge industry to cooperate in this security initiative by:

- making sincere attempts to understand DoD security needs,
- making an investment in security R&D,
- being more open with the results of their security R&D,
- joining the Bazaar.

The Bazaar can support the University to train our future security staffs. The Bazaar can allow access to the best technology for student training. Security verification teaching efforts using remote network tool access at UCLA, UCSB, UofT, Mitre, and elsewhere can form the basis for Bazaar-based security courses. Unlike other nets, our security technology should be capable of withstanding student attack.

Technology Transfer: Acquisition Policy

The Bazaar can't make policy, but it can support existing policy in new ways. For example, sole-source justification may be more easily satisfied if polling the net for a specific need yields one supplier, or if peer review of a SOW indicates only one contract source is available. ARPAnet has been used extensively for CDRL distribution and review, as well as for collecting cost and administrative data. As a natural testbed, the Bazaar enhances the Carlucci initiatives regarding built-in, up-front T&E efforts. It should be possible to enhance those efforts with common tests scenarios and data bases. Lastly, the Bazaar could be a vehicle to limit "limited rights" clauses in contracts. For example, a contractor might be willing to make available to the DoD some proprietary product if it were limited to access only from the Bazaar.

Technology Transfer: Cost

Using a 100-node model of the Bazaar, I estimate an incremental cost to DoD of about $1 million a year to cover IMP/TIP depreciation plus 56kbps line leases for 50 nodes not currently ARPAnet hosts. For the contractors, a 25% share of a VAX-level host for the Bazaar results in $50,000 per year of computer cost plus labor of an equal amount. Most major DoD contractors can afford the $100,000 IR&D investment for security R&D laboratory support.

Overall, the Bizarre Bazaar could be operating within a year, since many of the R&D community are already on the net. The major increased use would come from the military users and the vendors. Let's get on with it.

REFERENCES

[1] J. P. ANDERSON
"Accelerating Computer Security Innovations," Proc. IEEE Symposium on Security and Privacy, April 1982,

[2] T. M. P. LEE, Chairman
"Processors, Operating Systems and Nearby Peripherals," in Part VIII of AUDIT and Evaluation of Computer Security II; System Vulnerabilities and Controls, Proc. of NBS Invitational Workshop, November 1978.

[3] G. H. NIBALDI
"Proposed Technical Evaluation Criteria for Trusted Computer Systems," M79-225, The Mitre Corp., October 1979.

[4] C. WEISSMAN
"Security Controls in the ADEPT-50 Time Sharing System," Proc. AFIPS 1969 FJCC, Vol. 35.

[5] B. W. LAMPSON
"Dynamic Protection Structures," ibid.

[6] D. E. BELL and L. J. LAPADULA
"Secure Computer Systems," ESD-TR-73--278, Vol. I-III, The Mitre Corp., November 1973—June 1974.

[7] D. E. DENNING and P. J. DENNING
"Certification of Programs for Secure Information Flow," Communications of ACM, Vol. 20, No. 7, July 1977.

[8] C. E. LANDWEHR
"Assertions for Vertification of Multilevel Secure Military Message System," ACM Software Engineering Notes 5, July 1980.

[9] G. J. POPEK, et. al.
"UCLA Secure Unix," Proc. AFIPS NCC Vol. 48, 1979.

[10] G. GROSSMAN
"A Practical Executive for Secure Communications," Proc. IEEE Symposium on Security and Privacy, April 1982.

[11] D. L. GOLBER
"The SDC Communications Kernel," Proc. of the Fourth Seminar on the DoD Computer Security Initiative, August 1981.

[12] B. D. GOLD, et. al.
"A Security Retrofit of VM/370," Proc. AFIPS NCC Vol. 48, 1979.

[13] M. GASSER and D. P. SIDHU
"A Multilevel Secure Local Area Network," Proc. IEEE Symposium on Security and Privacy, April 1982.

[14] J. P. ANDERSON
"Computer Security Technology Planning Study," ESD-TR-73-51, ESD/AFSC, October 1972 (NITS AD-758-206).

[15] E. B. ROBERTS
"Generating Effective Corporate Innovation," MIT Technology Review, November 1977.

[16] F. C. CARLUCCI
"Acquisition Improvement Task Force, Final Report," Acquisition Improvement Steering Group, D.D. December 1981.

[17] C. LANDWEHR
"Best Available Technologies (BATs) for Computer Security," NRL Report 8554, 1981. □

MEETING POLICY REQUIREMENTS USING OBJECT-ORIENTED SYSTEMS

Susan A. Rajunas
Technical/Staff
MITRE Corporation

Susan received a B.S. in Astronomy and an M.S. in Computer Science from Boston University. She worked eight years at M.I.T. Lincoln Laboratory designing and building applications systems to analyze real and simulated radar system and component design data, and in the process became familiar with a wide variety of programming languages, operating systems, and computer architectures. Susan has over two years experience with IOCS (a DoD contractor) performing independent validation of a radar controller subsystem, and installation and testing of a message-handling system. She joined MITRE three years ago, where she has monitored the KSOS-11 and KVM, analyzed various operating systems, and evaluated commercial operating systems.

INTRODUCTION

I have been involved since 1979, first working for Steve Walker and now working for the DoD Computer Security Center, with capability-based operating systems (IBM's System/38, Tymshare's Gnosis, and Intel's iAPX 432). Despite their excellent protection features, it was not obvious how DoD multi-level security policy could be implemented on these systems. Since policy support is crucial in any evaluation conducted by the Center, I organized a workshop on the subject and part of my talk today will cover the results of that workshop.

Since I originally worked designing and building applications systems, I am aiming my talk today particularly at those who have requirements (including security requirements) that will be realized in an application system. These people may not have been previously aware of the approach I will describe. It is an interesting coincidence that the May issue of IEEE Computer magazine is devoted to articles on specifying user requirements for applications systems.

BASIC CONCEPTS

Objects, as I use the term, are the objects of data abstraction, not the "subject references object" in the reference monitor context. I am referring to data structures and the operations on them. Capabilities are the unforgeable tokens of access by which objects are addressed. A capability is an abstraction combining addressing and access rights, which is one of the reasons why the computer security community has been interested in capability-based systems. (Capabilities are minimally represented by a unique identifier and some attributes, such as access rights.) Domains, in capability-based systems, are sets of capabilities which form access environments for processes, or instantiations of processes.

Addressing objects by capabilities minimizes the opportunity for human error. The level of the architectural interface is raised above dealing with bits, bytes, and words. The complexity the designer must handle is reduced; he is freed to think in terms of objects, not the bits that represent them. The design can now be organized around the requirements of the mission. It need not be organized around the quirks of the hardware.

I should also mention another feature of object-oriented, capability-based systems which attracted the interest of the computer security community. The principle of least privilege, the idea that no process should have more privilege than it needs to perform its function, is an important concept in security. In conventional architectures, a domain is a privilege state, and there typically will be two or three states: user/supervisor, or user/supervisor/kernel. Rings offer more states, for example four in SCOMP and eight in

Multics. However, the possible privilege states are few in number and rigidly hierarchical. And if a process requires even one privilege of a state, it will get ALL of them; this problem is called the pervasive privilege problem. It is important to note that capability-based systems have an arbitrary number of domains (privilege states).

