C CE & TECHNOLOGY:

THE NETWORK SECURITY CENTER: A SYSTEM LEVEL APPROACH TO COMPUTER NETWORK SECURITY

DEPARTMENT OF COMMERCE ★ NATIONAL BUREAU OF STANDARDS ★

NBS Special Publication 500-21, Volume 2
U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards

NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards[1] was established by an act of Congress March 3, 1901. The Bureau's overall goal is to strengthen and advance the Nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research and provides: (1) a basis for the Nation's physical measurement system, (2) scientific and technological services for industry and government, (3) a technical basis for equity in trade, and (4) technical services to promote public safety. The Bureau consists of the Institute for Basic Standards, the Institute for Materials Research, the Institute for Applied Technology, the Institute for Computer Sciences and Technology, the Office for Information Programs, and the Office of Experimental Technology Incentives Program.

THE INSTITUTE FOR BASIC STANDARDS provides the central basis within the United States of a complete and consistent system of physical measurement; coordinates that system with measurement systems of other nations; and furnishes essential services leading to accurate and uniform physical measurements throughout the Nation's scientific community, industry, and commerce. The Institute consists of the Office of Measurement Services, and the following center and divisions:

Applied Mathematics — Electricity — Mechanics — Heat — Optical Physics — Center for Radiation Research — Laboratory Astrophysics[2] — Cryogenics[2] — Electromagnetics[2] — Time and Frequency[2].

THE INSTITUTE FOR MATERIALS RESEARCH conducts materials research leading to improved methods of measurement, standards, and data on the properties of well-characterized materials needed by industry, commerce, educational institutions, and Government; provides advisory and research services to other Government agencies; and develops, produces, and distributes standard reference materials. The Institute consists of the Office of Standard Reference Materials, the Office of Air and Water Measurement, and the following divisions:

Analytical Chemistry — Polymers — Metallurgy — Inorganic Materials — Reactor Radiation — Physical Chemistry.

THE INSTITUTE FOR APPLIED TECHNOLOGY provides technical services developing and promoting the use of available technology; cooperates with public and private organizations in developing technological standards, codes, and test methods; and provides technical advice services, and information to Government agencies and the public. The Institute consists of the following divisions and centers:

Standards Application and Analysis — Electronic Technology — Center for Consumer Product Technology: Product Systems Analysis; Product Engineering — Center for Building Technology: Structures, Materials, and Safety; Building Environment; Technical Evaluation and Application — Center for Fire Research: Fire Science; Fire Safety Engineering.

THE INSTITUTE FOR COMPUTER SCIENCES AND TECHNOLOGY conducts research and provides technical services designed to aid Government agencies in improving cost effectiveness in the conduct of their programs through the selection, acquisition, and effective utilization of automatic data processing equipment; and serves as the principal focus wthin the executive branch for the development of Federal standards for automatic data processing equipment, techniques, and computer languages. The Institute consist of the following divisions:

Computer Services — Systems and Software — Computer Systems Engineering — Information Technology.

THE OFFICE OF EXPERIMENTAL TECHNOLOGY INCENTIVES PROGRAM seeks to affect public policy and process to facilitate technological change in the private sector by examining and experimenting with Government policies and practices in order to identify and remove Government-related barriers and to correct inherent market imperfections that impede the innovation process.

THE OFFICE FOR INFORMATION PROGRAMS promotes optimum dissemination and accessibility of scientific information generated within NBS; promotes the development of the National Standard Reference Data System and a system of information analysis centers dealing with the broader aspects of the National Measurement System; provides appropriate services to ensure that the NBS staff has optimum accessibility to the scientific information of the world. The Office consists of the following organizational units:

Office of Standard Reference Data — Office of Information Activities — Office of Technical Publications — Library — Office of International Standards — Office of International Relations.

[1] Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington, D.C. 20234.

[2] Located at Boulder, Colorado 80302.

COMPUTER SCIENCE & TECHNOLOGY:

The Network Security Center: A System Level Approach to Computer Network Security

+Special publication, no. 500-21

Frank Heinrich

System Development Corporation
2500 Colorado Avenue
Santa Monica, California 90406

Sponsored by the
Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, D.C. 20234

U.S. DEPARTMENT OF COMMERCE, Juanita M. Kreps, Secretary

Dr. Sidney Harman, Under Secretary

Jordan J. Baruch, Assistant Secretary for Science and Technology

u.S

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Acting Director

Issued January 1978

Reports on Computer Science and Technology

The National Bureau of Standards has a special responsibility within the Federal Government for computer science and technology activities. The programs of the NBS Institute for Computer Sciences and Technology are designed to provide ADP standards, guidelines, and technical advisory services to improve the effectiveness of computer utilization in the Federal sector, and to perform appropriate research and development efforts as foundation for such activities and programs. This publication series will report these NBS efforts to the Federal computer community as well as to interested specialists in the academic and private sectors. Those wishing to receive notices of publications in this series should complete and return the form at the end of this publication.

National Bureau of Standards Special Publication 500-21, Volume 2
Nat. Bur. Stand. (U.S.) Spec. Publ. 500-21, Vol. 2, 69 pages (Jan. 1978)
CODEN: XNBSAV

Library of Congress Cataloging in Publication Data
Heinrich, Frank
The network security center.
(Computer science & technology) (NBS special publication ; 500-21 Vol. 2)
Supt. of Docs. no.: C13.10:500-21, Vol. 2
1. Computers–Access control. 2. Electronic data processing departments–Security measures. I. Institute for Computer Sciences and Technology. II. Title. III. Series. IV. Series: United States. National Bureau of Standards. Special publication ; 500-21, Vol. 2.
QC100.U57 no. 500-21, Vol. 2 [QA76.9.A25] 602'.1s [001.6'4]
77-608304

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1978

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402
(Order by SD Catalog No. C13.10:500–21, Vol. 2), Stock No. 003–003–01881–3 Price $6 per 2 volume set; sold in sets only.
(Add 25 percent additional for other than U.S. mailing).

PREFACE

This Special Publication reports on extensions to work performed by the System Development Corporation which was reported in an earlier document, "Design Alternatives for Computer Network Security," NBS Special Publication 500-21, Volume I. Originally prepared for publication as an NBS Technical Note, this report is published in the new NBS Special Publication 500 Series in order to obtain wider distribution in an important topic area.

This document is one of the products of the Computer Security Project within the Institute for Computer Sciences and Technology at the National Bureau of Standards. The project has been structured to develop a set of technical safeguards which may be used to protect the computer systems of the Federal Government and the data that they process and store. A method of protecting data being communicated within a network of computers has been identified as a requirement within this set of technical safeguards.

This report culminates a project sponsored by NBS and carried out by the System Development Corporation. The information contained herein should be used as a tutorial on an approach for achieving network security. The approach has not been endorsed by any Federal Agency as the one to be used in all cases in providing network security.

The assistance of K. Auerbach, M. Berger, G. Cole, D. Kaufman, and R. von Buelow of the System Development Corporation in the development of concepts and the production of this report are hereby gratefully acknowledged.

TABLE OF CONTENTS

LIST OF FIGURES

EXECUTIVE SUMMARY

1.0 OVERVIEW

The current investigation has built upon an earlier study which produced the tutorial document, "Design Alternatives for Computer Network Security." Extensions of the effort were to address the specific aspects and usage of the NBS Data Encryption Algorithm in the computer network environment, and more specifically, for use with a Network Security Center. Guidance was required in these areas since network user groups, such as government agencies, will typically want to tailor their own Network Security Center(s) to reflect their particular usage and security policy requirements. Therefore, the principal objectives of the effort were to produce guidance in the areas of network security architectural issues and implementation options. A wide variety of networking configurations were to be considered, including existing and newly developed networks. In effect, our objective was to discuss what might reasonably be done by an agency which has need for computer networking, but which has privacy, fraud protection, or other security constraints to meet. Since this represents a very broad community, our discussions were to cover a wide range of access control and surveillance needs, while still providing a sufficient level of detail to guide an actual implementation. The resulting report is quite large in size as well as scope, and hence this Executive Summary has been created to discuss the highlights of these findings.

2.0 BACKGROUND FOR THE INVESTIGATION

Both the government and commercial enterprises have found an increasing need to interconnect computer systems into networks to share data, and other resources, in a timely and efficient manner. However, the difficulty in ensuring network security has impeded those network developments which deal with personal information and financial transactions for both privacy and fraud prevention reasons. A major step in the resolution of these problems has been the development of the NBS Data Encryption Algorithm, since it forms the basis of a solution to the networking portion of the overall security problem. Implementations of the Algorithm will be widely available and will typically allow the use of remote, electronic key distribution. This allows it to be utilized in a manner in which separate encryption keys are used for each different dialogue or connection. Such keying provides a basis for the authentication of the two communicating devices, for protection against message misrouting, and for possible access control enforcement, as well as the usual message protection against unauthorized reading or changes to the text. These separate keys can be generated at, and disbursed from, a central site, which will be called the Network Security Center or NSC. The NSC can also provide the network-wide authentication and authorization checks which are needed for network access control and can collect audit data regarding proper use of the net, as well as attempts at illegal access.

The Network Security Center is the subject of this Special Publication and of the companion document, "Design Alternatives for Computer Network Security." The latter document discusses the various issues and solution alternatives for network security using the NSC approach. This Special Publication expands upon these results to present more specific guidance in how an NSC should be designed and implemented for use in a network with encryption devices using the NBS Data Encryption Algorithm. In summarizing these results, we will describe the various aspects of the network security problem, define the overall solution approach, discuss the objectives in writing this Special Publication and outline the principal results of the project.

3.0 DEFINING THE NETWORK SECURITY PROBLEMS

The network security problems of concern to this investigation were those that are the direct result of connecting computer systems into a network configuration. Many of these problems exist in individual computer systems, but are addressed by the procedural, personnel or physical controls as used by most computer centers. These methods are ineffective in managing external users in a network environment, so new controls must be added. The problems of maintaining the access control data base for the network also become unmanageable due to the large number of persons, terminals and computers, as well as the dynamic nature of personnel assignments and the resulting need to change privileges, passwords, etc. Similarly, the collection of audit and surveillance data is frequently of little value since it tends to be fragmented across the network resources and hence has no global context for interpretation. Other vulnerabilities also arise, such as the possibility of misrouted data or disclosure in a switching center, particularly with the new Value Added Networks (such as implemented by packet or message switching).

It should be emphasized that the network security problems addressed in this Special Publication are above and beyond any individual host computer center security problems. The additional problems are due to the interconnection of a host with other systems which are outside the individual security perimeters as established by the procedural and other controls of the various sites. Since there is little that the network can do to strengthen the security of the individual host computer systems, the solutions which will be discussed relate primarily to the network security problems.

Solutions are most readily applied as an integral part of the design of new networks, and guidance is presented for such developments. However, in the near term, there are many existing systems and networks which require additional security controls, and the Network Security Center approach is shown to be viable for a wide variety of these existing network architectures. The success of the approach is largely due to the fact that it provides an additional layer of security controls, rather than attempting to modify the existing computer systems. The network access control and auditing functions provided by the NSC are completely external to the individual systems, as will be described in the following section.

4.0 THE OVERALL NETWORK SECURITY APPROACH

The network security solution approach which is advocated here is based on the usage of one or more Network Security Centers (NSCs) as global access control and auditing mechanisms, as well as a "security interface" between each site and the network. This interface includes a remotely keyable cryptographic device which, due to its capabilities of key management and interface protocols, is called a Network Cryptographic Device (NCD). All such encryption is end-to-end across the network, i.e., between the NCDs at the two communicating sites. In addition to the usual message protection on the communication links, end-to-end encryption has the following advantages over link-by-link encryption:

(1) Information is protected in intermediate switches as well as on the communication links. This also minimizes the verification concerns for the switches.

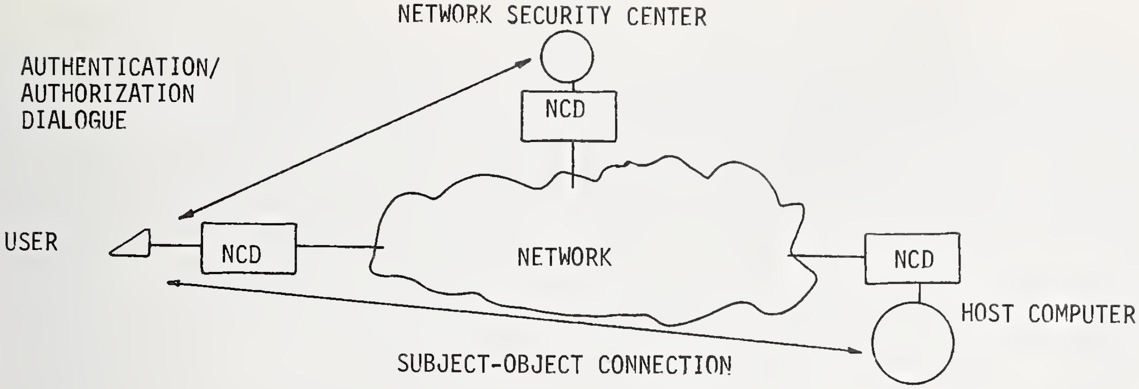
(2) Any misdelivered messages are unintelligible to the recipient.

(3) The ongoing (properly deciphered) communication gives implicit and continual authentication of the two communicating devices.

(4) The encryption keying mechanism can be utilized to initially authenticate the NCD (and its data processing device) to the NSC.

(5) Access control enforcement is possible by means of the keying of the two NCDs (by the NSC). Only NCDs with matching keys can communicate.

The operation of the NCDs and the NSC can best be described by an example. Suppose that a user is operating a terminal which is connected to the network via an NCD. The user wishes to communicate with a host computer which is also connected to the network via an NCD. However, the user must first communicate with the Network Security Center (which also contains an NCD).


NETWORK SECURITY CENTER
AUTHENTICATION/ AUTHORIZATION DIALOGUE
NCD
USER
NCD
NETWORK
NCD
HOST COMPUTER
SUBJECT-OBJECT CONNECTION

All initial cryptographic keying of the NCDs is such that any requestor must initially connect to the NSC, which performs authentication and authorization checking before enabling the desired connection by "priming" the two NCDs that are to communicate. This "priming" may be by means of matching encryption keys distributed to the NCDs or by appropriate permission messages to the NCDs, which then establish their own keys. If encryption keys are distributed from the NSC, they are protected during transmission by encryption using a special, pre-defined key. Audit data are collected by the NSC for all such requests and appropriate actions are taken when illegal access attempts are made.

An access request may originate from either a human user or from a computer system. In the former case, the NSC carries on an interactive dialogue with the user, requesting an identifier, an authenticator, etc. as needed. For computer requests, these items are rigidly formatted and are sent as a single request after the encrypted connection to the NSC is established.

One of the principal benefits of the NSC/NCD solution approach is that it provides logically separate subnets which share the same physical communication net. That is, different user communities can be kept isolated, and hence secure from one another, since the NSC controls which NCDs can communicate with other NCDs. Since the NCDs interface the terminals and host computers to the network, these data processing devices are also segmented into logically separate nets. In practice, even the NSCs could be separate, one per logical subnet, with NSC-to-NSC cooperation if any communication is desired between these otherwise separate groups. Separate NSCs might be desired for geographic or administrative reasons, and are discussed in the Special Publication as "Regional NSCs." The paper also discusses the situation in which two or more networks are interconnected and how the NSCs would operate in this environment.

The basic level of access control which the NSC/NCD approach can effectively enforce is that of creating a logical connection between the two communicating devices. A finer granularity of access control, such as to the file level, can not be enforced by the NCDs, but the NSC could be utilized as a secure repository of finer granularity access permission data. The NSC would act in an advisory role to the finer granularity enforcement mechanisms within a host. Guidance is presented in this area.

5.0 PRINCIPAL RESULTS

The principal results of the investigation are the definition of the requirements and design considerations/recommendations for an NSC/NCD-oriented solution to the network security problems. However, particular emphasis has been on the NSC functions and guidance related to its implementation. Major results presented in the Special Publication are described in the following sections.

5.1 Definition of Generic Functions

The basic communication functions of a network were shown to be: (1) the user interface, (2) connection management, (3) access control, (4) cryptographic protection, (5) internetting (if required), and (6) the network interface. Since the NSC is a network component, it must implement these functions. The functions are discussed as a set of inter-related levels in the sense that an implementation would consist of a layering of these functions. Several possible orderings exist, with differing characteristics. For example, the placement of the cryptographic protection level may depend on the physical architecture of the network. In an internetting situation, a logical connection may be implemented over several distinct cryptographic links or conversely, a single end-to-end cryptographic link might be implemented over several logical connections. Previous networking experience has shown that the most important aspect of this levels concept is that communications are symmetric relative to the various levels, i.e., the source and destination of messages should both be at the same functional level. This has not always been the case in practice, and the lack of a well-defined and implemented "levels concept" has resulted in difficulties in the introduction of new features such as internetting and end-to-end encryption.

Each of the functional levels is discussed in detail, and guidance is presented for determining their relative ordering. The audit and surveillance function is distributed across several levels, and is also discussed.

5.2 Fitting the NSC/NCD into Various Network Architectures

The NSC operates at the access control level (and possibly at the encryption management level) of a network, while the NCDs may include the connection management, encryption management, and network interface levels. The physical arrangement and implementation of the NSC will vary depending on the particular network architecture and the extent to which the network is already in existence. In each case, the NSC should serve as a policy interpreter and enforcer with the policy being tailored to the individual security requirements of the network.

Three different network architectures were considered in the investigation: (1) a single host computer with its attached terminals, (2) a single host computer with a communications front end for terminal handling, and (3) a multi-host intercomputer network. Each of these is summarized in the following paragraphs.

An NSC can be added to the single host with attached terminals in either of two ways. First, it could be implemented entirely in software within the host computer system. This provides little direct benefit over conventional host access control mechanisms except that it mav allow one to evolve to an external NSC. That is, the NSC functions would be logically separated from the rest of the system and hence could be physically removed at a later date. A second and more useful solution would be to impose the NSC between the terminals and the host computer, much like a communications front end. However, it would also provide access control and encryption keying (if desired). As an extension to the above approach, the NSC might be integrated into an existing communication front end or could be added as an external device to the front end.

In a multi-host network, the NSC would appear as an additional host which controls all network accesses (via the NCDs). The extent of access control would typically be in terms of device-to-device connections, but the NSC may act in an advisory role for finer granularity access control exercised by the hosts themselves.

5.3 Audit and Surveillance

Since the NSC is the central controller for network accesses, it provides a valuable source of audit information relative to usage of network sites. Therefore, all access requests should be recorded, both those granted and those denied. Thresholds should also be set to indicate an abnormal number of improper requests to detect illegal attempts at network access. Guidance for these surveillance mechanisms is presented, but individual implementations will primarily reflect local policy concerns.

5.4 Guidance for the Usage of the NBS Data Encryption Algorithm in Networks

The security threats which should be addressed include message exposure on the communication lines and in intermediate switches, the possibility of misrouting messages, the unauthorized modification of message content, and the introduction of spurious messages (e.g., by recording and playing back legitimate messages). Guidance is provided in terms of how the NBS Data Encryption Algorithm can be utilized to circumvent these threats through direct message encipherment, key management, cipher check codes, and operations using enciphered data. The benefits of end-to-end versus link-by-link encryption are also described (as discussed earlier in this summary).