Now that I have mentioned some of the benefits of capability-based systems, I should mention some of the perceived problems. A capability by any other name could be a token, a ticket, a key, an access descriptor. But whatever the name, it is a distributed control mechanism, so there are dynamic access control problems concerning the propagation of rights, their review and revocation, and their migration beyond their sphere of definition. The issue of migration brings us back to another problem I previously alluded to, that capabilities must be unforgeable. By some means, which may be tagged memory words or representation in separate segments, or even encryption, the representation of the tokens-tickets-capabilities must be protected. These difficulties must be balanced against the difficulties inherent in list-oriented control, which is centralized, and entails overhead for each use, as well as presenting problems concerning the representation of the list.

LEVELS OF ABSTRACTION

At this point let me mention that I do not believe that even a capability-based system will be without lists, since lists are natural to the system's human interface. This diagram shows how I see the relationship. I believe that, given appropriate mechanisms to encapsulate capabilities, policy, along with any other application requirements, can be implemented on the left hand side of this diagram.

It must be recognized that a distributed abstraction (capabilities) may well be realized in terms of a centralized mechanism. However, I do not believe that it is appropriate to implement policy at the level of the most basic mechanism in the system. Policy concepts have meaning at the requirements level.

ACCURACY OF THE REALIZATION

I sometimes describe the overall degree of a system's security as a measure of the accuracy with which its high-level abstractions, its semantic fictions, are realized in terms of the basic mechanisms. As an example, consider VM/370. I picked VM since I worked with it for many years, and because it is a system with an explicit and useful semantic fiction, which is that each user has his own 370. Anything an operator can do at the system's console, a user can do at his "virtual" console; there was a useful and obvious way to reason about the system. However, when we look to see how this abstraction was realized in VM's control program, which is a resource manager for CPU, memory, and I/O devices, we discover a contradiction. The isolation of virtual machines is accurately reflected in memory management by virtual addresses; but when we look at I/O channels, we discover that they address absolute memory locations, thus providing the loophole that is the traditional penetration path in VM/370.

While it is true that many systems do not present as consistent a semantic fiction as VM does, which can be analyzed until a contradiction is revealed, the inconsistencies present even more obvious loopholes.

APPLICATION SYSTEM

In the case of building an application system, I see two levels of realization. The application designer maps the requirements to objects and operations, and the operating system nucleus or firmware designer maps the objects to the actual bits and bytes and words of memory.

THE WORKSHOP

Now that I have provided some background on the problem, I will discuss the Workshop on Implementing DoD Multi-Level Security Policy on Capability-Based Systems, which was held at Ft. Meade, Maryland on 23–24 March 1982. The workshop was concerned with operating systems using capabilities, policy models, and their relationship to the evaluation criteria.

The workshop, which I organized at the request of, and with the assistance of, the Standards and Product Evaluation branch of the DoD Computer Security Center, was motivated by the on-going preliminary evaluations of capability-based systems. Mechanisms to support policy had been suggested, and the

applicability of the evaluation criteria, as well as specialized policy models, needed investigation. The objectives were judgements on the effectiveness of the mechanisms and policy models, and recommendations regarding the evaluation criteria.

The workshop was attended by representatives from NSA, the Navy, DOE, Mitre, and private industry. The speakers were Dan Edwards (NSA/C11), Susan Rajunas (Mitre), Earl Boebert (Honeywell), Norm Hardy (Tymshare), George Cox (Intel), Marv Schaefer (NSA/C1A), Stephen Dahlby (IBM), Dan Nessett (Lawrence Livermore National Laboratory), and Carl Landwehr (Naval Research Laboratory).

CONCLUSIONS

It would be nice to say that definitive conclusions were reached, but only limited consensus was achieved. It was agreed that specialized, application-specific policy models deserved more work. In the limited time available, it was impossible for participants unfamiliar with specific systems to analyze the proposed mechanisms and judge their effectiveness. The discussion of the evaluation criteria yielded the conclusion that the criteria should apply to the application as well, if policy is implemented there.

MEETING POLICY REQUIREMENTS

I propose a pragmatic approach to meeting policy requirements. This approach was not feasible in the past; it requires an object-oriented architecture or operating system nucleus that provides a complete and consistent object interface (the semantics in which the application designer works). Using this approach, the user's requirements, including the application-specific policy requirements, are realized in terms of objects and the operations on them. The question of assurance that the requirements are met by the system as implemented now decomposes into two levels. The first is the mapping between the requirements and their realization in terms of objects; the second is the mapping (performed by microcode or operating system nucleus) of the objects to their ultimate representation.

THE POLICY IMPLEMENTATION DILEMMA

Given the lack of consensus from the workshop, we are left with the initial dilemma. We must implement an application involving policy in order to demonstrate the feasibility of the approach. But before we can begin an implementation, we must demonstrate the feasibility of the approach.

Given this situation, I encourage application designers to try this approach. I encourage formalists to work models which reflect user requirements. And I encourage end users to be aware of this approach. □

Introduction

Workshop

- Motivation and Objectives
- Summary

Relevance to Real World Problems

Basic Concepts

- Objects—Data Abstraction
- Capabilities—Unforgeable Tokens of Access
- Domains—Sets of Capabilities
  Access Environments

Addressing Objects by Capabilities

- Minimizes the Opportunity for Human Error
- Raises the Level of the Architectural Interface
- Reduces the Complexity the Designer Must Handle
- Allows the Design to Correspond Closely to the Requirements

Principle of Least Privilege

Domain = A Privilege State

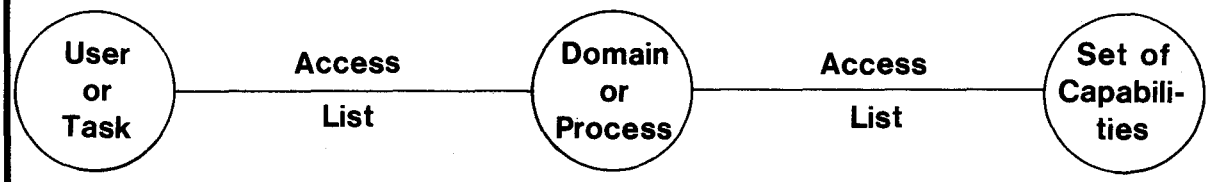
Capability-based Systems Have an Arbitrary Number of Domains

No Pervasive Privilege Problem

Token-oriented vs. List-oriented Control

Dynamic Access Control	Overhead for Each Use
Representation of Tokens	Representation of List

Levels of Abstraction



Under Control of System Administrator

Under Control of Operating System (or Microcode)

Faithfulness of the Realization

System Mechanisms Must Accurately Realize High-Level Abstractions

Loophole in VM/370-I/O Not Virtual

Semantic Fictions May Be Implicit

Inconsistencies Present Even More Obvious Loopholes

Application System

↕

OBJECTS

↕

BITS

By Application designer

By OS nucleus or Firmware Designer

Workshop on Implementing DoD Multilevel Security Policy on Capability-Based Systems