The NBS algorithm is shown to be an excellent choice for use in network security applications because:

(1) The secrecy of the transformation is dependent only on the secrecy of the key, not on the secrecy of the algorithm.

(2) The length of the key is 64 bits, eight of which are reserved for parity. Thus there are 2^{56} potential keys. The key is not so short as to make exhaustive search techniques feasible, yet not so long as to make distribution to a remote device difficult.

(3) The algorithm is block-oriented; that is, data is grouped into blocks of 64 bits which may be enciphered and deciphered independently of any other block. As long as the same key remains in use, position or time synchronization of encryption with decryption is not required.

Due to routing and transmission differences, message transit time through a network is somewhat variable. Messages may arrive at a destination in a different order than they were sent. Using the NBS Algorithm, cryptographic devices can be built which do not require position or time synchronization and are independent of the communication subsystem.

(4) When enciphering or deciphering, the change of a single bit in either the key or the input text has an unpredictable effect on the output text. This characteristic has two implications. First, the correct key must be known to make use of (i.e., decipher) enciphered information. Second, alterations to enciphered text cannot produce predictable changes to the corresponding clear text.

(5) Analysis of clear/enciphered text pairs does not aid in code-breaking to determine the key used. Penetrators are forced to use impractical exhaustive search techniques for code-breaking.

(6) The NBS algorithm is expected to be available as an LSI package. This will provide a low cost, high speed implementation suitable for use in network cryptographic devices.

In addition to the direct usage of the NBS Algorithm for communications security, several other applications are of value in the context of an NSC-oriented net. Data files could be enciphered at the various hosts with the encryption keys being stored at the NSC or in a separate special purpose host (as a secure repository). The NBS Algorithm is also useful as a generator of the key values for use with the enciphered files and with the NCDs.

5.5 Definition of the Access Control Data Structure for the NSC

Access control consists of two parts: (1) authentication that the requestor is valid, and (2) authorization checking to ensure that the requested resource is indeed accessible in the requested manner. These two portions of access control, and the corresponding data structures, are described in an implementation independent manner and then an implementation example is presented.

This example considers a network of 1000 users and 10 host computers and is based on the assumption that each user would access the net once every 20 minutes, and would have an average of 25 objects which he could legitimately access. The resulting data base was 780,000 bytes. Performance estimates indicated that a moderate size minicomputer with a moving head disc could handle this level of traffic using the recommended form of data storage and retrieval.

The NSC-oriented solution was shown to be appropriate for a wide range of network sizes, with the I/O service being the most critical limiting factor in the upward growth for a single NSC. However, the design guidance is oriented towards providing multiple NSCs if required, so there is adequate growth potential for very large networks. At the other extreme, the limiting factor is economics, i.e., when the cost of the simplest NSC is high relative to the rest of the network.

5.6 Recommended I/O Structure

Since I/O response time is a critical factor in the NSC operation, the I/O structure was investigated in detail to provide implementation guidance in this area. The following sequence of operations is recommended to minimize the I/O problems:

The first action to be performed is to locate the user-profile block. This action isolates the portion of the total data base relevant to a single request. The logical I/O software will be aware of the user-profile structure. An I/O request from a process will specify the owner ID for the desired profile-block. The I/O software will locate and read into the separate process address space the appropriate profile block.

In the course of processing a request, the process will require additional I/O service to read subsequent portions of the user-profile. Due to space restrictions, we assumed a request process would only contain buffers for one disc sector at a time. Thus, the requests for subsequent portions of a profile block will be in terms of disc sector addresses, which are contained as pointers in the entries in the profile block.

Thus the logical I/O structure will accept initial I/O requests in terms of owner ID and subsequent I/O requests in terms of disc addresses. The unit of I/O transfer is one sector. Each sector is read into the separate address space of the requesting process.

Updates to the data base are handled by a request to a single update process. This process can issue appropriate locks on portions of the data before updating.

5.7 Recommended Control Structure

The approach to controlling processes is based on three major concerns: (1) the ability for the NSC to interface to both human and computerized requests for network connections, (2) a modular structure made up of simple components, and (3) the ability to separate processes into different physical machine environments. The latter feature allows the modular expandability to multiple NSCs when necessary for growth and also helps to ensure proper interprocess operations since all inter-process communications are explicit.

The process structure is defined in terms of a set of request processes (one per active requestor), another set of support processes (such as required for secondary storage I/O, communication management, cryptographic key generation, data base updating and auditing), and a set of nucleus functions (including memory management, process scheduling, interrupt handling, and I/O drivers).

The process structure presented has a great deal of error control inherent in the design. The use of separate processes for each user request, separate address spaces, separate code and data segments, hardware address protection enforcement, and a centralized update process all contribute to error control. Inadvertent errors are isolated and restricted in scope.

5.8 Definition of Message Formats

The various message formats involved in the process-oriented NSC are defined in terms of generic message contents and the relationship between the various messages, e.g., the message sequences and acknowledgements. These formats include those for the user-to-NSC dialogue, the computerized requestor-to-NSC messages, the I/O request messages, the cryptographic key-related messages, the inter-NSC messages, and the audit messages. The particular bit patterns are not defined; instead the necessary fields and message sequences are specified.

6.0 CONCLUSION

The operation of the Network Security Center is described and recommendations are developed for various implementation options. The NSC approach appears to provide a unique solution to the network security problems since it can effectively control network access, provide audit data collection, and provide protection against tampering or modification of the access control data base. Since multiple NSCs can operate together, issues such as modular expandability, regional subnets and local control over resources can also be addressed by this solution approach. With the introduction of the NBS Data Encryption Algorithm and remotely keyed cryptographic devices using this algorithm, the NSC can provide a viable solution to the problems caused by interconnecting computers into network configurations.

THE NETWORK SECURITY CENTER: A SYSTEM LEVEL APPROACH TO COMPUTER NETWORK SECURITY

Frank Heinrich

This report describes a unique approach to the solution of computer network security problems, and provides guidance in the areas of network security architectural issues and implementation options. The approach is based on a network resource, called a Network Security Center (NSC), which performs the functions of user identification/authentication and access request authorization. The NSC works in concert with Network Cryptographic Devices (NCDs) to enforce access control policy through the creation or denial of logically separate cryptographic connections between subjects (users) and objects (resources). The use of a NSC in a network permits effective control over network access, provides for audit data collection, and provides protection against tampering or modification of the access control data base. The architecture presented permits multiple NSCs to operate together, thus addressing issues such as modular expandability, regional subnets, and local control over resources.

Network Cryptographic Devices that use the NBS Data Encryption Standard algorithm and are capable of being remotely keyed are a vital part of the NSC security approach. NCDs provide end-to-end cryptographic message protection, source-destination authentication of identity and, through the remote keying capability, the enforcement mechanism for NSC access control decisions.

Implementation options for an NSC are presented, covering the areas of data structures, I/O structure, control structure, and size and performance limitations.

Keywords: Access authorization; access control; authentication; computer network security; cryptography; end-to-end encryption; inter-computer network; internetting; NBS Data Encryption Standard; Network Cryptographic Devices; Network Security Center.

1.0 FUNCTIONAL REQUIREMENTS AND SYSTEM DESCRIPTION

1.1 Introduction

Sharing of first generation computers was accomplished primarily by dividing the usage of the computer into dedicated time allocations, with carefully controlled job set-ups between jobs. Multi-processing and multi-programming provided a more efficient usage of the computer hardware by virtue of rapid context switching between jobs and by overlapping operations. Under these systems several jobs could be executed concurrently, adding to the security problem. As software and data resources grew in size, multi-resource systems such as inter-computer networks were created. These networks may involve a large number of system users who may be dispersed geographically, with many different security privileges and requirements. Procedural and administrative controls may no longer be sufficient to augment the security provided by the computer system. Present concerns in these new environments are:

- insuring that subjects (users) are identified and authenticated;
- insuring that no subject is allowed access to an object (resource) for which he is not authorized;
- insuring that all references are consistent with the access request;
- keeping audit records of which subjects access which objects;
- protecting transmitted data from modification or disclosure by advertent or inadvertent means (e.g., wire tapping or misdelivery).

Cryptographic mechanisms directly address this last concern, but can also be a tool used in implementing mechanisms to address the other problems.

This report describes an approach to meeting the security needs of networks. It is based on the concepts described in "Design Alternatives for Computer Network Security," [1] subsequently referred to as "the tutorial."

The tutorial discusses a broad range of issues which are of concern to inter-computer network security. The general conclusions are that use of a Network Security Center (NSC) and Intelligent Cryptographic Devices (ICD) is a viable approach to providing network security, and that the NBS Data Encryption Standard is ideal for such usage.

The current report narrows the range of problems and provides more detailed specific recommendations for design and implementation of an NSC. The approach described here evolves from the tutorial, in that the functions of cryptographic protection have been more clearly separated into a distinct implementation level.

Access control at the NSC level and the cryptographic level still have a complementary relationship, each supporting the function of the other.

Although this report is more specific and detailed than the tutorial, it is still not intended as a complete design. Included are guidelines and recommendations, often in the form of examples, which apply the considerations developed in the tutorial to more specific situations. The detailed requirements of a particular implementation may be interpreted in relation to these guidelines.

1.2 Scope and Purpose of the Network Security Center (NSC)

1.2.1 General Characteristics. The NSC shall serve as a mechanism which provides the functions of identification and authentication of subjects, authorization of access requests, and audit and surveillance in a network which connects terminals and host computers. The access control mechanism provided by the NSC may either replace or supplement the normal host computer access control mechanism.

Path oriented protection methods shall be used by the NSC to exercise its control over access to objects (resources) by subjects (users or programs). That is, the NSC shall control the logical access paths between subject and object and thus may regulate access by permitting or denying the creation of such logical connections. The NSC shall operate in conjunction with other network components, specifically the communication system interfaces, in order to implement and enforce this control. The cryptographic mechanisms shall maintain separation of distinct logical connections by using different encryption keys for each connection. End-to-end encryption also provides an implicit authentication that the other party in the communication path is the one intended. Use of the NSC approach, as developed in the tutorial and as amplified here, implies use of cryptographic devices at each site in the network. Although this will remain the major emphasis, Section 1.4.2 will discuss other possible roles for the NSC.

The tutorial specifies three basic functions to be provided by the network security mechanisms (including, but not limited to, the NSC).

1. Providing controlled access to resources. This includes identification/authentication and access request authorization.

2. Providing controlled usage. This involves ongoing checks to insure that references are consistent with the authorization.

3. Providing that assurance protection is maintained. This includes audit and surveillance to provide records of permitted and denied access, as well as accreditation of the mechanisms themselves.

The nature of the subjects and objects may vary with the particular application. For example, a subject might be a person, a terminal, a host computer, a process, or some combination of these. An object might be a host computer system, a data file, a data item, a program, or another user.

1.2.2 Relationship to Reference Monitor. The reference monitor approach to security mechanisms [1] indicates that such mechanisms must always be invoked, isolated from unauthorized alteration, and accredited as trustworthy. The NSC system separates the reference monitor function into two components. The NSC itself operates as a policy interpreter which performs the functions of identification/authentication of subjects, and authorization of access requests. Site interfaces serve as the reference enforcement mechanism.

To insure that the mechanism is trustworthy and isolated from unauthorized alteration, both the NSC and the reference enforcement mechanism shall be implemented in components separate from the host computers. If secure host computer systems are available, it may be feasible to consider placing some of the reference monitor functionality within such a host. The NSC shall be involved only in the initial decision of whether or not access shall be granted (Security Function 1 above). If the access is permitted, the NSC shall allow creation of a logical connection between the subject and the object. The interface mechanisms at either end of the communication path that connects the subjects and objects must provide ongoing enforcement to ensure that each subsequent reference conforms to the original request for authorization (Security Function 2).

Although the general policy that is interpreted by the NSC is not restricted by the approach outlined here, it makes sense to implement a policy which can be enforced by the reference enforcement mechanism. The policy must therefore be consistent with capabilities of both components, the NSC and the interface mechanisms.

An example of an inconsistent policy would be one in which the NSC controlled access to objects of the granularity of a data file, but the site interfaces cannot control or even ascertain which file is being referenced. Section 3 of this report discusses a general access control schema, and a specific implementation example.

In order for the security mechanisms to be effective and the policy to be consistent, the following conditions are necessary:

- The site interfaces must prevent all access to resources, until permitted by the NSC to allow access;
- The site interfaces must allow the subject to communicate with the NSC itself, to permit identification/authentication and authorization dialogues;
- The site interfaces must be able to accept an 'instruction' from the NSC that will cause a logical connection to be created;
- The subject must be an entity identifiable (and authenticated) by the NSC;
- The subject must be an entity identifiable to the subject site interface;
- The object must be identifiable (at some level of granularity) by the NSC;
- The object must be identifiable and access to it must be controllable by the object site interface.

1.2.3 Policy Variations. As previously mentioned, although the NSC approach does not restrict the type of policy that may be implemented, the nature of a specific situation may limit the desirable alternatives. The architecture of the network, the logical placement of the NSC, and the capabilities of the network components will all have an effect on the type of policy that can be implemented.

Since the NSC makes a one-time decision to grant access and is then no longer involved, the policy that the NSC implements must be consistent with the capabilities of the site interfaces that actually perform the reference enforcement function. The subjects and objects must be identifiable and controllable by the site interfaces if the site interfaces are to effectively enforce the policy implemented by the NSC. Thus the granularity of access control is directly tied to the capability of the reference enforcement mechanism.

More complex access requests may involve a qualifier as well as an object. A qualifier restricts the range of operation permitted in subsequent references or defines a subset of the object data that will be referenced. Such qualified access requests may be enforced only if the enforcement mechanisms have access to the parameters of the qualifier in order to verify and control the access. For example, if the access request (and the policy) include an operation type as a parameter, then the reference enforcement mechanism must be at an appropriate level in the network to interpret each reference as to operation type. This would be feasible at the site interface only if the site interfaces handled predictable, known format messages, rather than unformatted character streams (or binary input).

Section 1.4 discusses various network architectures, the role of the NSC in networks of the various types, and how the logical placement of network components affects the type of policy that can be enforced.

1.2.4 Cryptographic Mechanisms. Development of the NBS standard encryption algorithm has made it feasible to consider using encryption techniques as a part of the reference enforcement mechanism. The tutorial recommended a close functional relationship between the NSC and the encryption mechanism. The advantages of that approach can still be obtained when the two implementation levels are more completely separated. Section 2 of this report describes in more detail these cryptographic approaches.

1.3 The Levels Concept

To more clearly describe the function of the NSC and how it interacts with other components of a network, the decomposition of a network is described in terms of distinct implementation levels. This decomposition is somewhat idealized, in that many existing networks do not evidence a clean separation into levels. However, it will serve as a good framework for analysis, as well as a guideline for future network development.

The levels concept is a way of examining the various communications functions existing in a computer network. This concept was originally developed to describe intercomputer networks, and in particular the logical message flow within the network. For example, a user at one network site wishes to send a message of some sort to another user at a different site and he is not particularly concerned how that message gets there, merely that it does arrive unaltered and in a timely manner.

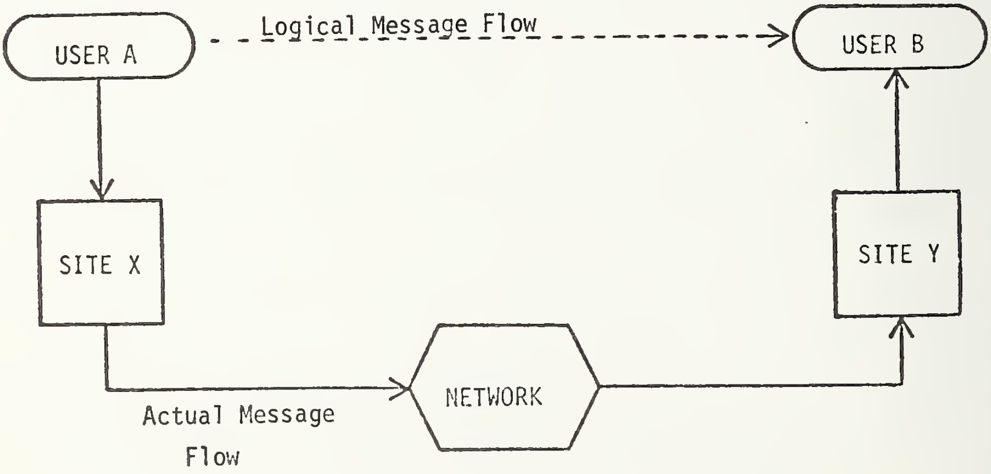


Figure 1-1. Logical Vs. Actual Message Flow

Therefore we can think of a user level where messages are transmitted back and forth. In actuality, however, the messages may go "down" to another level, or "down" through several levels, before the actual physical message transmission takes place. At the user level messages are composed and then handed to the next level to be handled and routed as appropriate. Each level would see the level(s) below it as the message carrier. The interface between any two levels would be relatively simple. Compared to the message text, the amount of header information is relatively small. Even though the amount of information going through the carrier level is hundreds of characters long, only a small portion of the message (i.e., the header) is actually inter-level communication.

In an actual computer system, or network of computer systems, there are a number of generic areas of functionality. In some systems all of these functions may be provided in one form or another within a single program, while in another system the functions may be distributed in different hardware devices. In many cases in current systems the functions are not clearly defined, a situation that may hinder system development, modification, and/or expansion. Separation of functional levels limits the span of control of each networking functions, localizes the effects of modifications, and facilitates testing. Therefore it is advantageous to define and separate these levels.

Among the levels defined for a (inter-computer) network are:

1. User/Process Level;

2. Access Control Level;

3. Connection Level;

4. Message Protection Level;

5. Network Interface Level.

It should be noted that the purpose for understanding these levels is that when considering a specific system it may be desirable to move the levels around in relation to each other, to move the functions at a given level from software to hardware (or vice versa), to combine two or more levels in one program or module, or to otherwise take advantage of this functional modularity.

1.3.1 The User/Process Level. This level deals with the interface with the ultimate correspondents in a dialogue. This level is required to collect character (or binary) data and assemble it into messages properly formatted for communication through the lower levels. The NSC will expect certain standard information organized into predefined formats. The user level will be required to collect the appropriate information and place it into the required formats, performing any required conversion to standard forms.

In the case of an inter-computer network, the user level at each site is required to perform these functions. In the case of a user, functions at this level would include such things as interpretation of strokes at a keyboard, light pen input, card reader input, et cetera. For a process this level might deal with such things as buffering of character strings into messages and buffering of messages. Dialogues at this level may be between user and user, user and process, or process and process. For a user at a terminal, a terminal handler function would deal with the individual characteristics of the various terminals in order to make the differences in these terminals transparent to "lower" levels of processing.

1.3.2 Access Control Level. This level is usually composed of two parts, an identification/authentication part and an access authorization check. The identification/authentication function must be able to recognize and authenticate processes and users. The access authorization function must determine, based on some set of access control information, the allowability of an access request.

The Access Control Level performs the following functions: 1) conducts an identification/authentication dialogue with each subject to determine the subject's identity; 2) conducts an authorization dialogue with the subject to determine its access request;

3) determines whether the request is permitted; and 4) if access is authorized, permits connection of the subject to the requested object.