23-24 March 1982 Ft. George G. Meade, MD

Topics: Proposed Mechanisms to Support Policy

Policy Models

Evaluation Criteria

Workshop Motivation

Preliminary Evaluations

Mechanisms to Support Policy

Applicability of the Evaluation Criteria

Specialized Policy Models

Objectives

Judgements on the Mechanisms and Policy Models

Recommendations Regarding Evaluation Criteria

Conclusions

More Work Needed on Policy Models

No assessment of Proposed Mechanisms

Criteria Should Apply to Application as Well—If Policy is Implemented There

Meeting Policy Requirements

Application—Specific Policy Requirements Described in Terms of Objects and Operations

Operating System Nucleus or Architecture that Supports Objects

MULTI-LEVEL SECURITY TODAY

Lester J. Fraim
Manager, TCB Development
Honeywell Federal Systems Division

Les received a B.S. in Applied Mathematics/Computer Science from California State Polytechnic College, and an M.S. in Technology of Management from The American University, Washington, D.C. He began his career with RCA in 1970 as an operating systems specialist, and later spent two years as a statistical analyst in the Office of the Secretary of Defense. Les joined Honeywell in 1973, where he has held positions as systems analyst, system supervisor, Manager of WWMCCS Site Support, and Manager, Communications Systems. In 1979, he was assigned responsibility for the SCOMP program, and is now overseeing the transition of the R&D program to the development of the SCOMP product.

BACKGROUND

The development of computer systems capable of processing multiple levels of secure (MLS) data has progressed from research activity to the development of production prototypes. During this period of development there have been several major programs supported by the Department of Defense to meet the operational requirements of the defense and intelligence communities. Some of these efforts are continuing today to provide the basic environment for processing specialized multi-level applications.

In 1978, the DoD Computer Security Initiative was established to promote the widespread availability of "trusted" ADP systems for use in the DoD. A key element of the initiative's goal is the use of commercially developed, trusted ADP systems. Several manufacturers are now working on the development of systems which will meet the various requirements of the DoD. Some of these manufacturers presented their efforts at the fourth seminar of the DoD Security Initiative Program at the National Bureau of Standards on August 11–13, 1981.

In January 1981, the DoD established the Computer Security Center as part of the National Security Agency. The Center is responsible for the evaluation of computer systems' security for the Department of Defense, and for the continuation of the Computer Security Initiative program. This organization will have a great impact on the success and use of systems developed by commercial manufacturers. The Center's ability to disseminate timely information on the capabilities of the evaluated systems, and to provide a streamlined evaluation process, will enhance the manufacturer's desire to produce these systems.

Since early 1975, Honeywell has been working with the DoD and the Naval Electronics System Command (NAVELEX) on the development of the Honeywell Secure Communications Processor (SCOMP). SCOMP was originally part of a larger program, called Project Guardian, which was to investigate and enhance the security of the Honeywell Multics system. When Project Guardian was cancelled, Honeywell and the government felt that the secure front-end development should be continued. The SCOMP today is significantly different from its original design. It is a multi-purpose computer system, rather than strictly a secure front-end processor.

The following describes the SCOMP mechanisms and their use as a base for implementing applications in a multi-level security environment.

BASIC SECURITY MECHANISMS

The basic concept implemented in the development of many trusted systems is the reference monitor. The reference monitor is an abstract mechanism that controls the flow of information within a computer system,

between subjects (active system elements such as processes or users) and object (units of information or data). The reference monitor enforces a specific security policy, which in the case of the DoD policy has been formulated into a mathematical model. One such model is the Bell and LaPadula model developed by the Mitre Corporation [1].

The basic rules enforced by the model are simple security *-property. Simple security does not allow the reading of data, if the level of the data is higher than the level of the reader (process); the *-property does not allow the writing of data, if the object receiving the data is a lower level than the writer (process) of the data. In addition to the security rules, which prevent the disclosure of data, a similar set exists to prevent the corruption of data. These rules, called simple integrity and the integrity *-property, control the modification of data.

The implementation of the reference monitor, which enforces these rules, is called the security kernel. The implementation of the reference monitor for SCOMP is a unique combination of hardware and software; it provides 1) complete validation of access between subjects and objects, 2) protection against modification of the security kernel or its data, and 3) an implementation based on a design that has been formally verified against a model of the DoD security policy.

SCOMP satisfies the complete validation through the interaction of the software security kernel and the SCOMP unique hardware mechanism. The software provides the initial validation and develops a database in the form of a descriptor for the hardware's use in the continued validation of the access. The isolation mechanism in SCOMP is provided by the implementation of a Multics-like ring mechanism. SCOMP supports four rings, with ring 0 containing the security kernel. Controlled ring-crossing is provided to allow less privileged software to access an inner ring for a service function. The verification property is provided by the use of SRI International's Hierarchical Development Methodology [5]. This method of formal verification requires the development of a top level specification (TLS) in the language SPECIAL. The SCOMP TLS formally specifies 38 software and 12 hardware user-visible functions. The TLS is then verified against the Bell and LaPadula model of DoD security policy. Any false theorems must be corrected or resolved by manual methods. The state of the verification art does not, as yet, provide the system builder with the tools to verify the implementation of the security kernel.

SCOMP IMPLEMENTATION

The SCOMP implementation is a combination of special hardware and system software implemented on the Honeywell Level 6 minicomputer. The Level 6 is a bus-structured 16-bit minicomputer. All system components connect to the Level 6 bus, allowing access to the various I/O controllers, processors, and memory. One of the goals of the SCOMP hardware design was to use all standard peripherals and provide the security mechanisms totally through the special hardware. This makes the conversion of a standard Level 6 to a SCOMP the simple operation of removing the standard processor and adding the SCOMP processor and the Security Protection Module (SPM).

The Security Protection Module enforces the complete validation and isolation properties of the reference monitor and provides several performance advantages. The SPM resides on the Level 6 bus between the SCOMP processor and all other system elements. This enables the SPM to capture all processor requests and perform the required validation prior to accessing memory or I/O devices. Figure 1 shows the placement of the SPM on the bus relative to the other system components.

Mediation Mechanisms

The SPM mediates access to objects using virtual addresses and a process identifier called the Descriptor Base Root (DBR). The DBR points to the memory descriptors and I/O descriptors for the resources available to the process. If the process requests an action to be performed, the SPM mediates it using the information in the appropriate descriptor. If the request is valid, the SPM maps the virtual request to a physical request and allows the action to take place. If the action is not allowed, the SPM generates a trap, which is processed by the security kernel.

The SPM supports a segmented virtual memory organization. A segment is treated as an object and is variable in size, with a maximum size of 2K words. A segment may be of zero size. The virtual memory for

a process is established by assigning a DBR to each process. Each memory descriptor, pointed to by the DBR, contains a pointer to physical memory, access permissions, and memory management data. Each process may address a million-word virtual address space (512 segments). Descriptors may be paged to a maximum of three levels, allowing the interprocess sharing of page descriptors (with access control specified at the segment level). This provides SCOMP with a simplified memory management capability, with reduced descriptor storage.