The first function to be performed is to conduct an identification/authentication dialogue with the subject. There are several alternative methods for authentication of identity [1,2]. Basically, these methods consist of the use of secret passwords and the verification of personal characteristics. In a general sense, these schemes are functionally equivalent. Each relies on forming an association between an identifier, which is assumed to be public knowledge, with a supposedly non-forgeable authentication value. This authentication value is made somewhat non-forgeable in the password scheme by keeping the password secret. The password may be entered by the user or read from some storage medium, such as a magnetic card. In the personal characteristics scheme, the authentication value is derived from a characteristic that cannot be forged, such as a fingerprint, voiceprint, or the like. In either case, the access control mechanism (the NSC) collects both parameters (the identifier and the authenticator), then checks to see if the pair corresponds with a previously stored association.

The difference between the two schemes is that a personal characteristic authenticator generally requires a large data value to describe the characteristic in order to be effective, while a password is generally a more limited size. Current personal characteristics authentication systems have a limited capacity in terms of number of users, and also have unacceptable rates for passing imposters and rejecting valid users. Thus for the present, secret passwords are the alternative recommended for the NSC.

The general form of the identification/authentication dialogue shall be for the user level to pass to the NSC the value pair <Identifier, Authenticator>. The NSC shall check for a proper match, and return a response, either <Valid Identity> or <Invalid Identity>. The user level shall return an appropriate response to the subject (determined by what type of subject is involved) that reflects the response from the NSC.

Since knowledge of the pair <Identifier, Authenticator> is sufficient to forge an identity to the NSC, any network component that deals with the authenticator in the clear (i.e., above a message protection level) must take special care that it is not disclosed. The communication system is protected by encryption devices, so this requirement falls mainly on the user interface, the site interfaces and the terminal itself.

To guard against penetration attempts which use trial and error to attempt to find the proper authenticator through a process of elimination, the NSC shall limit the number of successive tries permitted. That is, only a limited number of 'mistakes' will be allowed from one subject. Also, the NSC shall provide the <Valid Identity> or <Invalid Identity> response only after a fixed delay, thus preventing a penetrator from using the speed of response to determine if a portion of an authenticator (the first "n" characters) is correct. All externally observable characteristics of the NSC shall be identical whether the authenticator is found to be invalid on the first, intermediate, or last characters.

The second function of the Access Control Level is to conduct an authorization dialogue with the subject. The authorization dialogue is concerned with a specific access request. The information that the NSC shall obtain is as follows:

- Authenticated subject identity (as established in identification/authentication dialogue).

- Subject profile, which includes any characteristics of the subject that are required by the security policy. This may include information collected from the user level, such as current subject security level, as well as stored information about the subject's permitted actions.

- Object identity. This will be part of the access request from the subject.

- Object profile. As with the subject profile, this may include current state information, as well as stored characteristics.

Access Request Qualifier. This will be part of the access request from the subject and will include the type of access requested and any additional qualifiers to the access request as may be appropriate.

The subject user level shall supply an access request composed of the subject identity, the object identity and the access request qualifier. In addition, the subject user level will provide any portion of the subject profile that is within its domain (e.g., current security level). The object user level will provide any portion of the object profile that is within its domain. The NSC itself will maintain the remainder of the subject and object profiles.

The third function of the Access Control Level is to determine if the requested access is to be permitted. If it is not, the NSC will return an <Access Denied> response to the subject's user level, which will pass an appropriate indication to the subject. If the access is to be permitted, the fourth function is establishing a connection, which is discussed in Section 1.3.3 immediately following.

1.3.3 Connection Level. These functions shall establish, maintain, and terminate logical connections that are permitted by the access control level. Once established, a connection provides the logical "pipeline" through which information can pass.

Based on the information collected at the user level and access control level, the NSC will determine whether the requested access is to be permitted. If it is permitted, the NSC shall return an <Access Permitted> response to the subject user level, which in turn will pass an appropriate indication to the subject. A similar indication may be passed to the object user level.

The NSC shall instruct the site interfaces at both subject and object to establish a logical, separately keyed connection. Depending on the type of protection level that is used, the NSC may generate and distribute an encryption key for use on that connection. In any case, the NSC will establish a unique connection identifier so that the respective site interfaces may coordinate their actions. If the access request involved a qualifier that must be monitored by the site interfaces, the NSC shall pass the appropriate parameters to the site interface.

The previous discussion centered on connections established between a single subject and a single object. Other possibilities are conceivable, such as single subject to multiple objects, multiple subjects to single object, or multiple subjects to multiple objects (conference call).

There are several ways in which such complex connections might be implemented. The underlying communication subsystem may offer primitives that permit establishment of such connections. In this case, the connection level would simply be a reflection of the primitives offered at lower levels. The message protection level would have to be aware of the nature of the connection so that the multiple sources and destinations may be appropriately keyed to permit the connection. The NSC would be aware of the type of connection as specified in the access request, and would determine if such a connection was authorized according to the security policy.

If the communication system does not offer primitives that permit such complex connections, the connection level may still implement such a connection for higher levels. In this case, the connection level would translate the multiple connection request into a set of normal one-to-one connections. The protection level would establish encryption keying based on the single connections. The NSC would be aware of the multiple nature of the connection, since it deals with the connection request as the user sees it.

If the connection level does not provide the complex connections, users may still establish such complex connections by their own coordinated actions. Each user would establish the appropriate one-to-one connections such that the resulting interconnection is equivalent to that resulting from a multiple connection. The users would be responsible for the details of coordinating message assembly and duplication, message addressing, multiple message transmission, flow control and error response. The connection and protection levels would deal only with the various one-to-one connections. The NSC would also deal only with one-to-one connections. However, since the NSC is involved with every access request from a subject, and since the NSC keeps a record of all authorized connections, the policy that the NSC implements could take account of potential multiple interconnections. The NSC could determine if the pattern of separate one-to-one connections violates the policy. Regardless of whether the multiple connection was intentional or not, the NSC would make a decision on each separate connection, denying only those that would cause an illegal interconnection.

1.3.4 Message Protection Level. There are two basic ways of providing message protection. One way is to provide physical protection for all of the hardware and communications lines through which messages travel. Another way is to utilize cryptographic methods to provide protection. It is normally not possible, or is prohibitively expensive, to physically protect and shield the entire communications channel. Therefore cryptographic techniques are often a better choice when protection is required.

The message protection level will utilize cryptographic methods to provide protection of information from disclosure or modification and implicitly authenticate both ends of a communication path. The NBS standard encryption algorithm is well suited for this purpose.

The message protection level shall provide end-to-end encryption that 'spans' lower levels of the network. Thus the devices that implement the protection level must distinguish between message control information needed by lower levels and text information that must be protected. The cryptographic level is symmetric about lower levels of functionality. Lower levels need not be certified or trusted, except that they provide functional service. All software and hardware components that lie above the cryptographic level must be trusted, since data is not protected by encipherment. The cryptographic level itself must be certified that it is trustworthy. The level of certification may range from testing to formal models and rigorous proofs.

The message protection level shall maintain a distinct encryption key for each logical connection. The protection level shall accept from the NSC an instruction to create a separately keyed logical connection between a subject and object. The key may be supplied to both ends of the communication path by the NSC, or the protection level may establish its own keys. Logical connection identification is provided by a unique identifier supplied by the NSC when it instructs the protection level to create the connection. Any messages that are misrouted or delivered with an incorrect identifier cannot be decrypted properly. The protection level may incorporate redundancy check information within the encrypted data so it may determine if a message has been decrypted properly.

Once a connection is established, the protection level shall be transparent with respect to protocols at other levels. If the levels are kept strictly separated, there will be no crossing of information between levels, except as specified in the restricted protocol that defines the inter-level interfaces. Thus, lower levels will only need to interpret control information, which the protection level has left in the clear. The text portion of a message, the content of which is of no importance to lower levels, will be encrypted.

By leaving control information in the clear, message communication is subject to traffic flow analysis. Protection against this can be provided by line level encryption on each physical line.

1.3.5 Network Interface Level. This level is made up of the functions and facilities that deal with the actual transmission of messages from one site to another. It can be composed of packet switches, telephone lines, physical wires, a commercial telecommunications network, or the like. Functions at this level also provide supplemental support with data flow control and error detection facilities.

The Network Interface functions provide the actual connection to the communication network. The complexity of the Network Interface functions will vary with the type of network. For example, the interface to a packet switched network is much more complex than the interface to a dedicated private line. The details of the Network Interface Function are independent of the NSC access control functions. By separating these functions into a distinct level, access control (implemented through the NSC connection control functions) and message protection can remain independent of the type of communication facilities used.

In order to insure that the access control mechanism is always invoked, each site shall insure that the primitives of the network interface level are only available through the mediation of the NSC. Only a properly keyed logical connection established by the NSC shall have access to the primitives of the network interface level.

1.3.6 Levels Concept for Single Host Networks. The "levels" concept can be applied to a network of terminals attached to a single host computer. The same areas of functionality as in the previous situation can be identified. Users are interacting with user level functions. There is a logical connection function, object level entities, access control, message protection and a communication subsystem. The ordering and inter-relationships of these functional modules is different in the single host situation. Figure 1-2 shows a logical diagram of a terminal network.

Although ultimately communication occurs between a user and an object, to the communication subsystem the terminal user and the user-level process are the correspondents. The communication subsystem in this situation is much less complex than for inter-computer networks. In most cases, the communication system will simply be a line connecting a terminal with a host or front-end communication processor.

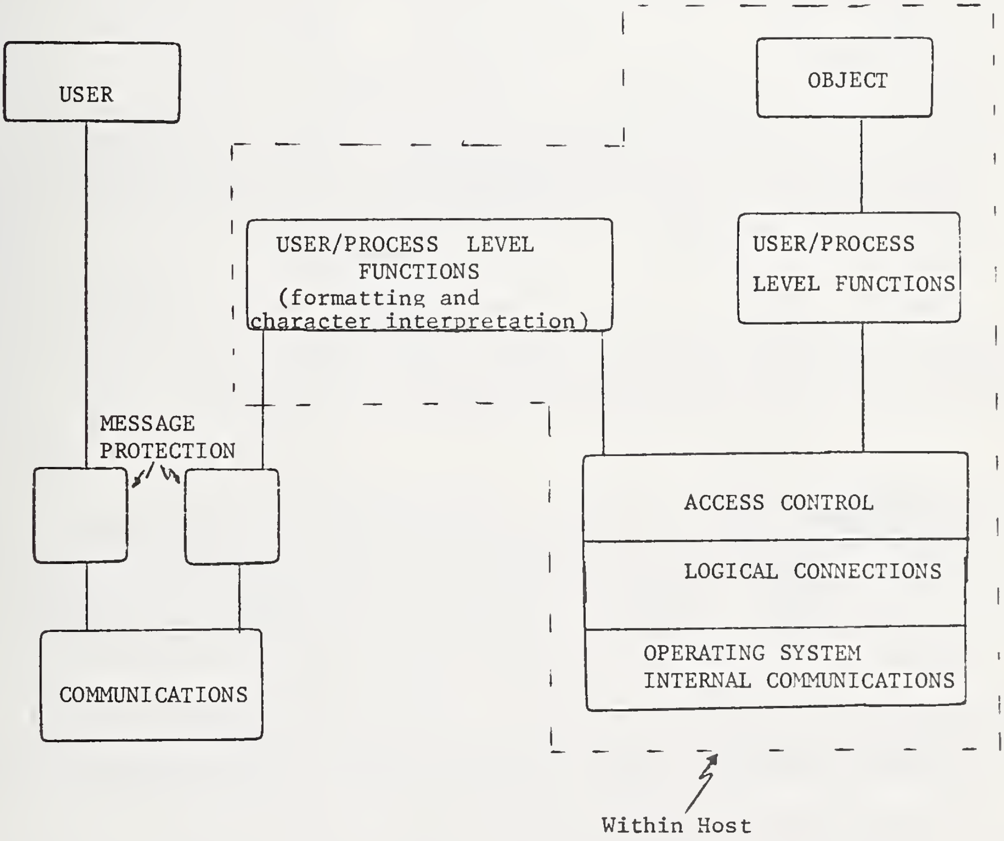

USER
OBJECT
USER/PROCESS LEVEL FUNCTIONS (formatting and character interpretation)
USER/PROCESS LEVEL FUNCTIONS
MESSAGE PROTECTION
ACCESS CONTROL
LOGICAL CONNECTIONS
OPERATING SYSTEM INTERNAL COMMUNICATIONS
COMMUNICATIONS
Within Host

Figure 1-2. Logical Diagram - Terminal Connection to Host

A message protection level is placed symmetrically about a communication subsystem, depicted in Figure 1-2. The implementation of the message protection level may be quite simple in this case, since the communication system does not require extensive information nor complex protocols for its operation. Control information is often implicit, thus obviating the need for separate header and text data in the messages.

The functions of the user level, access control, logical connection, and inter-process communication may be subsumed within the host. If the NSC is to effectively perform access control, it must be separated from the host. Consequently, either the user level functions must also be provided outside the host, or the NSC must duplicate those functions for itself.

The NSC will enforce access control by interaction with the message protection level. To prevent bypassing the NSC controls, the NSC must be placed so it will interpose between the user and the host system.

Figure 1-3 shows a terminal system with a grouping of functions into components.

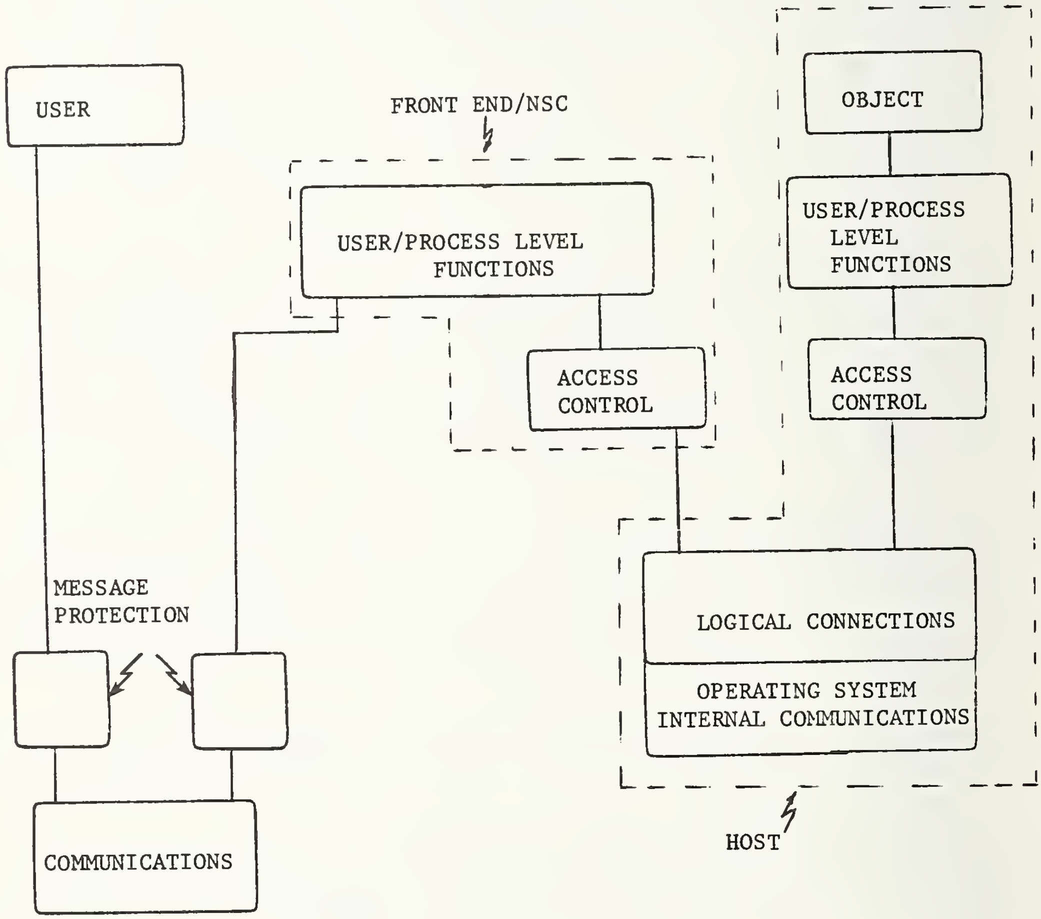


Figure 1-3. Terminal System - Functional Grouping into Components

1.4 Role of the NSC in Various Network Architectures

1.4.1 NSC as a Policy Interpreter and Enforcer. The NSC corresponds to an implementation of the Access Control Level, previously discussed, while the site interfaces correspond to the connection level and the protection level. In the tutorial, the connection and protection level were combined into a single component which was called an Intelligent Cryptographic Device (ICD).

The general scenario of NSC operation is as follows. The NSC conducts a dialogue with the subject to determine and authenticate its identity. The NSC conducts a dialogue with the object to determine if the requested access is authorized according to the policy. If everything is proper, then the NSC acts to establish a connection between the subject and the object. The exact nature of this action varies according to the type of network.

The NSC might interact directly with the protection level, or indirectly through a connection level. In the ICD approach, as presented in the tutorial, the NSC establishes the cryptographic connection by passing the encryption key to devices at both ends of the protection level. Another approach separates the function of key distribution from the NSC. In this case, the NSC has a command input to the protection level that causes the protection level to establish a separately keyed connection.

At the conclusion of use of an object by a subject, the enforcement mechanism must insure that the connection is broken. The NSC must be notified so that audit and surveillance functions can be performed.

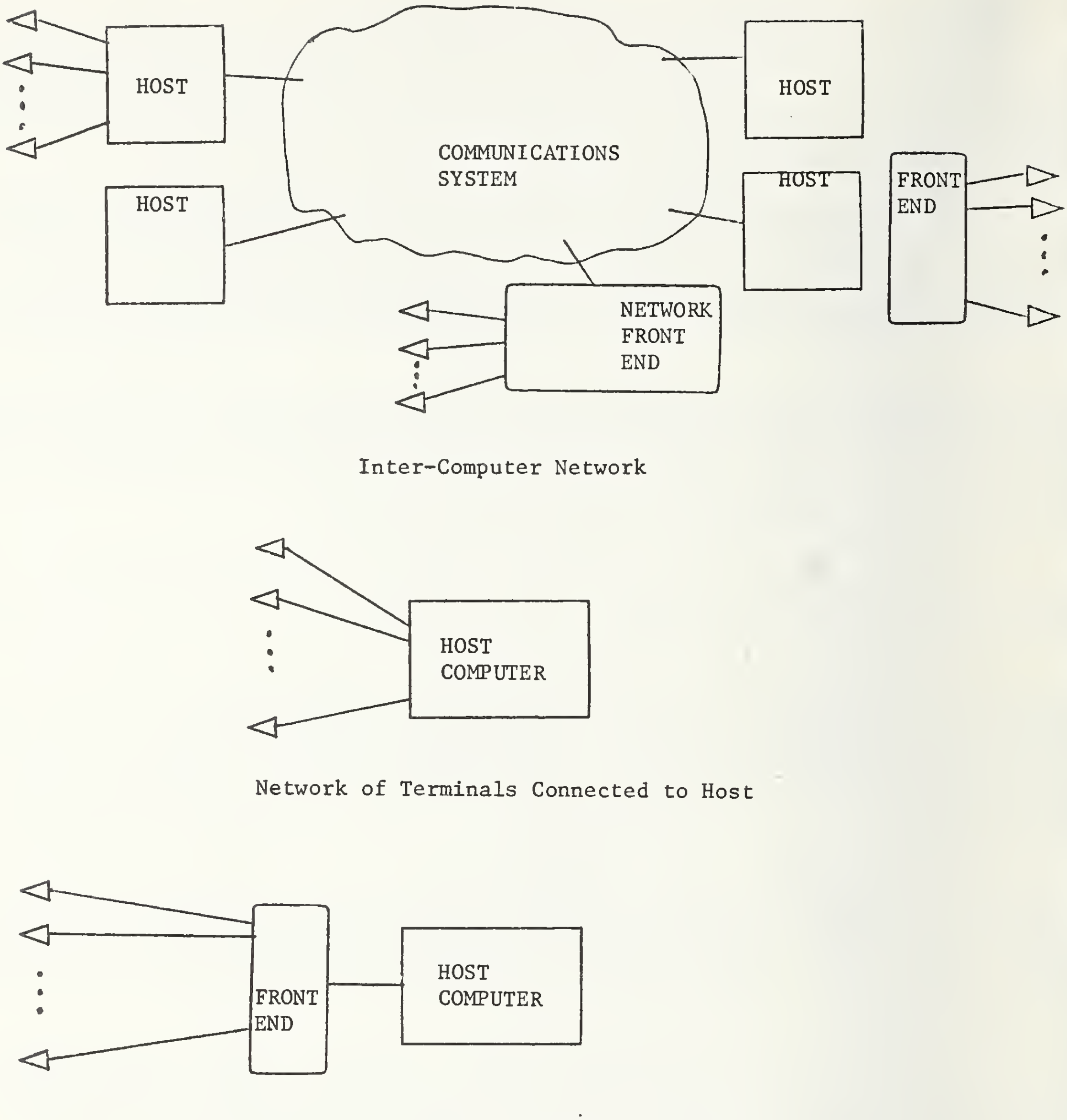
One of the requirements for security mechanisms is that they be isolated from unauthorized alteration. To meet this requirement, the state-of-the-art requires that these mechanisms be implemented in components separate from the host computer system. If a mechanism were implemented as part of the host computer, that mechanism would be as vulnerable as any resource on that host. Thus, this report focuses on implementations where the NSC and reference enforcement mechanisms reside in components separate from the host computers.

In subsequent sections, this report concentrates on three network architectures. (See Figure 1-4.) First, an inter-computer network that connects host computers having attached terminals, host computers without terminals, and front ends which allow direct terminal access to the network. Second, a network of terminals connected to ports on a host computer. Third, a network of terminals connected to a front end processor which is connected to a host computer.

1.4.2 The NSC in an Inter-Computer Network. The first architecture interconnects three types of components: host computers without attached terminals, host computers with attached terminals, and network terminal handlers that support direct terminal access into the network. Each of the entities being interconnected has 'intelligence' and supports user level functions. To the communication system, the difference between processes and users at terminals is transparent. The communication system provides the functions of message routing, message delivery and logical connection levels.

The NSC appears as an additional host in this type of network. Initially, all subjects (users or processes) are connected to the NSC. They conduct appropriate identification/ authentication and authorization dialogues after which the NSC will cause a logical connection to be created between the subject and the appropriate object.

The logical connection level must allow a subject to connect only to the NSC until another connection is authorized by the NSC. The protection level will maintain cryptographic separation of logical connections throughout the communication subsystem, but each site is responsible for maintaining separation within its own local environment (above the protection level).



Inter-Computer Network

Network of Terminals Connected to Host

Network of Terminals Connected to Host Through Front End

Figure 1-4. Three Network Architectures

1.4.3 The NSC in a Terminal Network Connected to a Host. In this architecture, the host computer handles the function of terminal support and command interpretation. Thus, if separate components are to provide access control and reference enforcement, they must operate on uninterpreted character streams. The NSC must provide terminal handling facilities in order to carry out the identification/authentication and authorization dialogues. Since the protection level operates on uninterpreted characters, the only enforceable policy is one which simply permits or denies full access to a port on the host computer. Any further controls must be implemented within the host.

This approach provides a separate, accredited identification authenticator and insures (through the protection level) that the identity of subjects is valid throughout the life of the connection. This implementation of an NSC has many of the characteristics of a front end. When a subject initiates communication from a terminal, the initial dialogue must be with the NSC. The NSC must include sufficient terminal handling functionality to connect to each terminal and to carry on the necessary dialogues. When a connection is approved, the NSC drops out of the communication path. One way to accomplish this is for the NSC to physically stay in the path and to just pass through all subsequent communication. The difference between this configuration and a front end, as described in the next section, is in how the connection appears to the host. In this case, there are still many lines (one per port) from the NSC device to the host. The NSC is transparent to the host, appearing as a set of lines generating uninterpreted character streams, and not as a single front end device that has some functionality of its own.

As a preliminary step to a network system that includes a separate NSC it may be desirable, in some circumstances, to implement the NSC functions within the host. This subjects the NSC to all the security vulnerabilities of the host and merely supplants the normal host access control mechanism, with the added function of logical connection control. However, it will provide a smoother operational transition as the system is upgraded to include a separate NSC.

1.4.4 The NSC in a Terminal Network Connected to Front End. In this environment, the NSC and its enforcement mechanism could be placed before the front end and the situation would be equivalent to that in the previous section. However, the concept of the front end may be expanded to include the NSC and a logical connection level. When the front end includes a command interpreter as well as a terminal handler, a reference enforcement mechanism could screen the interpreted commands, enforcing a policy that controls access to various resources within the host computer. For example, a particular subject might be prevented from invoking all but certain subsystems of the host. Depending on the nature of the command interpreter, it might be possible to control access even to the granularity of individual data files controlled by the host.

In this type of situation, the NSC would be interacting with a protection level on the terminal side of the front end. In addition, the NSC would interact with a logical connection level between the command interpreter and the host.

A separate protection level could be implemented between the front end and the host, if that communication path were not secure. Since the command interpreter operates on clear text, the protection level cannot 'span' the front end.

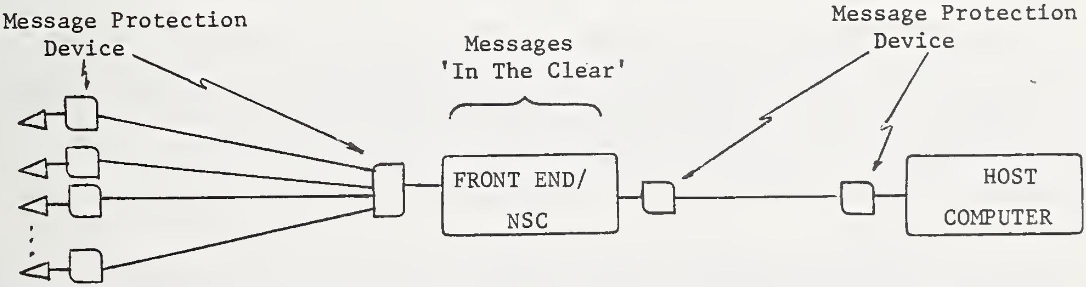


Figure 1-5. Front End/NSC in Terminal Network

If the front end did not, or could not, support a command interpreter, then the NSC would still be able to make use of the terminal handling capability of the front end to conduct its various dialogues with the subject. This approach would be similar to the NSC device as discussed in Section 1.4.3, except that the interface with the host might be different. The host would recognize the NSC/Front End as an explicit network component, rather than it being transparent to established connections as in the previous case.

1.4.5 NSC as Policy Interpreter. One of the requirements for security mechanisms is that they be isolated from unauthorized alteration. An NSC implemented in a separate component can meet this criterion, even if this component does not take an active part in the ongoing enforcement of the policy. Section 1.4.1 discussed the role of the NSC as a policy interpreter and enforcer (through its relationship with the site interface); however, there is another role that the NSC may take on. Connection control and reference enforcement mechanisms might be implemented in the hosts, or in other network components. The components could pass the parameters of the access request (collected from the subject) to the NSC. The NSC would interpret the policy and return an authorization or denial to the requesting host, which would in turn attempt to enforce the decision.

Another possible role for the NSC would be that of a secure repository for the access control information required to make a policy interpretation (the subject and object profiles). In this role, the host access control mechanisms would collect the access request parameters from the subject and receive the subject and object profiles from the NSC. The host mechanism would then perform the interpretation of the policy and make the decision itself. However, such an approach could endanger sensitive information, such as passwords, if the host is not secure.

Neither of these schemes provides the level of security that is provided by the methods discussed in Section 1.4.1; however, removing the storage of the access control profiles from the hosts would tend to increase security by isolating the access control mechanisms. An additional benefit not directly related to security is that the hosts are relieved of the operational burden of storage and update of such information. Such approaches do have merit.

A combined approach may prove most useful. In this approach, the NSC interprets access requests with a fine degree of granularity. The NSC and protection level can enforce access control only to the granularity of a complete host system. The NSC can, however, pass the results of the finer granularity access decision, along with any parameters necessary for enforcement, to the host system. The host can then provide the finer granularity of enforcement. In subsequent sections, this dual role for the NSC will be assumed. Section 3 will discuss in detail the data structures necessary to support this role.

1.5 Regional NSCs

There may be occasion to separate subjects and objects in the network into subsets, which we will call domains, each controlled by a different NSC. The motivations for this separation may vary. There may be operational limitations on capacity and performance of a single NSC; there may be administrative or political divisions which divide subjects and/or objects into regions of local control; there may be geographic separation of regions; or there may be topological separation due to the network structure. Section 1.6 discusses how the concept of domains may be applied to the interconnection of distinct networks (internetting).

Regardless of the motivation for multiple NSCs, the user interacts with what appears to be a single unified mechanism. The NSCs operate in concert with each other to provide such a mechanism.

Each subject shall be within the control of a single NSC. Similarly, each object will be under the control of a single NSC. In the most general case, there may be up to three NSCs involved in a single access request. These three are: the NSC that controls the current location of the subject; the NSC that maintains the subject profile; and the NSC that maintains the object profile. Since the function of the access control mechanism is to prevent unauthorized access to objects, the final control will reside with the NSC which controls the object.

A brief scenario will serve to illustrate the nature of the multiple NSC interaction necessary to connect a subject to an object.

1. The NSC responsible for the current location of the subject collects the appropriate identification/authentication parameters. If the subject is locally known to the NSC, the NSC performs the appropriate authentication. If the subject is not locally known, the NSC that maintains the subject profile is found. The location of this NSC might be found on the basis of a portion of the identifier offered by the subject, on a directory listing the appropriate NSCs for all non-local subjects, or by 'broadcasting' a request to all other NSCs.

2. The identification/authentication parameters is forwarded (via a secure, inter-NSC connection) to the subject profile NSC. This NSC performs the authentication.

3. The subject profile NSC will collect the access request parameters. This might be accomplished either by establishing a new connection between the subject profile NSC and the subject, or by relaying the information through the first NSC. If the requested object is local to the subject profile NSC, then it will make the access authorization determination and set up a connection if access is permitted. If the object is not local to the subject profile NSC, the appropriate object NSC will be identified. The location of the object NSC might be determined on the basis of a portion of the object name, by a directory, or by communication with all other NSCs.

4. The access request, together with the access permission portions of the subject profile, will be forwarded to the object NSC. The object NSC will determine if the requested access is to be permitted. If it is, then the object NSC will establish a logical connection between the subject (at its current location) and the object.

In order for such multiple NSC mechanisms to function properly, the following requirements must be met:

- The NSCs must be homogeneous in that they accept the same identification/authentication and authorization dialogues, that they require the same authentication and access request parameters, and that they establish connections in the same manner.

- Any NSC must be able to identify the NSC responsible for subjects and objects non-local to itself.

- The NSCs must have the capability for secure communication among themselves.

- The connection and protection levels must be homogeneous over the entire network, and be able to provide a connection between any subject and any object, regardless of location.

- The protection devices must be able to accept a connection authorization from any NSC in the network.

If the NSC distributes keys to the protection level, then the regional NSCs must coordinate their actions to establish a single common key and distribute it properly to the appropriate protection devices.

1.6 Internetting

Internetting is the interconnection of two or more computer networks. There are several alternative ways that the interconnection may be logically accomplished. A preferred relationship is one in which an additional level, an internet level, is added to the levels structure. This internet level provides to higher levels the appearance of a single large virtual network. The internet level translates the protocols of the virtual network into the protocols required by the local networks. A "gateway" (a node common to two or more local networks) implements the actual connection between the networks and transfers messages from one local network to another (see Figure 1-6).

This type of internet relationship is well suited to the NSC access control approach. The NSC, including the access control and message protection functions implemented in it, resides above the internet level and thus operates on the single virtual network. If each network has a separate NSC, difficulties to be dealt with are those of regional NSCs in a single network, since to all levels above the internet level there appears to be a single network. However, multiple NSCs may not be necessary if a single NSC could handle the size and performance requirements for the entire internet environment.

The type of internet relationship described above is an ideal, and probably is feasible only when all the networks being interconnected are designed with the interconnection in mind. This type of interconnection is not generally feasible for ad hoc connection of existing networks.

Another approach to interconnecting existing networks is for each local network to appear as an object to other local networks. The NSC would then control access to the remote network as if it were any other object. Once connected to the object network, the subject would have to interact with the access control scheme present in that network. The gateway node would appear as a host in the local network and also serve as the access point to the distant network. A subject would be required to imbed the distant network's protocols within its own local protocols for accessing the gateway. Thus the internetting functions as discussed in the previous example would be at the subject level itself.

Yet another conceivable internet relationship is a hybrid of the two previous ones. In this case the responsibility for the normal internet protocols will reside with subject, but the NSC may provide an access control protocol that appears to be a single virtual network (see Figure 1-7). If the various networks are similar enough to employ compatible NSCs, an access request for any object in the entire internet could be authorized by the same methods employed in the regional NSC example. Establishing the connection would be more complex and in some combinations of networks it may not be possible. The NSCs must take the responsibility to set up appropriate logical connections from subject to gateway, and from gateway to object. The process is further complicated if there are one or more intermediate networks between the subject network and the object network. Since there is no end-to-end message protection, the routing of messages through the various networks must be fixed (to maintain synchronization of the various distinct cryptographic mechanisms) or else identical keys must be distributed to all potentially used cryptographic devices. Once the NSCs establish the various connections necessary to connect subject to object, both parties to the communication must be aware of the various intervening networks and provide message imbedding as in the previously described internet relationship. This may not be possible unless predetermined, fixed routing is used.

In the case where an internet level makes the connected networks appear as a single virtual network, the NSCs shall have the same capability of operating as described for the case of regional NSCs in a single network.

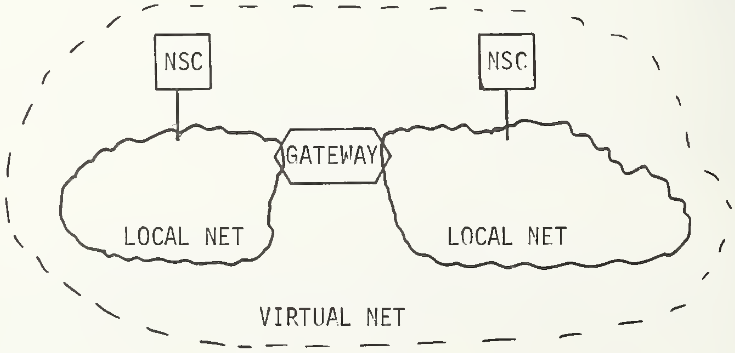


FIGURE 1-6. Virtual Network

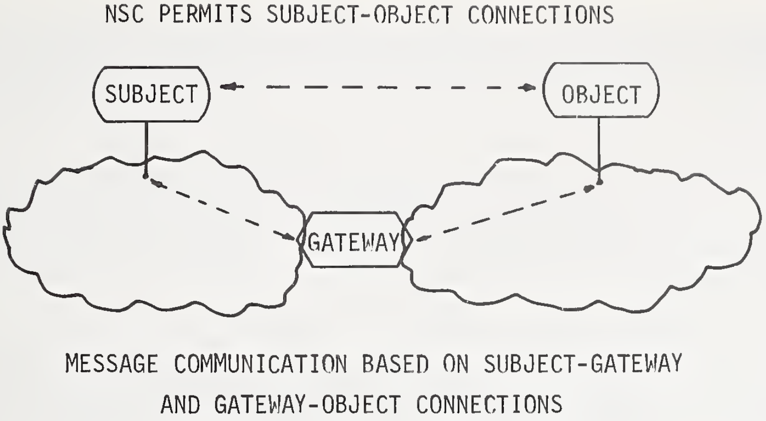

NSC PERMITS SUBJECT-OBJECT CONNECTIONS
SUBJECT
OBJECT
GATEWAY
MESSAGE COMMUNICATION BASED ON SUBJECT-GATEWAY AND GATEWAY-OBJECT CONNECTIONS

FIGURE 1-7. Hybrid Connections

Thus the requirements for functions to be placed in NSCs in an internet situation are dependent on the type of internet relationship. If the networks are treated as objects, then the NSC shall treat access requests to the other network as it would any other access request. If authorized, the NSC shall cause a connection to be created between the subject and the gateway to the object network.

In the case where the NSCs implement a single access control system, while the network connections are still on a subject-gateway and gateway-object basis, the NSCs shall operate as in the regional NSC case. In addition, the NSCs shall have the capability to determine the message route through the various gateways, and to properly establish the various logical and cryptographic connections necessary to complete the path between subject and object.

1.7 Audit and Surveillance

Audit functions at each level collect and record information for later processing. Surveillance is an active process involving gathering of data, immediate interpretation and reactive controls. The results of surveillance may be recorded as audit data.

Since the NSC is the central source for access control authorization, the NSC shall record for all authentication attempts, the identifier, the authenticator, and the disposition of the request. All access requests shall be recorded. This shall include the subject, object, qualifier (if appropriate), and the disposition of the request. A record will be kept of all connections authorized, including connection identifier, subject site, and object site. All this data shall be recorded in chronological sequence. Offline processing may be used at a later time to interpret the audit records to trace a suspected penetration or to discover distributed penetration attempts.

Surveillance functions of the NSC shall be concerned with recognizing abnormal activity as it occurs. For example, repeated invalid authentication attempts for a single user or from a single terminal may indicate a penetration attempt in progress. The surveillance mechanism will recognize such activity and initiate appropriate reactive controls. These controls might block access from the terminal or user in question, or might involve notification of appropriate personnel, who would institute extra-system controls. The record of the suspected penetration shall be retained as audit information. Similar surveillance and controls shall be provided for the access authorization functions of the NSC.

In many cases, various other levels within the network may collect audit information. For example, connection and/or message protection levels may keep records of the connections they establish. Since the NSC is a centralized location which is already equipped for auditing functions, it may be desired that the NSC perform the audit functions for these other network components. These other components would send the data to the NSC for recording. If this type of operation is required, the NSC shall have the capability to receive and record this audit data so that it may be retrieved as if the separate components had recorded it individually. That is, the data from each component shall be kept logically separated.

The NSC shall serve as the audit center for the data relating to the surveillance functions of the various components. The NSC shall also provide reactive controls for use by the various surveillance functions. For example, notification of personnel shall be coordinated through the NSC.