All I/O requests are captured by the SPM. The SPM performs the mapping of the virtual name to a physical device. The SPM provides for two types of DMA transfers. Premapped I/O is accomplished by the SPM mediating the device and memory resources and then initiating the transfer using an absolute address. Subsequent requests by the device to memory are made with no SPM intervention. Mapped I/O is similar, except that the device controller is provided with a virtual memory address, and each request by the device for memory is captured and mediated by the SPM. The mapped I/O mechanism requires more overhead. However, it reduces the risk of error created by an I/O device hardware failure, and reduces verification requirements on controllers.

Isolation Mechanism

The Multics-like ring structure in the SCOMP supports four rings. Ring (0), the kernel ring, is the most privileged ring and ring (3), the user ring, is the least privileged. The hardware supports a call/return mechanism which provides the capability for a less privileged procedure to request a service from a more privileged procedure. The hardware also supports a mechanism which allows a called procedure to access caller-supplied arguments at the level of the caller. This is called the "argument-addressing" mode.

Performance Mechanism

The performance of trusted systems has always been a major concern. Many of the early attempts to provide the reference monitor functions in software have proven to be extremely slow. The KVM370 and Ford KSOS-11 programs have both produced systems which at best are many times slower than similar systems without security mechanisms [2].

The major reason for implementing the SPM was to build those functions in hardware which can enhance the performance of a trusted system. This is not to say that there will not be a degradation due to the hardware and software needed to enforce security. However, the magnitude of this degradation should be greatly reduced. The SPM has several capabilities which are specifically designed to reduce the overhead of the security mechanism.

One of these capabilities is the use of the DBR to define a process descriptor tree in memory. This approach allows the SPM to load descriptors from memory as needed; no descriptors need to be preloaded at dispatch time. Only those descriptors required for mediation are loaded by the hardware, and there is no requirement to save or restore descriptor memory in the mediation mechanism. This keeps the overhead associated with process switching and dispatching to a minimum. Measured results show that a process switch (from process A to process B) is accomplished in 1.85 milliseconds.

Another capability is the use of a descriptor cache to save the most recently used memory descriptors. This cache is a part of the SPM called the Virtual Memory Interface Unit (VMIU). The VMIU mediates memory requests and saves the descriptors in its cache. If another request is received for the same page, the VMIU can mediate it without performance penalty since no descriptor fetch is required. The overall system degradation caused by the mediation process is a function of the number of times the descriptor is in the cache when it is requested (hit ratio). If the hit ratio is 95% to 98%, then the performance ratios in this range do not appear to be difficult to maintain using good programming techniques.

OPERATING SYSTEM CHARACTERISTICS

The traditional approach to building a trusted operating system has been to build a security kernel and an operating system emulator to run on top of the kernel. This was the approach taken by UCLA and Mitre in their early development programs and by Ford for KSOS-11 [3]. One result of these efforts was the realization that this use of the operating system emulator produced results which were many times slower than the emulated system.

The original interface for SCOMP was to be a UNIX* emulator. This was the same type emulator used by KSOS-11, and the goal was to provide a compatible interface on both systems which could use the vast amount of software that existed on current UNIX implementations. However, the KSOS-11 implementation proved very inefficient, and an alternative user interface was developed for SCOMP.

Various alternatives were evaluated in an effort to develop an efficient interface for SCOMP. The final choice was to provide a low-level operating system interface using the SCOMP mechanisms to provide an efficient applications environment. The SCOMP operating system consists of three major functions. The security kernel enforces the security mechanisms and controls all access in the system. The Trusted Software provides the administrator, operator, and user services necessary to interface with the security kernel. The SCOMP Kernel Interface Package (SKIP) is the low level applications interface which provides a file system mechanism, event mechanism, and process control. These three elements of the SCOMP operating system combine to provide the user with an efficient interface for the development of application. Figure 2 shows how these software capabilities are structured, using the SCOMP ring mechanism to provide a layered operating system.

Security Kernel

The security kernel is the basic operating system that performs all resource management, process scheduling, memory management, trap and interrupt management and auditing. The security kernel also functions as the software portion of the reference monitor implementation. As such it controls access to objects in accordance with its imbedded security policy. The kernel supports three types of objects: segments, devices, and processes. Processes can also be subjects, using the reference monitor nomenclature. Each of the kernel's objects is identified by a 64-bit unique identifier. This unique identifier will never change for the life of the object in the system. The kernel also maintains data on each object in the system. This data is of two types, access information and status data. The access information consists of the security level and category set, and the integrity level and category set. The levels are hierarchical, and the category sets are 32 separate compartments for both security and integrity. Also contained in the access information is the discretionary information, which includes read, write, and execute permissions for the owner of the object, the groups of the owner, and all others. Additionally, the kernel maintains ring brackets for owner, group, and other which limit the ring of privilege required for access to an object. Subtypes are also provided to allow the user control of objects. The status information varies depending upon the object type.

The security kernel provides thirty eight (38) functions, called gates, which may be used by a process. These kernel gates allow for the creation and deletion of objects; the mapping (included in process address space) and unmapping of a segment or device; the wiring (keep in main memory) and unwiring of segments; the getting or setting of status; inter-process communication (IPC); and the reading and setting of the system clock.

The SCOMP security kernel is written in UCLA Pascal and consists of approximately 10,000 lines of code. It requires approximately 54K words of text and 9K of global data. Dynamic table space requirements depend upon the number of processes and is required for the management of the SCOMP demand paging virtual memory.

Trusted Interface

The interface to the SCOMP system for the user, system administrator, and operator is through trusted software. This software uses the security kernel for service and for special privileges to perform the trusted functions. This set of software is considered trusted for one of two reasons. First, it is software which requires the ability to violate one of the security or integrity properties enforced by the kernel (e.g., simple integrity or the security *-property). Second, trusted software utilizes functions which by their nature must be correct because the system's enforcement of security policy relies on their processing. An example of this class of trusted software is the data base editor, which builds the user access database. If it does not properly construct the database for the login process, then the login actions cannot be assured.

*UNIX is a trademark of Bell Laboratories.

The trusted software for SCOMP is being defined in the specification language GYPSY. This could enable eventual proof of the design. However, there are no current plans to verify the areas according to the type of trusted service provided. Trusted User Services provide the interface to the SCOMP system for the user; Trusted Operation Services provide the system operator with the capabilities necessary to run the system; and Trusted Maintenance Services provide the system administrator with the capability of building and maintaining the SCOMP system.

User Services

The user service functions provide the user with the ability to communicate through a trusted mechanism with the SCOMP system. These services allow the user to establish a processing environment in which applications can be run. The easiest way to understand these functions is to step through the process of logging a user on and establish his working environment. To initiate the login sequence, the user must depress the break key on the terminal connected to SCOMP. The break key has a special meaning for the kernel and is known as the "secure attention key." The kernel recognizes this condition and notifies the secure initiator process. The secure initiator controls terminals and creates a secure command processor, called the secure server, for the user's terminal. The secure server then prompts the user for the function to be performed. If this is the user's initial login, the login function will be invoked by the server. The login function will then validate the user through the user's identification and password. If the user were already logged in, he would be prompted for his next request. Services available to the user are login, change group, modify password, set access level, set default access, logout, file access modifier, reattach, process status, and kill.

These functions provide the user with the ability to establish an application environment, at a given security level, to allow an untrusted application to execute. The "reattach," "process-status," and "kill" commands allow the user to control the execution of various untrusted processes, which can be at any level for which the user is authorized.

Operation Services

This set of services consists of the functions necessary to start the system and to ensure the continuation of normal operation. These functions include secure startup, audit collection, secure loader, and operator commands.

Secure startup is the process that receives control at the end of the security kernel initialization. It is responsible for initializing all devices on the system and creating the audit collection process. The audit collection process receives audit records from both the kernel and trusted software, and builds accounting files. The secure loader is the mechanism that loads a secure process. It is known to the kernel and will be used for all requests to load trusted software.

The operator commands process will be called by the secure server after the operator has established a connection to the system. The operator commands include setting the system clock, system shutdown, switching accounting files, changing device attributes and device status.

Maintenance Services

This set of services provides the system administrator or operator with the ability to manipulate the system data. These services all fall into the class of trusted software which must execute correctly to ensure the validity of the system mechanism. These functions include the ability to initialize a kernel file system, to perform consistency checks on kernel file systems, to repair inconsistencies in the file system, and to dump and restore the file system content. A database editor is also provided to allow modification of system databases by the system administrator. The databases that can be maintained include the Access Authentication Database, Group Access Authentication Database, Terminal Configuration Database, Security Map, and Mountable File System Database.

Applications Interface

The poor results generated by attempts to build UNIX emulators on secure systems have resulted in Honeywell taking a new approach to building an interface for SCOMP. The basic requirements for this interface are:

1 - Provide a hierarchical multi-level file system
2 - Provide the ability to create child processes
3 - Contain an event mechanism for process synchronization
4 - Use the SCOMP hardware and kernel capabilities to provide an efficient interface
5 - Provide a low-level interface that could be used for multiple purposes or systems

In conjunction with a group of experts in the field of security and operating systems, Honeywell has designed and implemented the SCOMP Kernel Interface package (SKIP). SKIP is not an operating system but rather an interface to the secure environment which enables the users to effectively interface applications and systems with the security mechanism of SCOMP.

There has been some concern that this approach eliminates the existence of a UNIX emulator or UNIX environment on SCOMP. This is not true. The way to build a system which would look like UNIX, but utilizes SKIP, would be to write a command processor or shell and develop a set of interface subroutines which can map UNIX system calls to the proper combination of SKIP calls. This minimizes the modification to existing applications and provides the UNIX environment.

SKIP consists of two sets of routines. The major portion of SKIP resides in ring 2 and is activated through SKIP gate calls. The other portion of SKIP consists of a library of routines which execute in the user ring (ring 3). The following is a summary of the capabilities provided by this interface.

SKIP File System

The SKIP file system is a hierarchial structure comprised of directories, files, and links. It is an entry-naming system in that there is no interpretation of pathnames by the SKIP gates. The pathname interpretation is performed by a SKIP subroutine and may be modified if the user desires a different pathname interpretation from that provided. The file system is protected through the use of the SCOMP ring mechanism and subtypes. Only ring 2 software is allowed to modify the file system structure. The security level of the file system must be monotonically non-decreasing from the file system root. The directories, files, and links are identified by names of up to 24 printable characters. Directories are entries which contain information about other file system entries. A file is an entry whose contents may be directly modified by the users. A file is a collection of segments with a maximum size of 4,951 segments (18,804,736 bytes). All segments in a file must be at the same security and integrity level. The third file system entry is a link that points to a directory, file, or another link. The SKIP link mechanism is based on the Multics implementation of a link.

SKIP supports the capability of creating non-file-system segments. The user is able to perform functions similar to those provided by the kernel. To protect the integrity of the file system, there are only a limited number of kernel functions that can be called directly from the user ring.

SKIP Process Control

SKIP provides the user with the capability to create and delete processes, set priority, and send and receive event messages. In turn, these capabilities provide the basic requirement to build and manage processes in an application environment.

SKIP Event Processing

Events are messages indicating the occurrence of something meaningful to a process. The SKIP event mechanism is interrupt driven which allows for immediate processing when notification of an event occurs. The user has the flexibility of providing handlers for different events, as required. Events will be queued by SKIP such that the user will receive them in order of occurrence. The event mechanism is used to relay interrupt information to a process owning a device. This enables the user to process I/O using SKIP subroutines since the I/O service function is not supported by the security kernel.

SKIP Trap Processing

A trap is a software detected error that usually results in the abnormal termination of a process. The user may, however, desire to process the trap under certain conditions. SKIP provides for a user-written, trap-handling routine to process and take appropriate action when a trap occurs. The hardware trap save area is passed to the user by the system.

TOOLS AND APPLICATIONS

The SCOMP system will require the development of applications software and software tools. Many tools can be ported to the secure system from already existing operating systems. One of the first tools will be the Honeywell level 6 C–Language Compiler. This compiler runs on the Level 6 MOD 400 operating system and the Honeywell Level 6 UNIX. Other compilers and applications packages can be moved as they are required. During the course of application development, additional tools can be developed to meet specific needs in the multi-level security environment.

The initial test site for SCOMP is the NAVELEX–sponsored ACCAT GUARD. In this environment, the SCOMP will act as a GUARD between two networks with different security levels. The SCOMP will provide the interface for screening of data from the high-level network and the transmission of data to the low-level network. These networks will communicate with the SCOMP, using the DoD standard TCP/IP protocols.

The SCOMP capability can be used in many similar environments that have a requirement to pass information from systems or networks with different security levels. This general class of systems performs two basic functions. They filter requests for service to the high-level system by limiting the capabilities of the low-level user. They also provide the screening of information that flows from the high system to the low system. This type of system was demonstrated during the FORSCOM Security Monitor (FSM) project [4]. Trusted systems which support these applications provide the user with more efficient use of computer resources which require different security levels.

The SCOMP capability will support a variety of applications which require an enhanced architecture to provide for the processing of multiple levels of secure information. Potential applications include front-end processors, secure gateways, access control systems, secure database management, and multi-level mini-hosts.

SUMMARY

Honeywell has developed the SCOMP to meet the needs of the Department of Defense and industry. It provides an efficient system capable of processing information with multiple security attributes. There are many areas, both in government and private industry, where periods processing or the system-high mode of operation is used, which are costly and inefficient. The SCOMP capability can be the solution to providing the user with an efficient trusted computer system.

The development of trusted systems has had many ups and downs. The SCOMP program has tried to develop a system that meets the requirements for a trusted system without an unacceptable reduction in systems performance. The unique hardware and software security kernel is the key to meeting these requirements. The development of applications and tools will occur over the next several months as the use of the SCOMP is expanded.

A key element in the SCOMP program is the timely evaluation by the DoD Computer Security Center. The use of the SCOMP will expand when the center is able to provide potential users with a completed evaluation. In the center's first Product Evaluation Bulletin, the SCOMP was described as "a state-of-the-art base for a variety of security-sensitive applications."