1.8 Size and Performance

The baseline NSC (see section 3.2) shall handle a network that consists of approximately ten computer systems and about 1000 users. The NSC shall accommodate pre-connection dialogue simultaneously with approximately 50 subjects. The approximate time to complete a pre-connection dialogue will be one minute (including typing delay, if appropriate).

This pre-connection dialogue is an additional overhead delay imposed by NSC operation. It is somewhat analogous to the overhead operation of opening a file by the operating system in a host computer. Once a connection is established by the NSC, however, there is no further overhead imposed. The NSC is not a part of the normal communication path and thus does not contribute to the ongoing communication delay. NSC overhead is involved whenever an object controlled by the NSC is created or destroyed. NSC operation does pose significant delay to the host operations of creating and destroying objects. The NSC must also be involved in adding or deleting subjects. NSC operation adds an additional delay to processing of additions or deletions of subjects.

The NSC shall be implemented on a mini-computer of current architecture. Main memory and peripheral storage requirements shall be consistent with those normally available on current mini-computers. Sections 3 and 4 of this report will describe in more detail the hardware size and performance requirements.

2.0 USAGE OF THE NBS DATA ENCRYPTION ALGORITHM

2.1 Basic Message Protection Concepts

Information is transmitted between correspondents in the network in the form of messages. Carriers (lower functional levels) perform routing and delivery functions to actually transmit the messages to the receiving correspondent. Each message is a separate, self-contained unit of information, including a text portion that carries the information being transmitted between correspondents, and a header portion that carries information for use by the message carrier (control information).

2.1.1 Security Threats. Information in messages is subject to security threats as it is transmitted via carriers. Unauthorized exposure of information is a major threat. The text portion of a message is of concern only to the correspondents. Exposure of the text to any other component or to a penetrator constitutes a security compromise. Unauthorized exposure may result from a penetrator tapping communication lines, or similarly "tapping" a processor or other component of a carrier. Message exposure may also result from messages being misrouted and delivered to the wrong correspondent. This misrouting and misdelivery might be accidental due to hardware or software errors, or it might be malicious due to a penetrator subverting a portion of a carrier.

Another threat to messages being carried in the network is improper alteration. This may be due to either accidental errors or to a malicious penetrator. Although only physical protection of all carriers can prevent alteration of messages by a penetrator, if such alteration can be detected, correspondents may ignore altered messages, perhaps requesting a retransmission from the source. A substantial number of detected alterations can degrade network efficiency, but our concern is to be able to detect such changes as easily and as quickly as possible, and to have reactive controls (such as breaking a connection or initiating human intervention) to minimize the effect of the message alteration.

Alteration can affect either message text or message header. Undetected changes to message text will compromise the communication between correspondents. Undetected alteration of a message header may result in misrouting or misdelivery.

Another threat to message communication is the introduction of spurious messages into the carrier. These spurious messages might be new messages generated by a penetrator, or old, valid messages that have been recorded and are being "played back" into the carrier.

2.1.2 Cryptographic Techniques. Threats to communication security may be countered by cryptographic techniques. Encipherment is a transformation performed on data to conceal its meaning. A reverse transformation, decipherment, renders the data once again intelligible. Due to the nature of encipherment, any data so protected will be unintelligible and thus protected from disclosure. Since lower level carriers must be able to interpret the control information in a message header, network cryptographic devices which are placed above a carrier must encipher only message text and leave the message header in the clear.

Alteration of messages may be detected if the enciphered portion of a message includes an error detection field, such as a checksum or cyclic redundancy check. The error detection field is derived from the clear text and is then included in the information to be enciphered. A penetrator cannot predict the effect on clear text of any modifications made to the enciphered text, and thus cannot make changes which appear valid after decryption.

Information which must be transmitted in the clear can also be protected against undetected alteration by use of an enciphered error detection field. It is not sufficient to simply encipher an error detection value, as it is possible to make changes to the clear text that do not change the error detection value. The clear text data must be enciphered and the error detection value derived from the enciphered data. Then the original clear text and the error detection value only are transmitted. At the destination, the clear data is again enciphered, and the error detection value re-derived and compared to the transmitted value.

Error detection fields also prevent the introduction of spurious messages, since a penetrator cannot generate a message which possesses the error check after decryption. Playback of previously recorded valid messages can be prevented by changing the encryption key (which defines the exact transformation) often. Between key changes, playback can be detected by including time stamps or message sequence numbers within the enciphered data. Such values cannot be altered without detection and will indicate when a received message is a copy of one which has been previously accepted.

There are various methods used to encipher data. Three basic encryption techniques are described. All methods rely on a secret parameter, called a key, for their secrecy. Use of a different key completely changes the transformation, thus details of the algorithm itself need not be secret for protection of the data. Only the key need be kept secret.

Long key encryption uses a very long, random sequence of values as a key. These values are successively combined with the data values to produce encrypted text. When the end of the key is reached, the process continues with the beginning of the key again. Often the method of combination is fairly simple (e.g., exclusive-or) relying on the random pattern and length of the key for protection. Long key methods require that source and destination always keep in precise synchronization, otherwise the decipherment will result in unintelligible text. Usually some means of reinitialization is required when the correspondents become desynchronized.

Self-synchronizing encryption (cipher feedback) also treats the data being transmitted as a single, continuous stream. At the source, data values in the cipher data stream itself are subjected to a complex transformation to generate the "random" long key values which are combined with succeeding clear data values to provide cipher text data. At the destination, the same complex transformation is used to recover the "random" long key values from the cipher text data. The key values are then used to recover succeeding clear text. Thus, source and destination are synchronized by the cipher text data stream.

If synchronization is lost, successive cipher text data values will reestablish synchronization. This scheme gives the advantage of a continuously changing, long key with the ability to resynchronize when data is lost. There is an overhead involved in resynchronizing, as it will take several data values to get the proper key re-established. Such methods are useful when the transmitted data contains sufficient redundancy to tolerate several lost data values, or when loss of synchronization is so infrequent that the overhead of re-establishing synchronization is negligible. Also, collecting and partitioning the data into arbitrary, and perhaps inconveniently large blocks is not required.

Block encryption treats the data as being composed of separate, independent blocks. Each block may be enciphered (and deciphered) independently of any other block. Time or position synchronization between blocks is not required.

The National Bureau of Standards Data Encryption algorithm is a block-oriented algorithm with several characteristics which make it well suited for use in networks.

1. The secrecy of the transformation is dependent only on the secrecy of the key, not on the secrecy of the algorithm.

2. The length of the key is 64 bits, eight of which are reserved for parity. Thus there are 2^{56} potential keys. The key is not so short as to make exhaustive search techniques feasible, yet not so long as to make distribution to a remote device difficult.

3. The algorithm is block-oriented; that is, data is grouped into blocks of 64 bits which may be enciphered and deciphered independently. As long as the same key remains in use, position or time synchronization of encryption with decryption is not required.

   Due to routing and transmission differences, message transit time through a network is somewhat variable. Messages may arrive at a destination in a different order than they were sent. Using the NBS Algorithm, cryptographic devices can be built which do not require position or time synchronization and are independent of the communication subsystem.

   The algorithm may also be used as the "complex transformation" in a cipher feedback scheme.

4. When enciphering or deciphering, the change of a single bit in either the key or the input text has an extensive effect on the output text. This characteristic has two implications. First, the entire correct key must be known to make use of (i.e., decipher) enciphered information. Second, alterations to enciphered text cannot produce predictable changes to the corresponding clear text.

5. Analysis of clear/enciphered text pairs does not aid in code-breaking to determine the key used. Penetrators are forced to use impractical exhaustive search techniques.

6. The NBS Data Encryption Standard is expected to be available as an LSI package. This will provide a low cost, high speed implementation suitable for use in network cryptographic devices.

2.1.3 Key Management. Message protection is placed symmetrically about a message carrier. The encryption devices at both correspondents must use the same key in order for the correspondents to communicate meaningfully. Use of different keys for enciphering and deciphering will result in meaningless text. Initialization and restart procedures must establish consistent keys.

Key update is a process by which correspondents begin to use a new key which is unrelated to the previous key. This is necessary because the strength of a key is dependent on its secrecy. The longer a key is used, the more subject the key is to disclosure, either by code-breaking or by procedural or operational compromise. Keys may also be changed as a communication control mechanism, as discussed in Section 2.2

Keys should be replaced after being in use for a certain time interval or a certain number of messages, or both.

New keys may be distributed and installed by manual methods, such as setting switches or inserting cards into an encryption device. Such methods are expensive in time and resources. It is difficult to synchronize the transition from old to new key at the various devices, and manual updates cannot be performed frequently enough to limit key exposure or for communication control. Consequently, automatic update is preferred.

One method for automatic update is to transmit to each device a new key enciphered in the old key. This can be quite effective; however, if the old key has been compromised, the new one will be similarly compromised.

Another method for key updates is for each device to have pre-stored a list of new keys. On command (which may be externally or time or event initiated), an old key is discarded and the next key is selected. Compromise of old keys is limited, since new keys are never transmitted through the network. This method requires extensive key storage within each device.

A combination approach may be most effective. Most key updates would be accomplished via remote distribution, possibly using a special separate key which is used only for key distribution. Occasionally, there would be reinitialization of keys from a stored list of new keys.

2.1.4 Other Benefits to Encryption. There are other beneficial effects to using encryption in addition to countering the previously mentioned threats. Encryption can be the means of providing implicit, mutual authentication of the identity of network devices. Common knowledge of an encryption key permits communication to occur. If each pair of correspondents uses a distinct key, they can be assured that they are not communicating with a counterfeit device masquerading as the apparent correspondent. The same protection is provided against misdelivered messages. If each pair of correspondents uses a distinct key, messages delivered to the wrong destination will not be usable.

Encryption keying may be used in the authentication of persons as well as devices. As proposed by Branstad [3], users would provide a pre-assigned key to a device at the user's access point (e.g., terminal). The key would be provided either from memory, or be read from some storage medium such as a magnetic card. Some data value would be enciphered using this key and the result sent to the Network Security Center (NSC).

The NSC would retrieve the stored key that corresponds to the user's supposed identity (identity is transmitted in the clear). If decipherment resulted in the proper data, the person would be presumed to be who he says he is. This system uses a secret password (the encryption key) that can be verified without transmitting the actual value through the network. Subsequent communication would be protected by a dynamically established encryption key. The actual key values must be entered into the NSC through some secure means. This might be through some local input mechanism, or remotely through the network, protected by message encipherment.

A slightly different approach to the same concept improves security, especially when authentication must be carried out by a component less secure than the NSC. This scheme uses encryption as a one-way transform, relying on the property that even if both clear and enciphered text are known, a penetrator cannot determine the key used. As in the other scheme, a user supplies a password which is used as a key to encipher a data value. The enciphered data value is transmitted through the network. The difference is at the authentication site, where only the enciphered data value for each user is stored. The authentication site need know only the enciphered value, not the key used; if the enciphered values correspond, the identity is authenticated. The value of this scheme is that the stored data values need not be kept secret, since knowledge of the enciphered value does not aid in determining the key used. Thus security is not compromised if the list of data values is exposed.

In these schemes, it is important that the communication paths in the network be protected from message playback or introduction of spurious messages, so that a penetrator cannot play back valid authentication messages, or generate new ones based on the enciphered values read from previous messages.

These systems provide excellent security for passwords, but have some limitations. Each access point must be equipped with a device capable of selectively enciphering data, and capable of interacting according to the NSC protocols. If it is expected that the same device will provide end-to-end dynamically keyed protection, either the user or the device will be forced to perform all functions that normally lie above the message protection level (e.g., message formatting, logical connection control).

2.1.5 End-To-End Vs. Link Encryption. Encryption can be used either to protect messages at all times from the original sender to the ultimate receiver (end-to-end encryption) or may be used only on the lowest level communication links between intermediate points in the carrier (link encryption) (see Figure 2-1).

End-to-end encryption protects against security threats at all points between sender and receiver. Link encryption achieves protection between intermediate nodes, but fails to protect against security threats at the nodes themselves. Data is in the clear within the intermediate nodes, and is thus susceptible to compromise. Only the nodes themselves are authenticated as to identity; a subverted node can misroute messages without being detected. End-to-end encryption authenticates the identity of the source and ultimate destinatio Misrouting within the communication subnetwork will be detected if source-destination pairs use distinct keys.

Link encryption is the form of protection in use today. End-to-end encryption requires a specialized Network Cryptographic Device (NCD) that can separate header from text, and encrypt only the text. The NCD must also multiplex several logical connections, each with a distinct key. One devide can handle communication with several other sites, maintaining a distinct key for each source-destination pair.

The NBS Data Encryption Standard, because of its block encryption and other properties discussed in Section 2.1.2, is best suited for a Network Cryptographic Device. Although such devices do not currently exist, they can be built using current technology.

2.2 Encryption as A Communication Control Mechanism

In addition to countering the communication threats discussed in the previous section, cryptographic devices may be used as a means of controlling communication through a network.

2.2.1 Error Control. The inclusion of an error detection field within the message to be enciphered was mentioned in Section 2.1.2 as a means of detecting alteration of a message by a penetrator. It is also an effective means of detecting non-malicious alteration (e.g., data transmission errors). Due to the unpredictable and extensive effect on deciphered data of a single bit change in enciphered data, very simple error detection methods can be quite effective when coupled with encryption.

2.2.2 Access Control. Cryptographic mechanisms can serve as the means of enforcing network access control decisions. Keys may be managed to permit only those connections which network security policy permits [3,4].

The Network Cryptographic Device (NCD) utilizes a distinct key for each logical connection. Unless a pair of correspondents have a mutual key, it is impossible for them to communicate.

The Network Security Center (NSC) could manifest its decision to permit a connection between correspondents by distributing a key to both parties. Similarly, a decision to refuse a connection could be manifested by withholding a key. The NCDs must permit only connections which are established (keyed) by the NSC, thus there must be a close cooperative relationship between the NSC and the cryptographic devices.

Rather than distribute keys to the cryptographic devices, it is possible to simply inform the cryptographic devices that a connection is permitted, and allow the devices to establish the actual keys themselves. Of course, the devices must only key a connection if it is explicitly authorized by the NSC. Connection will be allowed with the NSC, however, so the authentication and authorization dialogues may take place.

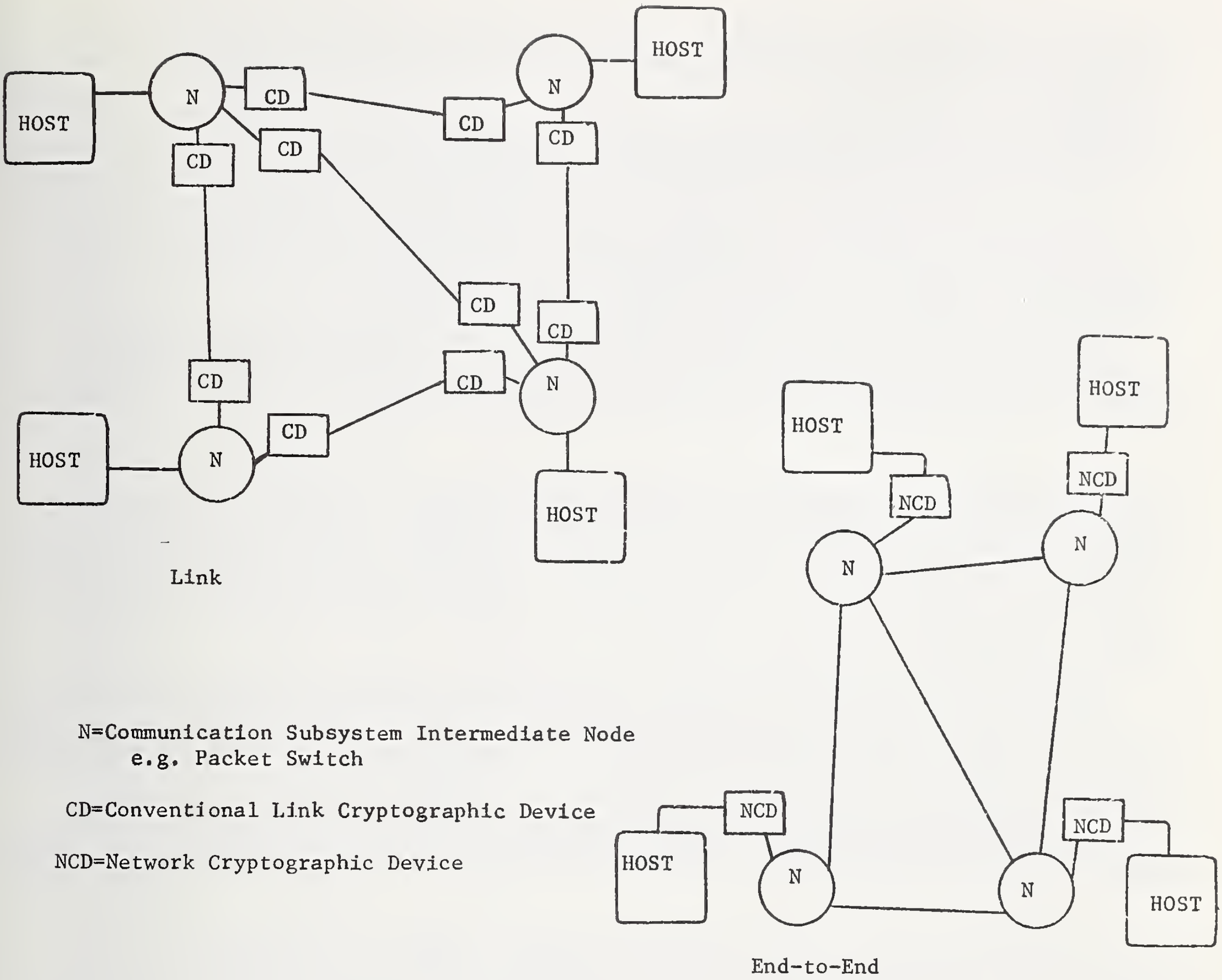

HOST
N
CD
HOST
N
CD
CD
CD
CD
CD
CD
CD
CD
CD
HOST
N
CD
N
HOST
Link
N=Communication Subsystem Intermediate Node
e.g. Packet Switch
CD=Conventional Link Cryptographic Device
NCD=Network Cryptographic Device
HOST
HOST
NCD
NCD
N
N
NCD
NCD
HOST
N
N
HOST
End-to-End

Figure 2-1. End-to-End Vs. Link Protection

In any electronic key distribution scheme, transmission of keys must be protected by encryption. The components that send and receive keys (whether they be NSCs or NCDs) must use distinct keys for protecting key distribution.

2.2.3 Example Scenario. An example will be helpful in illustrating the use of encryption keying as an access control mechanism.

A user (U) at a terminal (T) desires access to a process (P) at a distant host (H). Before requesting access to P at host H, the user must carry on a dialogue with the NSC in which he identifies himself and provides additional information, such as a password, to authenticate his identity.

All messages in the user-NSC dialogue are protected by Network Cryptographic Devices (NCDs) at T and at the NSC. The device at the NSC is a special cryptographic device called the Master NCD. Each NCD maintains a distinct predefined key known by the Master NCD, and will use this key for protecting communications with the NSC.

Up to this point, the scenario is similar to that of a user attached directly to a host with an access control mechanism. To the user, the NSC mechanism appears to be a conventional access control system. The difference is the way in which the NSC creates connections.

If the requested access is not permitted, no connection will be established between U and host H, and U will not be able to access any resource at host H.