REFERENCES

1. Bell, D. E., and L. J. LaPadula, "Secure Computer Systems: Mathematical Foundations and Model," M74–224, The MITRE Corporation, Bedford, Massachusetts, October 1974.
2. "Panel Session—Kernel Performance Issues," Proceedings of the 1981 Symposium on Security and Privacy, IEEE Catalog No. 81CH1629–5, April 27–29, Oakland, California, 162–178.
3. "Proceedings of the Second Seminar on the DoD Computer Security Initiative Program," National Bureau of Standards, Gaithersburg, Maryland, January 15–17, 1980, Sections R, T, and U.
4. "Proceedings of the Fourth Seminar on the DoD Computer Security Initiative Program, "National Bureau of Standards, Gaithersburg, Maryland, August 10–12, 1981, Section U.
5. Robinson, L., and K. N. Levitt, P. G. Neumann and A. R. Saxena, "A Formal Methodology for the Design of Operating System Software," in Current Trends in Programming Methodology, R. T. Yeh (ed.), Vol 1, Prentice-Hall, Engelwood Cliffs, NJ, April 1977. □

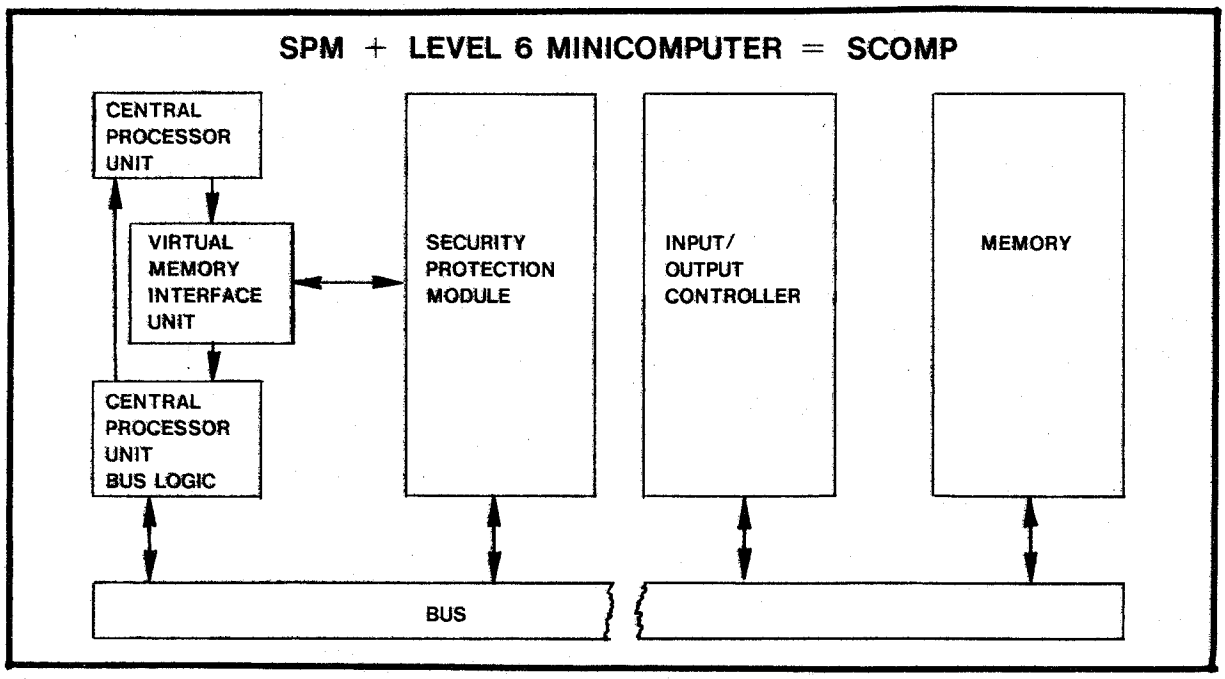


Figure 1. SCOMP Hardware Implementation

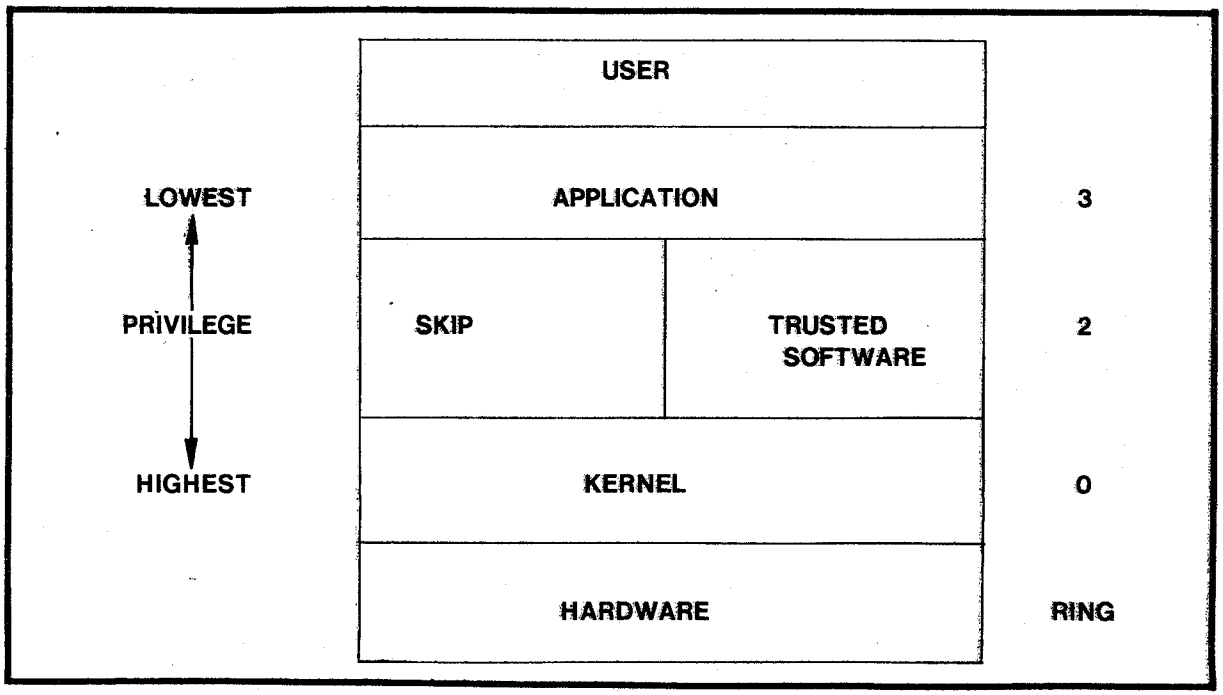


Figure 2. SCOMP Software Architecture

*UNIX is a trademark of Bell Laboratories.

AN UPDATE ON COMPUTER SECURITY ACTIVITIES AT DIGITAL

Andrew C. Goldstein
Consulting Software Engineer
Digital Equipment Corporation

Andy received a B.S. and an M.S. in Electrical Engineering and Computer Science from M.I.T., where he had his first exposure to computer security problems. He has been with DEC for over eight years, working in the file system area for RSX-11. Andy has also been part of the VAX/VMS program since its inception, and is currently involved with further development in the areas of the file system, security enhancement, and miscellaneous exec work in VMS.

Several other papers at this symposium have offered definitions of computer security. Software engineering at Digital has an implementor's viewpoint to offer: A verifiably secure system is one whose security and correctness have been established using mathematical models and proving techniques that are beyond the comprehension of the security evaluation team.

This paper presents an overview of the computer security-related activities that are going on at DEC. This is an update to a presentation that was given here in November 1980 by Paul Karger. There are two groups of significance doing work in the security area: Steve Lipner is in charge of security research and advanced development; I am in the VMS product development group and am involved with software that will be released to our user community in the near future.

Our overall goal is improved security for our computer systems. This means security for both commercial and government users. DEC sells to just about everyone. We sell to both the commercial market and to the government market; we sell to the telephone company; we sell to large corporations; and we sell to OEM's who in turn sell to corporations and small business. We sell to universities who, needless to say, have unique security problems of their own. We have a very broad market to address and the things that we are doing in terms of security development reflect that.

Digital is interested in evolvable security. We want to evolve existing software systems to become more secure. We are working on security-enhanced systems, and we are working on security kernels. We are also very concerned about networks, and have some security projects underway in that area involving encryption and authentication. And finally, we are also looking at layered product security. Layered products are the products that are available separately from the base operating system; this includes languages, data base packages, query facilities, and the like.

Layered products present a number of interesting security problems. We must make sure that they will continue to function in a security-enhanced system. It is always possible that some particular software product may take advantage of a lack of security in a particular system for its operations. Some of the layered products, in particular the data management products, need to have security controls of their own. We want to ensure that there is consistency between the security controls in layered products and the security controls that are provided by the base operating system.

In terms of evolvable security, it is very important to us that security features fit in with our existing products. DEC has been in business quite a while. We have been selling PDP-11's for well over ten years, and we have been selling VAX's for several years. Not only do we have a lot of investment in that software and a public commitment to its stability, but our customers have an enormous investment in their own

---

software that uses our operating systems. We must be very careful not to invalidate customers' software with future security enhanced products; doing so would severely discourage the market for such products.

Let us look at the specific work in this area of evolving security: security-enhanced systems and security kernel research. We have built a security-enhanced VMS prototype. It supports the lattice model for mandatory security controls, including confinement. It includes access control lists of the sort found in Multics and several other operating systems. It includes security event auditing and a number of other features that improve its overall security and integrity above the base version of VMS upon which it was built.

A very important aspect of this project was that there was no major restructuring of the executive. In this respect, it was similar to the Multics AIM security enhancements. In other words, the additional security controls were added to the security controls that already existed in the base VMS system. We feel that a system of this sort, once completed, would be rated at B1 or B2 in the proposed evaluation criteria. This research prototype was completed about a year and a half ago, and runs on a VAX-11/780. We feel that the effort was quite successful. It ran with a minimal performance penalty. The performance of this system was essentially the same as the performance of the original VMS system that it was built from. We must stress that this was a prototype; this is not a product currently available from Digital. Among other things, the implementation is incomplete. For example, the mandatory security controls were not applied to all of the storage objects in the system; completing their implementation is essential for any kind of viable commercial system. There were a number of user features needed to make the system run smoothly that were either not done or done as kluges of insufficient generality.

It is our intention to evolve these basic features into the product over time. The prototype has proven that the basic concepts work. Therefore, the next step is to start phasing the features into the VMS product. This work depends on a number of factors. Most important, it depends on compatibility with existing VMS systems, since we have a very strong release-to-release compatibility commitment to our customers. It also depends on priorities from our users. We have a large and diverse user base who have their own priorities and needs, which include not just security but many other system features. Obviously, our resources are limited.

We are pleased to see the security guidelines come out at this time because the development of the security enhancements in the VMS product is just getting underway. Therefore, the guidelines will provide useful input in planning the details and priorities of this project.

In TOPS-20, we have a similar security enhancement project underway. There has not been a breadboard done yet. This project is in a preliminary design phase. Its features and objectives are similar to the VMS prototype. They include the addition of mandatory controls, enhanced discretionary controls, and security event auditing, to improve the ease of use and the net resulting security of TOPS-20. We intend to evolve these features into a product over time.

We also have research underway into a security kernel for the VAX-11. This is also in the preliminary design stage. Once again, we plan to produce a system compatible with VMS. This is an evolutionary product, although in a somewhat different sense. Although the system internals will be quite different, the interface presented to users will be the same as VMS. Our general approach is to build a small kernel containing just the minimal functions necessary to support the security features. We will be using a layered design, patterned after the Multics kernel approach (work done by Jerry Saltzer at MIT).

Being a manufacturer, we are very concerned about the performance and cost effectiveness of the systems that we build. This system will be performance-tuned for VMS. In other words, its behavior will be tuned so that the higher levels of VMS will run well on it.

We intend to formally specify and verify the design of the security kernel to the Bell and LaPadula security model. We are pleased to hear the declaration of support by the Computer Security Center for the specification and validation tools; we expect to find such support useful.

Our target for the security kernel is to achieve an A1 rating according to the proposed criteria if the design verification is successful. As noted previously, this is and will be a research prototype; it is not a

committed product. Making it a product will depend on the overall success of the prototype, the level of cost performance that we realize, and the market demand that we see once it is built.

We are also very concerned about network security problems. DEC has a computer network that ties together most of the timesharing computers that are used by the different engineering groups. This network has upwards of 300 systems on it. It spans the continental United States with a heavy concentration in New England, and sites in Colorado Springs, Seattle, and Europe. The engineering net is used by a great diversity of users of DEC, including engineering groups, writers, marketeers, and customers coming in for demonstrations—all kinds of people who don't particularly trust each other (not to mention the occasional intruding high school student).

While I cannot claim that our security problems are at the same level as those of the DoD, we do have a substantial security awareness of our own. The very existence of our own network provides us with a lot of incentive to be looking at network security.

There are two areas in network security that we are exploring: encryption and authentication. Aspects of DEC's network design, both present and future, affect how we would implement encryption. For example, our current network implementations are point-to-point networks with automatic routing by intermediate nodes. These routing nodes are likely to be untrustworthy. DEC is developing ETHERNET support for release in the near future. The ETHERNET is a party line to end all party lines, so it presents security problems of its own. And, finally, we are looking at interfaces to packet switched networks which are also not trustworthy. All in all, it is clear to us that end-to-end encryption is the only viable technique for secure networks.

We have encryption research currently underway. It is specifically directed to serving DECNET; we intend to use the DES algorithm, implemented with the 14 megabit AMD chip. We will be using the DES algorithm for both packet encryption and key distribution. We have a research and advanced development prototype currently underway.

We are very interested in how the DoD would like to interface with our DECNET encryption facilities. We assume that DoD will not use DES, yet we feel some level of responsibility to make our network encryption features work for the DoD. What is needed in terms of session level protocol? How do we interface to different cryptographic devices? Would an interface to the TCP 4 protocol suffice? Since there is already a third party TCP 4 IP available for the VAX, integrating it into VMS would be a very convenient way to allow VMS to be hooked up to DoD networking facilities. These questions require further discussion.