If the required access is permitted, the NSC will distribute (through the Master NCD) an encryption key to the ICDs at U and at H. The transmission of each new key will be protected by enciphering each key update message in the appropriate predefined key. U will now be able to communicate with H using the newly distributed key.

The NSC will pass to the access control mechanism of H the result of the access request decision and, in this case, the process name P. The access control mechanism within H will be responsible for restricting U's access to process P only. Cryptographic keying can enforce access control only to the granularity of a complete host.

2.2.4 An Alternative to Centralized Key Distribution. An alternative approach to key distribution and access control separates the functions of encryption keying and access control enforcement into logically separate levels. Although both levels may be implemented within the same device, they are logically independent. Access control enforcement is performed by a logical connection level, while encryption keying is distributed among the various NCDs rather than being centralized in a Master NCD. Encryption keying no longer serves to enforce access control decisions directly, but still serves to authenticate both ends of the logical connection.

In this type of system, the example scenario is the same until the dynamic working key for the connection between U and H is to be established. When the NSC approves the access request, it will inform a trusted logical connection level (which has responsibility for access control enforcement) at both U and H. The connection will be indicated by a unique connection identifier generated by the NSC. The logical connection level will establish connections only when authorized by the NSC. Since the logical connection level must be trusted to establish only approved connections, it must be implemented in a trusted network component outside a host.

To implement a secure logical connection across the network, the facilities of a self-contained message protection level will be used. When a new connection is to be established, the logical connection level will request that a new, secure cryptographic connection be established by the message protection level. The cryptographic devices at U and at H will communicate between themselves to establish a distinctly keyed, cryptographically secure message path for the logical connection level. Each cryptographic device must have stored a distinct key for communication with every other cryptographic device. The device at the NSC is no longer different from the other devices; all are identical.

These cryptographic devices are more complex than the previous NCD, and must have greater key storage, since they must store a private key for each other site rather than just for the NSC. Thus they are less attractive for use in networks that have a very large number of sites. However, the separation of encryption keying from access control allows the cryptographic devices to be used without an NSC in situations where NSC-type access control is not appropriate [5]. Since the NSC access control is enforced by logical connection control, rather than encryption keying, the NSC could be used without NCDs in a system where cryptographic protection is infeasible or not necessary.

2.3 Internetting

Internetting is the creation of a network of networks by interconnection of two or more computer networks. Section 1.6 discussed some possible ways to accomplish the interconnection of networks. Unless the internetwork connections are standardized, and internet functions create a single virtual network, and the cryptographic level is above the virtual internetwork, end-to-end encryption is effectively precluded. For end-to-end encryption to be possible, the cryptographic methods (and devices) must be uniform throughout the entire internetwork. The representation of control (header) and text information must be standardized and consistent in all the networks that are interconnected.

When diverse networks are internetted, messages are subject to exposure, alteration, or removal at the gateways (where they are in the clear). Also, there is no mutual authentication of the identity of the end points of the connection. The situation is analogous to the comparison between link encryption and end-to-end encryption within a single network. In the internet case, message protection within each constituent network (even if it is end-to-end within that network) is analogous to link encryption in a single network, with the internet gateways taking the role of the communication system intermediate nodes.

Internetting will require that Network Security Centers for each network cooperate with one another. This will require standardization of NSC protocols. Also, key distribution is greatly complicated when keys may be distributed by more than one NSC and possibly to different types of cryptographic devices. This will require special protocols, except in the case of the single virtual network, where normal regional NSC protocols will be sufficient.

2.4 Networks with a Single Host

A common form of network involves a number of remote terminals connected to a single host. As described in Section 1.3.6, the communication system, and thus the cryptographic devices, occupies a different place in the logical structure of the network than in an inter-computer network. Thus the power of the cryptographic devices for access control is limited.

Cryptographic devices are placed between the user and the user-level functions. The user cannot be expected to perform the functions of message formatting and command interpretation for himself, thus messages must be in the clear for the user-level functions. The cryptographic devices are essentially link encryption devices, since communication protocols are often very simple, and in general each line serves only one user.

The NSC may still use remote keying as a means of access control enforcement. However, the NSC must interpose between the terminal and the host. Otherwise authentication and access request authorization messages must pass through the non-secure host, negating many of the advantages of the NSC.

2.5 Other Uses for the Data Encryption Standard Algorithm

The NBS Data Encryption Standard algorithm may be used for purposes other than communications protection and access control. The most obvious use is for protection of data files, using distinct keys to provide separation and protection. The block nature of the algorithm is well suited to random access techniques. The prime difficulty is in finding a secure way of maintaining the keys under a possibly non-secure operating system. One possible approach is for the files and encryption keys to be maintained in a facility remote from the non-secure host. This "data vault" would have access controlled by the NSC network mechanisms, and would be trustworthy in the storage of keys and data. This data vault might be a separate, secure device attached to a host, or it might be a self sufficient network site that operates as a host itself.

The use of encryption to provide a one-way transformation for personal identification has been mentioned in Section 2.1.4. The same process can be used in other circumstances where a one-way transformation is required. Whenever data values must be verified, yet protected from exposure of their actual value, one-way transformations are valuable.

Finally, in circumstances where a non-predictable sequence of values is required, the NBS Data Encryption Algorithm can be used with a simple pseudo-random number generator to generate both data values and keys to produce quite unpredictable (yet repeatable) sequences of output values.

3.0 THE NETWORK SECURITY CENTER - DATA STRUCTURES

The general function of the Network Security Center (NSC) has been described in Section 1. Section 1.3.2 described the role of the NSC in authentication of identity and authorization of access requests. This section describes data structures to be used in the NSC to support these NSC functions.

This section will first describe the authentication and authorization modules at an implementation independent level. Then an example implementation of a high-level specification model will be discussed. In this high level specification and implementation example we will focus on a particular model of identification/authentication and access request authorization.

Identification/authentication was discussed in previous sections. All schemes were similar, involving the checking of an offered <identifier, authenticator> pair against a similar stored pair. The simplest form occurred when the values compared were the actual values stored and the actual values input. Another scheme involved a one-way transform performed on the offered authenticator prior to comparison. Both these methods involve identical processing at the authentication module of the NSC.

A third scheme involved using a stored key to decipher a received value which was enciphered by a user's individual key. In this situation, the keys themselves are the authenticators. Rather than simple equality, the comparison operation involves a transformation.

Our specification of the authentication module will be for a simple equality comparison authentication scheme. The specification could easily be extended to include the more complex comparison process of the third scheme mentioned above.

The conceptual model for Access Authorization is a three-dimensional access control matrix [6]. The matrix consists of a Boolean entry for each possible combination of subject, object and access right. The Boolean indicates whether or not that subject may access that object with that access right. Most entries in the matrix are false (i.e., most of the possible access combinations are not permitted).

3.1 General Model

The general model of the NSC data structures will specify the necessary NSC data elements and the relations between them in an implementation independent manner. The high level model specifies the external data structure and functions performed. Although particular implementations of an NSC may differ, externally they will all conform to the high level model.

The implementation independent specification serves as a general description of the system, but in many cases a formal specification of the NSC may also be required. A formal specification is the first step in many top-down development methodologies, especially those that attempt formal verification of a system. Adherence to such a top-down methodology will be of benefit in those cases where formal verification of the system is required.

In order to describe the NSC data structures in a manner that does not imply a particular implementation, the specifications in this section will be given in terms of sets of elements. A set is an unordered collection of elements, each of the same form. There are no duplicate elements in a set. The elements of the sets in these specifications will be tuples. A tuple is an ordered set of elements (e.g., 2-tuple is an ordered pair).

The types of tests made will be tests for values of elements of a tuple and tests for membership of a tuple in a set. The general form of the specification includes narrative description and programming language type control structures.

It should be noted that this specification describes only the authentication and authorization modules of the NSC and is not a complete specification of the design of a complete NSC.

3.1.1 Authentication. The model of authentication consists of comparing a pre-stored pair <subject identifier, authenticator>, with a similar pair offered by the subject. The NSC data structure used is a set called AUTHENTICATION. AUTHENTICATION contains 2-tuples of the form <subject identifier, authenticator>. There will be at least one such tuple in AUTHENTICATION for each valid subject known to the NSC.

The authentication module in the NSC acts as a predicate. The input is a tuple <subject identifier, password> offered by the subject. The output is either TRUE or FALSE, depending on whether the password is a proper authenticator for that subject identifier. The authentication module will perform range checking on the form of the input parameters.

To authenticate the tuple <subject identifier, password>:

```
IF subject identifier is valid AND password is valid AND
    <subject identifier, password> is a member of AUTHENTICATION

            THEN TRUE
            ELSE FALSE;
```

3.1.2 Authorization. The conceptual model of the access authorization structure is a three-dimensional matrix. The axes are Subject Identifier, Object Identifier, Access Right. The entries in the matrix are Boolean flags that indicate whether the corresponding triple (<subject identifier, object identifier, access right>) defines an authorized access. This can be represented in a set model of NSC data structures as a set AUTHORIZATION. AUTHORIZATION contains 4-tuples of the form <subject identifier, object identifier, access right, Boolean flag>. The set will contain one such tuple for each possible combination of subject identifier, object identifier, and access right. The Boolean flag will be TRUE if the specified triple defines a permitted access and FALSE if it defines a prohibited access.

The object identifier element has some additional structure to it. Object identifier is itself a 2-tuple of the form <host name, resource name>. When the resource name is null, it indicates the entire host treated as an object.

To determine if an access request of the form <subject identifier, object identifier, access right> is authorized:

```
IF <subject identifier, object identifier, access right, TRUE> is
    a member of AUTHORIZATION

            THEN TRUE
            ELSE FALSE
```

It can be seen that this reflection of the three-dimensional access matrix in the set model contains a great deal of unnecessary information. There is a tuple in AUTHORIZATION for every possible combination of subject identifier, object identifier, and access right. Since most of these tuples define prohibited accesses, it is unnecessary to include them explicitly in the data base.

The explicit inclusion of all possible tuples also greatly increases the complexity of operations on the data base. To add or delete an object, a tuple must be added or deleted for each subject-access right combination, even for subjects that had no permitted access to the object. A similar concern applies to operations involving subjects.

Thus we can consider modifying the set AUTHORIZATION to remove unnecessary tuples. The new set AUTHORIZATION is made up of 3-tuples of the form <subject identifier, object identifier, access right>. The relationship between the new set and the old one is that the tuples in the new one are only those tuples that had a Boolean flag of TRUE in the old set. That is, the new set contains only tuples which define permitted accesses. Now to check on authorization, the test is for membership of <subject identifier, object identifier, access right> in the set AUTHORIZATION.

3.1.3 Modification. Up to this point, the specification has involved only retrieval operations, which make up most of the operations of the NSC. However, there is a requirement for updating the access control information stored in the various data bases.

To change the authenticator for a subject, the tuple in AUTHENTICATION that contains the elements <subject identifier, old authenticator> is removed and a new tuple <subject identifier, new authenticator> is added to the set. It is assumed that a subject always has the right to change his own authenticator. In addition, a special subject, called the system administrator, may also be permitted to modify authenticators. The system can define this right by treating the set AUTHENTICATION as a resource (i.e., the object identifier is <NSC, AUTHENTICATION>) and giving the system administrator the access right MODIFY.

The module that changes a subject's authenticator requires three inputs: the <subject identifier, password> of the subject making the change, the <identifier, old authenticator>, and the <identifier, new authenticator>. The definition of the module is:

```
IF <subject identifier, password> is a member of AUTHENTICATION AND
    (subject identifier = identifier OR
    <subject identifier, <NSC, AUTHENTICATION>, MODIFY> is a member of AUTHORIZATION)

            THEN  remove <identifier, old authenticator> from AUTHENTICATION,
                  add <identifier, new authenticator> to AUTHENTICATION
            ELSE  null;
```

Note: If the subject identity has previously been authenticated, the <subject identifier, password> check is not required.

To modify a subject's access rights to an object involves adding or removing tuples from AUTHORIZATICN. To specify which subjects can modify the access rights to an object, an access right called OWNER is used. This access right permits the subject to modify the access rights of any subjects to that object.

The inputs to the delete access right module are <subject identifier, password> of the subject doing the changing and <identifier, object identifier, access right> being removed. The specification is:

```
IF <subject identifier, password> is a member of AUTHENTICATION AND
    <subject identifier, object identifier, OWNER> is a member of AUTHORIZATION

            THEN  remove <identifier, object identifier, access right>.
                  from AUTHORIZATION
            ELSE  null;
```

Note: If the subject identity has previously been authenticated, the <subject identifier, password> check is not required.

The add access right module is the same except the operation is to add the tuple <identifier, object identifier, access right> to AUTHORIZATION rather than remove it.

To add an object to the data base, a tuple which defines the access right OWNER for the object is added to AUTHORIZATION. Other access rights are added through the add access right module.

To remove an object from the data base, all tuples with that object identifier in AUTHORIZATION must be removed. Some check will be necessary to determine if deletion of an object is permitted. One possibility is to permit only subjects with the access right OWNER to delete an object. Another possibility is to have an additional access right, DELETE, which permits deletion of an object.

To add a subject to the data base, a tuple <subject identifier, authenticator> must be added to AUTHENTICATION. The add access right module is used to give the subject access rights to objects.

To remove a subject, all tuples with the subject's identifier in both AUTHENTICATION and AUTHORIZATION must be removed.

Adding a new type access right is simply a matter of defining the new type and adding tuples to AUTHORIZATION containing that access right where appropriate. Deleting an access right type involves removing all tuples from AUTHORIZATION that contain that access right.

3.1.4 Subject and Object Grouping. In many cases, it will be possible to organize users and/or objects into groups. These groups would have identical access control specifications.

In the case of subject groups, the group name would take the place of a subject name in specifying access to an object. Rather than a separate entry for each subject in the group, there need be only one entry for each <object, access right> with the subject group name becoming a "pseudo-subject."

An object group would specify a collection of objects that are treated identically in terms of access control. In this case, the group name would become a "pseudo-object."

Although such grouping can provide considerable savings of space, it can complicate the operations performed on the data.

For example, to determine if a subject has access right to an object, a check must be made not only for a tuple which contains subject and object, but also for a tuple that contains subject and object group name for all groups the object is a member of.

A similar concern applies to subject groups. Determining whether a subject has an access right to an object necessitates checking not only tuples with the subject and object, but also tuples with subject group name and object for all groups the subject is a member of.

Section 3.1.5, which deals with distribution and multiple NSCs, will discuss a form of grouping. The implementation example of Section 3.2 will illustrate subject and object grouping.

3.1.5 Distribution and Multiple NSCs. In Section 1, the possibility of multiple NSCs was presented. In this section we will present an organization for NSC data structures to facilitate use of multiple NSCs in a single network.

Use of multiple NSCs requires distribution of the NSC data bases, and cooperating procedures to enable NSCs to operate together.

Conceptually, we will still talk about a single unified NSC data structure. The distribution of data structures to multiple NSCs will be expressed as portions of the single (conceptual) data structure that would exist if there were a single NSC.

Distribution of NSC data structures will take place in two dimensions, by subject and by object.

Each subject will be in the domain, or control, of one NSC. Each host (and thus objects at that host) will be in the domain of one NSC.

The distribution of the data base can be described in terms of the previously described sets. For each NSC there is a set of subjects which are in that NSC's domain. Each subject is in the domain of one and only one NSC.

The authentication data base is distributed based on subject. At each NSC is stored the subset of AUTHENTICATION that contains tuples with subject identifier equal to a subject in that NSC's domain.

The authentication process for a particular user will be carried out by one NSC. As described in Section 1, an NSC receiving a request from a user not in its own domain will pass on the authentication function to the NSC responsible for that subject. Determination of which NSC is responsible might be by a portion of the subject identifier, by looking in a directory, or by broadcasting the function request to all other NSCs.

The authorization data base is distributed somewhat differently. Recall that objects have a structured identifier of the form <host name, resource name>. Objects of the type <host name, null> represent an object of granularity of an entire host system.

At the NSC responsible for each subject is stored the portion of the authorization data base that defines permitted accesses to the granularity of complete host systems. At each NSC is stored a portion of the set AUTHORIZATION. This portion of the set is defined as those tuples where the subject identifier is a subject in the domain of this particular NSC, and the object identifier is a host name with a null resource name. That is, only tuples that define access to complete host systems are included.

This permits checking if a subject is authorized to access any object at a host. It can be used as a pre-check to possibly eliminate further detailed checking,

The granularity of a complete host system is as fine a granularity as the NSC and cryptographic devices can enforce. In some applications, it may be sufficient for the NSC to store access control information only to this granularity.

However, it may be required that the NSC store and interpret access control information to a finer granularity, even though enforcement of the finer grain access controls is the host's responsibility.

In this case, each host will be in the domain of a single NSC. Each NSC will maintain that portion of the set AUTHORIZATION that concerns resources at hosts in its domain.

The resource portion of AUTHORIZATION stored at an NSC is defined as all the tuples where the host name part of the object identifier is equal to a host in that NSC's domain.

An access request operates as follows. First, the NSC which is responsible for the subject performs authentication of the subject's identity. The same NSC screens the access request to determine if the subject is permitted access to any resource of the host where the requested object resides. If not, the request would be rejected without further processing.

If the subject is allowed access to the host, the NSC responsible for that host would perform the finer granularity access authorization check. If the target host is not "local" to the subject's NSC, the request would be transferred to the NSC responsible for the host to complete the access authorization check.

If the finer granularity check passes, then the target host is informed, the subject is informed, and a cryptographic connection is established.

It is possible that a subject may not be initially connected to its "own" NSC. For example, subjects that are currently at, or are controlled by, a host will initially be connected to the NSC responsible for that host. If the subject is not "local," the request will be transferred to the NSC responsible for that subject. Of course, that NSC may in turn pass the request on to another NSC if the object is not "local" to it.

Thus multiple NSCs operate by distributing the conceptually unified data base and by passing portions of the access control processing between NSCs as appropriate.

3.2 An Implementation Example

Section 3.1 described, in an implementation independent manner, the NSC data structures for identification/authentication and access request authorization. The set notation that was used avoided details of implementation. This section will present an example implementation of the NSC data structures.

This section is an example, not a precise prescription for building an NSC. The example serves to illustrate how an NSC might be implemented. Each situation is different, however, and the circumstances of a different situation may make this particular implementation inappropriate. The example is still of value, since it illustrates the range of issues that govern the design of a specific implementation.

3.2.1 Scope of Example. This example will initially consider a network of approximately ten network sites (hosts) and about 1000 network users. It is assumed that each subject will require a new access via the NSC about every 20 minutes. It is also assumed that the authentication/authorization dialogue will be complete within one minute, including appropriate typing delays and user "think time."

Each subject is treated as a user. This is obvious for users at terminals; other entities (such as processes) are considered to be acting on behalf of a particular user.

For the example, user identifiers will be assumed to be eight alphanumeric characters long, which provides 36^8 (more than 2×10^{12}) unique identifiers, ample range to accommodate any foreseeable network population.

The size of the authentication value is assumed to be eight bytes, which is consistent with the length of an NBS standard encryption algorithm key, if a scheme where the authenticator is used as an encryption key is adopted. If a password scheme is adopted, any value longer than eight characters will likely prove too difficult for users to memorize.