The other aspect of network security has to do with authentication. We are moving into a more and more distributed environment. The typical ETHERNET is network with workstations and servers of all kinds where you may, for example, get a request from a user which is passed on from A to B to C and so on, until the request is finally satisfied. This poses a number of interesting questions in how you establish that the request really came from who you think it came from. We envision an authentication facility layered on top of the encryption facility so that the encryption facility forms a trusted channel through which authentication can be passed. Our objective is to pass user names from machine to machine without transmitting passwords. Sending passwords over the net poses extremely serious security problems, since they are prone to interception. Such a design also encourages the storage of passwords in files where they risk compromise.

Because of the complex network features that we anticipate, we must be able to deal with cascaded network connections.

To implement network authentication forwarding, we have developed a concept called proxy login. This is a facility that allows an individual or a server to grant proxy rights to specific individuals on other nodes in the network. In this context, individuals could mean real users (persons at terminals), or it could mean server processes.

The classic example is a network print spooler. When the print spooler requests access to a particular file, its identity and the identity of the user on whose behalf it is operating cause it to be granted a proxy login in the form of a particular local user name; this local user name is granted the necessary access to the file in the file's access control list.

Proxies may also be used to limit resource use as well as to restrict access to data. By providing sufficient detail in the proxy authorization record, the resources and capabilities available to a proxy may be limited, so that, for example, granting a proxy for the purpose of transferring a file doesn't give the proxy holder carte blanche to run batch jobs under that account.

The access control list semantics that we are developing for the security-enhanced VMS include features to support the proxies. They include features that will allow us to tie access control list entries to particular proxy entries coming in over the net.

In addition to the proxy facility, we are also looking at a feature called an authentication server. This is a concept that was developed some years ago at the University of Cambridge. An authentication server has been in operation for a while now on the Cambridge Ring. The authentication server is needed to validate proxy requests that go through cascaded network connections. Some kind of trusted technique is needed to forward the identity of a requestor through a cascade of servers, each one of whom is partially serving and then passing on the request. Encryption alone does not solve this problem because without such a server it is always possible for an intermediate node to either fabricate or play back an authentication identifier.

The original requestor registers with the authentication server and receives a one-time identifier. This identifier is passed with the network request; all of the servers along the path can have the identifier validated by the authentication server to verify the identity of the original requestor.

We have already implemented, as a prototype, a subset of the proxy login facility. It is present in latent form in the VMS V3.0. (The term "latent" means that the feature is present, but is not documented or supported.) We did this partly to gain some experience with the proxy login concept. We wanted to live with it for a while and find out for ourselves how it works. Also, we implemented it to serve the purposes of the engineering net, because we have some of our most severe security problems right there.

In summary, we hope that what I have discussed in this paper convinces you that Digital is active in computer security. We are working on it; we do care. We are aiming at both government and commercial requirements. We intend to produce a system that serves both markets and we intend to evolve security features and improve security in DEC products. □

GOAL

IMPROVED SECURITY OF OUR SYSTEMS
FOR
COMMERCIAL AND GOVERNMENT USERS

ISSUES

- EVOLVABLE SECURITY
- SECURITY-ENHANCED SYSTEMS
- SECURITY KERNELS
- NETWORK SECURITY
- ENCRYPTION
- AUTHENTICATION
- LAYERED PRODUCT SECURITY

EVOLVABLE SECURITY

- SECURITY MUST FIT IN WITH EXISTING PRODUCTS
- SECURITY-ENHANCED SYSTEMS
- SECURITY KERNEL RESEARCH

SECURITY-ENHANCED VMS PROTOTYPE

- FEATURES
  - LATTICE MODEL/MANDATORY CONTROLS
  - ACCESS CONTROL LISTS
  - SECURITY EVENT AUDITING
  - IMPROVED INTEGRITY
- NO MAJOR RESTRUCTURING OF EXECUTIVE
- B1/B2 ON RATING SCALE
- COMPLETED RESEARCH PROTOTYPE ON VAX-11/780
  - MINIMAL PERFORMANCE PENALTY
  - PROTOTYPE ONLY—NOT A PRODUCT
- FEATURES TO EVOLVE INTO PRODUCT OVER TIME DEPENDING ON
  - COMPATIBILITY OF INTERNALS AND INTERFACES
  - PRIORITIES FROM USERS
  - RESOURCES

TOPS-20 SECURITY ENHANCEMENTS

- PRELIMINARY DESIGN STUDY UNDERWAY
- FEATURES AND OBJECTIVES LIKE VMS PROTOTYPE
  - MANDATORY CONTROLS
  - ENHANCED DISCRETIONARY CONTROLS
  - SECURITY EVENT AUDITING
  - IMPROVED INTEGRITY
- FEATURES TO EVOLVE INTO PRODUCT OVER TIME

VAX-11 SECURITY KERNEL RESEARCH

- PRELIMINARY DESIGN UNDERWAY
- COMPATIBLE WITH VMS
- APPROACH
  - KERNEL WILL BE SMALL
  - MINIMUM FUNCTIONS TO SUPPORT SECURITY
  - LAYERED DESIGN
  - PERFORMANCE TUNED FOR VMS
- DESIGN WILL BE FORMALLY SPECIFIED AND VERIFIED TO BELL & LAPADULA MODEL
- LEVEL A1 ON RATING SCALE
- RESEARCH PROTOTYPE—NOT COMMITTED PRODUCT

NETWORK SECURITY

- ENCRYPTION
- AUTHENTICATION

ENCRYPTION ISSUES

- END-TO-END ENCRYPTION FOCUS
  - ETHERNETS HAVE NO LINKS
  - MIXED PACKET SWITCHED NETWORKS
- ENCRYPTION RESEARCH
  - DECNET FOCUS
  - DES ALGORITHM (14MB AMD CHIP)
  - RESEARCH/ADV. DEV. PROTOTYPE
    - NO PRODUCT COMMITMENT
- HOW ABOUT DOD?
  - SESSION LEVEL PROTOCOLS?
  - CRYPTOGRAPHIC DEVICES?
  - DOES TCP 4 DO IT?

AUTHENTICATION FORWARDING

- LAYERED ABOVE ENCRYPTION
- PASSES USER NAMES FROM MACHINE TO MACHINE WITHOUT PASSWORDS
- MUST ADDRESS CASCADED NETWORK CONNECTIONS

AUTHENTICATION FORWARDING RESEARCH

- PROXY LOGIN
  - ALLOW INDIVIDUAL TO GRANT PROXY RIGHTS TO SPECIFIC OTHER INDIVIDUALS ON OTHER NODES
  - CAN LIMIT RESOURCE USE AS WELL AS SECURITY
- DEFINE ACCESS CONTROL LIST SYNTAX/SEMANTICS TO TIE PROXIES TO SPECIFIC FILE ACCESS RIGHTS
- AUTHENTICATION SERVER
  - VALIDATES PROXY RESULTS FROM CASCADED CONNECTIONS
  - ENCRYPTION ALONE IS INSUFFICIENT
  - DEVELOPED AT UNIVERSITY OF CAMBRIDGE
- PROXY LOGIN SUBSET PROTOTYPED ON VMS VERSION 3.0

CONCLUSION

- DIGITAL IS ACTIVE IN COMPUTER SECURITY
- AIMED AT GOVERNMENT AND COMMERCIAL REQUIREMENTS
- SECURITY WILL EVOLVE IN DIGITAL PRODUCTS