This example considers objects of the form of data files located at network hosts. Since each host manages its own objects, the actual nature of the objects may be different. However, the NSC only distinguishes between objects by names, and it is up to the host to actually implement access to the object. For the purposes of the NSC, all objects will be considered to be files.

For the example, the average object identifier is considered to be 17 characters, with actual sizes covering a range of about eight characters to 40 characters. A typical host identifier will be assumed to be six characters.

The actual nature of the access rights will not be important to the remainder of this example. However, some probable access right types can be indicated. Since objects are considered as data files, access rights such as read, write, modify, and append make sense. Since a file may contain a program, and thus become a process if executed, an execute access right may also be required.

It is assumed that access rights may be encoded in two bytes (16 bits). Our examples may indicate access rights as character strings, but in actual implementation an encoding will be used (contrasting with the other data elements, which will be stored as characters)

In Section 3.1, grouping of subjects and objects is mentioned. The extent of such grouping will vary considerably with different circumstances. In a network which consists entirely of transaction-oriented systems, most users might fall into one of relatively few user categories. Depending on the nature of the system, objects may also fall into relatively few categories; perhaps entire host systems would be the finest granularity required for access control.

In systems that support general-purpose programing or access to major data bases, there would probably be little grouping of subjects or objects. The access profile for each user would be more individualized than in a transaction-oriented system.

The figures used here are examples only, and each individual situation must be modified to reflect unique circumstances.

It is estimated that the normal user will have a range of ten to 50 objects in his access profile. For the example, 25 objects is adopted as an average figure. To take account of object grouping, five of these objects are considered to be represented by an object group. It is assumed ten users share an object group. It is also assumed that there are about twenty subject groups, each with an object profile similar to an individual user.

3.2.2 Hardware Configuration. The hardware configuration assumed for the NSC is current generation minicomputer. No particular model or architecture will be specified, but generic mini characteristics will be assumed. Section 5 will recommend some desired hardware features that will aid in implementation of the control structure.

Thus the machine being considered most likely has a 16-bit word, about one microsecond cycle time, and a maximum memory capacity which may range from 32K words to 128K words.

Disc storage is available in three ranges of size. First, approximately two megabyte capacity cartridges, with average access time of about 75 milliseconds and a transfer rate of about 11 microseconds per word. Second, a pack of about 41 megabyte capacity, with access time about 50 milliseconds and a transfer rate of about 7.5 microseconds per word. Third, the large 3330 type packs with capacity ranging from 90-200 megabytes. Average access time is about 40 milliseconds with about 2.5 microseconds per word transfer rate.

Disc storage is generally available only in fixed length, sectored configuration of size 256 or 512 bytes. For large discs a 20-bit value is sufficient to address an individual sector.

3.2.3 Size of Data Base. In Section 3.2.1 the assumed size of the various data fields was given. These figures are repeated here for convenience.

User Identifier	8 characters (bytes)
Authenticator	8 bytes
File Identifier	17 characters
Host Identifier	6 characters
Access Right Code	2 bytes

Each user was presumed to have 25 objects in his profile. Since five of these are in an object group, 21 objects per user are assumed. Each object requires 25 bytes to specify (including Access Right Code). Including the User Identifier and the Authenticator plus an allowance for overhead in the data structure for pointers and unused space, we estimate 750 bytes storage required per user.

Each subject group can be treated as an additional subject; thus the network is considered as having 1020 subjects. This gives a storage requirement of 765,000 bytes.

There are about 100 object groups of five objects each. This adds another 17,500 bytes for a total of 782,500 bytes of storage required.

In the distributed version of the NSC data structures, there is an additional storage requirement for the separate host-granularity access lists at the subjects' NSC. The other totals should remain approximately the same, even if distributed differently. It is estimated that the additional requirement, owing to the distribution, is 10-25%.

Thus, for the baseline system, a two-megabyte disc cartridge is of sufficient size to support the NSC data structures.

A fixed head disc (one megabyte) might be just adequate for the initial baseline system, but provides little room for expansion and is more costly. Thus we do not consider it appropriate to consider use of the faster access time fixed head disc.

3.2.4 Factoring. The model of NSC data structures presented in Section 3.1 first specified explicit storage of <subject, object, capability> for each possible combination. The model was modified to reduce the requirement to only combinations defining permitted accesses. When actually implementing the authorization data structure, the storage requirements can be further reduced by the technique discussed below, called factoring.

When storing the authorization information, all the tuples which have the same value for the subject (or the object) can be stored together. The common value need only be explicitly represented once for all the tuples which have this common value. Thus, when factoring one common value, storage will actually contain

a subject identifier associated with a set of <object, access right> pairs

or

an object identifier associated with a set of <subject, access right> pairs.

(Note that an object identifier is actually the <host name, resource name> pair. Within object identifiers, the host name may also be factored for additional saving of storage space.)

It must be decided which factoring is most appropriate. A subject-oriented organization facilitates subject centered actions, such as simultaneously checking a subject's access to more than one object, adding or deleting a subject, or listing all accesses permitted to a subject.

An object-oriented organization facilitates object centered actions, such as adding or deleting objects, or listing which subjects have access to an object.

A decision on which organization is most appropriate must be based on expected relative frequencies of subject-oriented and object-oriented actions.

One criterion for determining the appropriate organization might be the number of items that must be searched to find the desired information. We would like to base the organization of the data item that leaves the smallest number of items to search. For example, we can compare the average size of {<object, access right>} for each subject to the average size of {<subject, access right>} for each object.

In the example, it was assumed there are 25 <object, access right> pairs for each subject. There was no presumption made about the number of <subject, capability> pairs for each object. This could vary greatly for different type objects, but in many cases the average may be less than the number of objects accessible by a subject.

This might indicate an object-oriented organization. However, there are several reasons to prefer a subject-oriented organization. First, there is no object-oriented counterpart to multiple-object access requests from a subject.

Second, the authentication information is stored in a subject-oriented manner. Since authentication is likely to be followed by an access request from the same subject, it makes sense to store the authentication information with the authorization information. Then, when the authentication information is located, no further search will be necessary for the upcoming access request.

The foregoing discussion applies to a single unified NSC where all the data base is implemented at the same location. In the model for a distributed NSC, part of the factoring is established by how the data is distributed.

The distributed model specifies a portion of the data as subject-oriented. At the subject's NSC, it makes sense to organize authentication and host-granularity authorization information together based on subject identifier. In this portion of the NSC, the object list for each subject contains only entire hosts. There will be fewer hosts in the network than there are subjects handled by each NSC. Thus organization by subject reduces to searches of short object lists.

The portion of the distributed NSC that handles object data may conceivably be organized either by subject or by object. Individual circumstances will determine the optimal organization. The advantage of associating authorization data with authentication data is no longer a concern and the probability of access requests for multiple objects at a single host is less than for the network as a whole. Thus a combination of approaches may be most feasible, with subject authentication and host-level authorization organized by subject, and finer granularity access authorization organized by object at the NSC responsible for each host.

Independent of the decisions on distribution or factoring, some means are necessary to locate the data in question. The selection of the method must be based on the circumstances of each individual case. However, due to the number of individual data entries involved and the required responsiveness, techniques such as search trees, directories, hash functions or some combination will probably be required.

In some circumstances, the storage savings resulting from factoring may not be important. If storage capacity permits storing the complete <subject, object, access right> triple for each permitted combination, a hash function based on both subject and object would provide quick access, at the cost of additional storage due to not using factoring.

3.2.5 The Implementation Structure. The tutorial "Design Alternatives for Computer Network Security" [1] outlined a linked-block structure for implementing the NSC data structure. This structure assumes a subject-oriented organization and factoring.

3.2.5.1 Linked-Block Structure. The basic structure is a user-profile block, which contains user identification, user authentication data, and access authorization for objects. Considerable factoring, and hence reduction of storage requirements, is possible. Figure 3-1 illustrates an example of a user-profile block.

The access control information is of several different types. The simplest consists of direct object-access right pairs (a). If there are several distinct access rights to be indicated for a single object, the user-profile block can contain a pointer to an access right list for that object (b). Anywhere a single object would normally occur, a pointer to an object group can occur. Two pointer types are illustrated: (c) indicates an object group with each object having a distinct access right list; (d) indicates an object group that has the same access right list. Note than any of these groups can be referenced from more than one user-profile block.

Subject groups have a special form of user-profile block, which contains the subject identifier for each subject in the group. The objects in the access control portion of the block are the same as for an individual subject.

Utilization of subject groups can greatly reduce storage requirements in certain circumstances, but cause substantial extra processing in a subject-oriented organization. Whenever a subject initiates an access request, the NSC must check not only the subject's user-profile, but must also determine _if_ the subject is a member of any access groups, and then must search _each_ of the appropriate group-profile blocks. One way of handling the problem of determining to which groups a subject belongs is to add another data element to the user-profile block. This would be a pointer (or as many as required) to the group-profile block for the subject group.

As discussed in Section 3.2.7, the major impact on service time is disc accesses. Each level of indirection probably requires an additional disc access. It appears that inclusion of a subject in a subject group multiplies the service time by approximately the number of profile blocks (user or group) that must be searched. Use of subject groups in a subject-oriented organization should be very carefully considered.

3.2.5.2 Storage Organization and Accessing. The linked block structure has been presented as if each block were sequentially organized within main memory. In reality, the data will be stored in sectors on external disc storage. Section 3.2.7 points out that the number of I/O requests is the dominant factor in request service time. Thus it should be the goal of storage organization to minimize the number of disc accesses necessary to process a request.

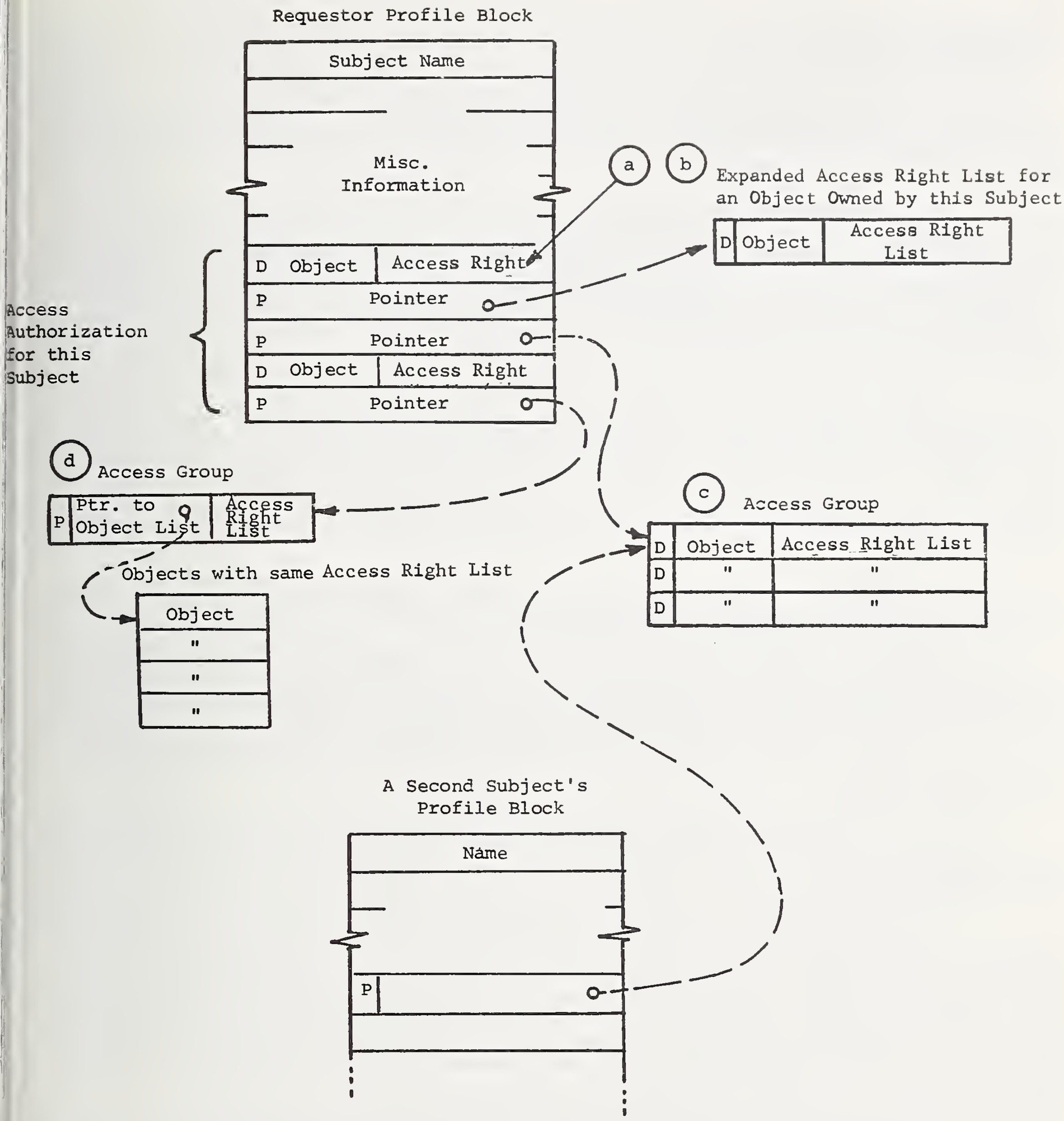


Figure 3-1. Linked-Block Structure

It was assumed that the storage required for each user is about 750 bytes. Depending on disc sector size and factors such as the number of object and subject groups referenced, a user's storage may occupy from two to six sectors.

In the example no attempt is made to utilize disc space optimally. Each sector will contain data for only a single user or a single object or subject group. By not packing data into disc sectors, the number of overflows from full sectors can be minimized. Having data for different users in a single sector appears to have no advantages, except to minimize total space used.

The subject of organization of storage to facilitate searching has been given extensive and continuing attention. Knuth [7,8] gives a very detailed treatment of the area. This section will only indicate some possible approaches and some considerations applicable to a detailed design.

It is assumed that the bulk of the operations will be search operations and that update operations will be required much less frequently. Once the beginning of the user-profile block (for authentication) has been located, it is desirable that any subsequent sectors required for that user be located without unnecessary disc accesses. If each entry (object-access right pair, or pointer) is a separate element of a linked list, rather than a sequential list, then the entries can be organized into any of several tree structures to reduce the length of the search.

Another possibility is for the initial sector to contain some sort of directory to the sectors that contain the data for the various objects. This only makes sense if the directory can be based on less than the full object identifier, since storage of the full identifier constitutes most of the entry (except for object groups which will be dealt with below). The directory might be based on host names, with all objects at a host (for each subject) grouped together in one segment. Another possibility is to use a hash function in the directory, in which case all objects that hash together would be stored in the same sector.

The additional complications of these structures may not be justified if object profiles are short, or if only a single additional disc sector is involved.

Regardless of the organization, or lack of it, object-groups must be treated somewhat differently. It will not generally be possible to determine, based strictly on object identifier and object-group identifier, if an object is a member of a particular group. Thus any object groups referenced in a user-profile block must be exhaustively searched. It will probably be wise to organize the search of the user-profile block to first search all direct object-access right pairs, then, if the object is not found, to search object groups.

As mentioned previously, the user-profile block for each user must be located for each access request. This is a much more substantial search than occurs within each user-profile block.

Since the baseline system assumed 1000 users, a sequential search of user-profile blocks is not feasible. Thus the user-profile blocks will need to be organized in some manner to permit more efficient searching. Use of directories or tree structures are two promising possibilities. Structures such as B-trees [8] are well suited to organizing data to permit efficient searching and yet retain update flexibility. Use of directories or tree structures may substantially increase storage requirements, possibly by as much as 25% but minimization of disc accesses, as discussed in Section 3.2.7, is a very important goal.

3.2.5.3 Operations. This section will indicate by selected examples how the operations discussed in Section 3.1 would translate into operations on the linked block structure.

The authentication module is a predicate which returns a true or false value indicating whether or not an offered authenticator correctly authenticates a subject. The specification to authenticate an offered pair <subject identifier, password> is:

```
If subject identifier is valid AND password is valid AND <subject identifier,
     password> is a member of AUTHENTICATION

          THEN  TRUE
          ELSE  FALSE
```

In the linked-block structure this would be accomplished as follows: Range checking, to determine if the offered subject identifier is valid and the offered authenticator is valid, is performed before determining if the pair is valid. Rather than looking these up in lists of identifiers or authenticators, the range checking is accomplished by processing on the form of the parameters (alphanumeric, proper length, etc.) rather than a precise check on the values. To check if the offered <subject identifier, authenticator> pair is in the approved set, the user-profile block for the subject would be located, then the stored value of the authenticator would be compared for equality with the offered value.

To check if an access request (for <Subject Identifier, Object, Access Right >) is authorized, the specification checks if <Subject Identifier, Object, Access Right, True> is a member of AUTHORIZATION (after range checking on parameters). In the linked-block structure, this would translate to: locate user-profile block for subject identifier, then search block for Object (may involve searching object groups). If Access Right code is present, then the requested access is permitted. If the object is not found, or the access right is not present, any subject groups the user is a member of must be similarly searched. If the object with the desired access right code set cannot be found, the request is rejected.

To modify an authenticator in the set model, the specification is: to change the authenticator for Subject Identifier from Old to New replace <Subject Identifier, Old> with <Subject Identifier, New> in the set AUTHENTICATION.

In the linked-block structure, this becomes: locate the user-profile block for Subject Identifier, and replace the old authenticator by the new one.

Modification of an access permission depends on whether it is to be added or removed. Removing an access permission involves locating the entry in the user-profile block and removing the <object, access right> entry. If the object is in an object group, a different procedure is necessary. In this case, the object group must be removed from the user-profile, and all the objects in the object-group except the one being changed, must be added to the user-profile block as individual object-access right pairs. If the access permission to the object is to be removed for all subjects with access to the object-group, then the object can be removed from the object group.

When an access right to an object is to be added, the user profile block is located. If the object is already in that user's profile block (not part of an object group), the new access right is added to that object's access right list. If the object is not present, the <object, access right> pair is added as a new entry in the profile block. The actual mechanism of this addition depends on how the user-profile block is organized internally, as discussed in 3.2.5.2. The block organization will govern how operations such as garbage collection and block overflow on updates will be handled.

Although this section has not provided a complete re-specification of all operations in terms of the linked block structure, the examples presented illustrate the type of operations required.

3.2.6 Distribution. The linked-block structure described in Section 3.2.5 is appropriate for a single unified NSC, but there must be some modifications made for the distributed multiple NSC case. In this case, the user-profile block contains only authentication information and complete hosts as objects. The entry for each host includes a direct reference by name (rather than an indirect reference by pointer) to its parent NSC. Since the object list for each subject is quite short, simple linear search will probably be most appropriate in this part of the structure.

The organization of the object data is a different matter. Of course, if a subject-oriented organization is chosen for this data, the user-profile organization of Section 2.3.5 will apply.

However, it is likely that this data may be organized by object. In this case, the structure of Section 3.2.5 would be inverted. The grouping would be by object-profile block An entry in this block would either be a direct <subject, access right> pair or a pointer to a subject group, analogous to the object groups of the previous example. Depending on the size of a typical object profile, organization to optimize searching may be required.

In the previous discussion of factoring, it was mentioned that use of subject groups could be dangerous in a subject-oriented organization. In the distributed organization this may no longer be a problem. If the data concerning each host is organized according to objects, then a subject-group identifier would appear there as a pseudo-subject. If the user-profile at the subject NSC lists all subject-groups that the user is a member of, these group names can be passed to the object's NSC with the access request. The search at the object NSC is then based on the subject identifier, or any of its subject-group names.

It is expected that operation in the distributed mode will require slightly longer service time, since part of the process involves message communication between distinct NSCs. In the baseline system, this additional delay will not result in a total request service time of more than one minute. If message communication adds .5 second to the processing of the request, this is less than 1% of the one-minute service time. Message communication delay will probably be a factor only during periods of very heavy utilization. This can be minimized when data is assigned to NSCs so that, based on expected access patterns, most access requests involve only one NSC (that is, the subject's NSC is the same one that handles the object's host).

3.2.7 Scaling the Implementation. An NSC implemented in a current generation minicomputer can handle the hypothetical baseline network. There is a question about how the implementation scales to larger networks. To answer this, we will analyze the limiting factors on the size of the user population the NSC will handle.

The tutorial [1] presented some simple analysis of the I/O load on an NSC. It will be summarized here as a basis for the discussion on scaling.

To estimate the expected number of subjects in dialogue with the NSC at any one time, use the equation

$$\overline{n} = N(T_s/T_a)$$

where N is the number of active users

T_s is the average service time

T_a is the average time between service requests

for $N = 1000$, T_s = one minute, T_a = 20 minutes

$$\overline{n} = 50$$

We assume that the I/O service time for one disc access is about 100 milliseconds. This includes seek time, rotational latency, transfer time and I/O software overhead. I/O transfer time and processing is assumed to be small relative to I/O access time.

If we assume each access request requires four disc accesses over the one minute service time, each access request generates one disc access per 15 seconds, on the average.

For 50 active requestors, the expected time between I/O requests is 15 seconds ÷ 50 = 0.3 seconds.

For a service time of 0.1 seconds the queueing "traffic intensity" is 0.1/0.3 = 0.33. This leads to

Expected number waiting for (or in) I/O service = 1/(1-0.33) = 1.5
Expected time for I/O service plus queueing delay 0.1/(1-0.33) = 0.15 seconds
For the baseline system, I/O queueing delay is not excessive.

There are several candidates for limiting factors on the size network an NSC may serve. Disc capacity is dismissed as a factor; even though the number of users may increase by two orders of magnitude, disc capacity is available to handle the increase of data base size. The NSC can support a much larger total network population if the size of the active population at any one time remains the same as in the baseline system.

A limitation on NSC capacity is found when I/O service times are examined. Assuming an I/O service time of .1 second and 4 disc accesses per request, when there are 150 simultaneous requests the I/O arrival rate will equal the I/O service rate; this is an absolute upper bound. According to the simplified analytic model, I/O queueing delays become substantial as the number of active requests approaches this limit. One hundred fifty simultaneous requests corresponds to 3000 active users in the network. Reducing the I/O service time in half to 0.05 seconds increases the supportable network population to 6000. Thus a single NSC cannot support even one order of magnitude increase in active network population. Of course, any change in the assumptions about user behavior, such as average service time or average time between service requests, will also affect this result.

The availability of main storage within the NSC for processing active requests may also limit the number of simultaneous requests that may be handled. Due to the process-oriented structure for handling separate requests, as discussed in Section 5, it appears necessary to allocate sufficient space to each request to accommodate a complete disc sector as well as other variable data needed to process each request. For a sector size of 256 bytes, we estimate that about 15K words of storage is required for 100 active requests and about 23K words for 150. For a sector size of 512 bytes, 100 requests require about 27.5K words and 150 requests require about 40K words. The baseline system can be handled with about 8K words for 256 byte sectors and 14K words for 512 byte sectors.

Thus, memory capacity for a large mini is consistent with the limit imposed by I/O service rates, and is well below one order of magnitude increase over the baseline system.

Another restricting factor is the number of simultaneous connections that the NSC may sustain with remote users. Depending on the I/O facilities available on the NSC and the logical connection capabilities of the network, the number of simultaneous dialogues that can be accommodated may be severely limited. For example, an NSC that is acting as a "front end" for a group of terminals attached to a single host will be limited by the number of I/O ports for terminal connections. The limit in such a situation may well be in tens of lines, and not approach the internal NSC processing limit of 100 to 150 dialogues.

It is difficult to make any firm statement on minimum feasible size for an NSC. If the network is to employ Network Cryptographic Devices, one key distribution scheme makes use of minicomputer-controlled master NCD. If so, even a very small NSC could be justified. Even if a separate processor must be used, minis are available in very low-cost models. For a network with a very small number of subjects (or a grouping into a small number of subject-groups) and very few objects (such as only complete host systems), it may be feasible to implement NSC data structures entirely in main storage.

In summary, due to I/O service rates and buffer size limitations, expansion much beyond the baseline network will require multiple NSCs. Each of these NSCs will be relatively small, since neither disc nor main storage requirements are extended beyond reasonable limits. The large network will have many small, cooperating NSCs rather than concentrating all NSC responsibility in one large component. This approach provides many benefits, such as modular expandability, low cost components, decentralization of control, and fail-soft operation. Fail-soft operation means that loss of one NSC does not affect the others, or block access for all subjects to all objects. Only the failed NSC's domain of subjects and objects are affected, and only for new connections. Established connections and subject-object combinations in other domains are not affected.

4.0 I/O STRUCTURE

Section 3 presented a model for the NSC data structures and an implementation example. This section will describe the I/O structure necessary to support the NSC data structures.

4.1 Hardware Base

The hardware base assumed for the NSC is current generation minicomputer. Disc storage for minis is available in fixed head and moving head types. The fixed head disc has a capacity of less than 1M byte with an access time of about 10 milliseconds. The moving head discs are available in three size ranges, from 2M byte cartridges to 200M byte packs. Access times range from 75 milliseconds to 40 milliseconds, while transfer times range from 11 microseconds per word to 2.5 microseconds per word.

Disc storage is organized into fixed length, sectored configuration, with the usual sector size of 256 or 512 bytes.

I/O transfers for disc storage are generally handled by Direct Memory Access (DMA), which performs the transfer of a sector (or consecutive sectors) without processor intervention. The processor is interrupted when the transfer is complete.

Minis with main memory configuration larger than 32K words generally use some form of memory mapping to allow accessing extended memory. The type of mapping may vary from a single base register to multiple segment mapping registers, with access control associated with each segment. Although processor memory accesses are mapped, some minis do not map the DMA I/O transfers.

Costs for minicomputers are changing rapidly. Currently, the cost of a 2M byte cartridge is about the same as a mini with about 32K words of main memory. The cost of a 40M byte disc is about 1.5 times that of the cartridge and the cost of the large disc is about three times. The cost of a fixed head disc is about equivalent to that of the large disc.

As the cost of memory and processors continue to drop, the cost of peripherals, such as discs, remains fairly constant (ignoring general inflationary economic trends). Thus, the relative cost of disc storage is increasing. The advent of solid state mass memory devices will undoubtedly change this trend, but we will not attempt to forecast those changes here.

4.2 Size Requirements

The size data base that the baseline NSC must support is .78M bytes (see Section 3.2.3). Although this would fit on a fixed head disc initially, there is little margin for expansion. Thus we will consider only moving head discs for NSC data storage.

For the initial baseline system, the 2M byte cartridge provides ample storage space, with enough room for expansion. As mentioned in Section 3.2.7, one NSC cannot support an order of magnitude increase in the number of active users. However, the total user population (active and non-active) may be increased by as much as two orders of magnitude and still be within range of available disc capacity. If a distributed NSC is adopted to accommodate a larger active population, the total storage requirements will increase by as much as 25%. The storage will be distributed over several NSCs, and the requirements for each will still be within reasonable bounds.

4.3 Response Requirements

Section 3.2.7 determined that I/O response time is the limiting factor in the number of simultaneous requests that may be processed by the NSC. Access times (average seek time plus average rotational latency) range from about 75 milliseconds for the cartridge to about 40 milliseconds for the large discs. Transfer time is about 11 microseconds per word for the cartridge and about 2.5 microseconds per word for the large disc. Transfer time is less than 4% of the access time for the cartridge and about 1% of the access time for the large disc.

To take account of software I/O delay and processing that is done on each disc sector, 100 milliseconds per disc access were allowed for the estimates in Section 3.2.7. This allows from 20 to 50 milliseconds for processing. At minicomputer execution speeds (assuming about 4 microseconds per instruction) this permits execution of about 5,000 to 12,500 instructions. This is certainly a generous allowance for any necessary processing.

Although fixed head discs have a much better access time (about 10 milliseconds), the storage capacity they provide is limited. In addition, they are much more costly than moving head discs of similar capacity.

4.4 Software Structure

The recommended I/O software structure is determined not only by the requirements of the data structure but also by the desired control structure, as will be discussed in detail in Section 5. Briefly, Section 5 will recommend that the processing of each request be handled independently by a separately instantiated process. These processes will operate in entirely separate address spaces. There will be no processing of data in a common area accessible to several processes.

Following the implementation example of Section 3.2, the processing of a request may be broken down into the following major steps:

. locate user-profile block,

. perform authentication,

. locate object-access right pair in profile block, and

. notify subject and object and create cryptographic connection.

The first action to be performed is to locate the user-profile block, which isolates the portion of the total data base relevant to a single request. The logical I/O software will be aware of the user-profile structure, and an I/O request from a process will specify the subject identifier for the desired profile-block. The I/O software will locate and read into the separate process address space the appropriate profile block.

In the course of processing a request, the process will require additional I/O service to read subsequent portions of the user-profile, such as object groups. Due to space restrictions, it was assumed that a request process would contain buffers for only one sector at a time. Requests for subsequent portions of a profile block will be in terms of disc sector addresses, which are entered as pointers in the profile block.

The logical I/O structure will accept initial I/O requests in terms of subject identifiers and subsequent I/O requests in terms of disc addresses. The unit of I/O transfer is one sector, and each sector is read into the separate address space of the requesting process.

Updates to the data base are handled by a request to a single update process, which can issue appropriate locks on portions of the data before updating.

4.5 Memory Mapping and Encryption

The control structure discussed in Section 5 specifies a separate independent process to handle each request. It is recommended that the hardware base that is used be capable of incorporating memory mapping to permit each process to execute in its own isolated address space. Although not strictly required to implement an NSC, memory mapping simplifies the maintenance of separate processes, eases implementation of reentrant processes and provides some degree of protection against accidental out of range references. When access control is associated with memory segments as part of the memory mapping mechanism, it is desirable that the memory mapping, and thus the access control, be active for I/O references as well as for processor references.

The logical I/O structure has two different types of requests to process. The first type is for locating a user-profile block, which involves reading data into an I/O system area until the desired sector is located, then reading the user-profile itself into the address space of the appropriate request process. The second type of request is for additional sectors which are part of the profile block being searched. For these requests, the data will be read directly into the address space of the request process.

Encryption has potential for protecting data on secondary storage media. The simplest way to use encryption is to encipher all data written to the disc and decipher all data that is read, using a single fixed key. This would protect the data from theft of the media to itself. Although with proposed LSI implementations of the NBS Data Encryption Algorithm, the disc transfer rate would be slowed for the large discs, transfer time was a minor factor in I/O service time.

Another possible way to use encryption with secondary storage would be to encipher different portions of the data with different keys. For example, each user-profile block might be enciphered with a distinct key. This raises two problems. First, shared data, such as object groups, could not be enciphered in one user's key, if other users are to have access to it. Thus each shared entry in a profile block would need to be enciphered with yet another key. The second problem is more substantial. This plethora of distinct keys must be stored, selected, invoked, and purged at the proper times. If the keys are managed by the processor and stored on the medium itself, there is no protection against a subverted machine. Depending on where in the logical structure an error occurred, it might not even provide much protection against accidental software or hardware errors.

The entire area of secure key management to provide data separation requires more extensive study and evaluation. Simple single key encryption of all data to protect against theft of the storage media is the approach recommended at this time, if encryption protection of the NSC is desired. However, since the NSC is a separate, trusted component which can be physically secured, the benefits of storage encryption may be marginal.

5.0 NSC CONTROL STRUCTURE

This section will present recommendations for the internal control structure of the Network Security Center (NSC). A wide range of implementation strategies might be used in construction of an NSC, each satisfying different objectives.

In this section we will present a structure which addresses the following goals:

- High degree of modularity

- Controlled inter-module interaction

- Flexible accommodation of user differences

- Easy expansion to multiple NSC architecture

Modularity is an important and well-recognized practice for producing well-engineered and reliable software. Modularity has the effect of segmenting both design and implementation into a set of manageable-size efforts. Modularity aids in the understanding of the functions, their implementation and their relationships.

The goal of controlled inter-module interaction is necessary to insure that each module performs its intended function and only its intended function. By controlling the manner in which modules interact and communicate, we can be assured that each module performs its proper function when properly invoked. We can also be assured that if a module is improperly invoked (e.g., with bad parameters), the module itself can refuse to function and can return an error condition. By strictly controlling module interaction, all invocation paths can be identified and thus controlled. Without controlled interaction it will be difficult to have confidence in the proper functioning of the collection of modules.

Since the NSC operates in a computer network, it must be able to accommodate both human and process users. Human users require much more accommodation in terms of prompting, error recovery, flexible formats, etc. For example, an experienced human user will probably prefer to specify all the parameters for both authentication and access request authorization as part of a single interaction with the NSC. A naive, inexperienced, or occasional user will probably prefer an extended dialogue, with the NSC prompting separately for each required parameter.

Previous sections have identified several reasons why a multiple NSC approach may be desired: an important one is the capacity of a single NSC to service simultaneous requests. Thus it is desirable to have an architecture that permits moving from a single NSC to multiple NSCs as a network grows. This transition should be accomplished with a minimum of reprogramming and no additional software design effort.

5.1 Process Structure

The recommended structure which accomplishes the goals set forth previously is based on a collection of independent processes communicating by exchange of explicit messages. The NSC structure will enforce separation and isolation of the processes. By limiting inter-process communication to only explicit messages, each process can be implemented to recognize and respond only to valid messages. Individual processes are thus isolated and cannot affect or be affected by other processes, except in the manner intended by the design of the NSC.

Each process can be certified to implement its portion of the NSC function correctly, independently of other processes. When each process is so verified, the task of certifying the entire NSC functionality is greatly simplified.

To simplify the implementation and insure separation of processing (and data) for each user request, there will be a separate and distinct instance of a request handling process initiated to handle each user dialogue. These multiple instances of the request process will be identical, each operating independently in its own address space.

There will also be a number of support processes, of which only a single instance of each will exist.

Re-entrant code is the appropriate technique for implementation of the multiple instance processes. Not only does re-entrant code greatly reduce space requirements for multiple instances of the same process, but the required separation of pure procedure and data provides a measure of protection against inadvertent errors causing program changes. Since the data for each request is separate, errors which do affect data will be limited to one request. The protection is not absolute and guaranteed, since the mechanisms that support and manage the separation may fail, but the range of problems that can cause total failure is substantially reduced.

The use of re-entrant code, and the desired separation of address spaces indicates a preference for a machine that has appropriate hardware support features. Memory mapping, definable memory segments, and segment granularity memory protection (as opposed to large fixed memory block protection) all serve to aid in implementation of a structure as described.

Hardware implementation of separate address space for each process can insure that explicit message communication is the only way for processes to interact. Although not all minicomputers have these features available, this suggested implementation indicates a definite preference for those that do.

The structure described meets the goals of modularity and controlled interaction, at the price of increased overhead. The overhead need not be as great as for a general-purpose process-oriented system. In the NSC there is a fixed set of processes; there are no user programs to be accommodated. Thus, at design time the size and behavior of all processes are fairly well known.

Although flexible inter-process communication is required, the communication behavior of each process is known; problems such as buffer allocation, flow control, message addressing and delivery are simpler than in the general case.

The modules of the NSC fall into three main groups. First is the request process group, consisting of the modules that perform the dialogue with the user, the authentication module and the access request authorization module. These modules have a separate instantiation for each active request being processed. The second group is the support process group, consisting of modules for secondary storage I/O, communication line management, cryptographic key generation, auditing, and data base update. The third group is called the nucleus, and consists of modules that implement inter-process communication, I/O drivers, internal scheduling and memory management, and cryptographic key distribution. Each group will be discussed in detail in subsequent sections.

5.2 Request Process

The request process group of modules is responsible for user dialogue, user authentication, and access request authorization. In order to maintain separation of the processing of each individual request, a distinct instance of the request process modules will be initiated for each user dialogue. The structure of the modules can be kept quite simple, since a module only processes a request for a single user at a time.

The NSC operates in a computer network and must expect requests from both human and process users. The experience and sophistication of human users may vary over a wide range. To provide flexible accommodation of these diverse users, the request process group will consist of two separate processes. One of these, the user dialogue process, will actually communicate with a user, providing prompting when necessary, providing input editing if necessary, accommodating free format input, and accepting request parameters in a single command or in a series of separate interactions.

The user dialogue process will translate the varied formats into a concise standard format which is expected by the actual request process. The user dialogue process will communicate with the request process via inter-process messages, which are in the standard, concise format. At this simple level, the translation is merely from flexible, free form text messages into concise coded form. In a more heterogeneous situation, there may be more than one type of user dialogue process to accommodate different user dialogue forms. The type of translation required is much less complex than provided by systems such as the NBS Network Access Machine (NAM) [11,12]. The NAM provides for user defined commands that are mapped into perhaps several differing command formats. User dialogue processes translate predefined command strings into a single fixed target format.

The request process accepts only the concise, standard format messages. Processes which act on behalf of users communicate directly with the request process, rather than through the user dialogue module. Terminals that are connected to the network through a secure network front end or a secure terminal controller can have an equivalent of the user dialogue process resident at the terminal interface. This remote user dialogue process communicates directly with the request process, using the concise standard format.

The user dialogue process provides easy flexibility in handling different terminal types. As an example, the method of concealing a typed password differs from terminal to terminal. The user dialogue process can be aware of these differences, or a different user dialog process can be invoked for each different terminal type.

The request process itself consists of an authentication module and an authorization module. The function of these modules has been specified in Section 3.1. The example implementation discussed in Section 3.2 indicates that there are two types of operations involved in processing the authentication request and the authorization request. The first is locating the user profile, which is performed by the system. The second is searching through the profile, which is done by the request process itself. Thus the request process must be able to issue both types of I/O requests (locate profile, read sector) and have sufficient buffer space to hold the I/O data for processing within its own address space. To insure separation, all processing of data will be done within the process' own separate address space, not in an allocated common-pool buffer.

When authentication is complete and the access request is authorized, the request process obtains a distinct cryptographic key for the new connection, and causes that key to be transmitted to the cryptographic devices at the subject and object.

When the request process has completed user authentication, access request authorization, and cryptographic keying, or when the processing of a request is terminated for any reason, an audit record will be transmitted via inter-process message to the audit process. This audit process is responsible for recording and analyzing the audit data from the various request processes.

The process structure for a multiple NSC architecture is only slightly different. In Section 3, the distribution of the access control information in a multi-NSC system was described. The access authorization information is separated into host granularity objects (grouped by subject) and resource granularity objects (grouped by object).

The authentication information and the host granularity access authorization information reside in the NSC responsible for the subject. The finer granularity access information is stored at the NSC responsible for the object.

The impact on the process structure presented here is minimal. In this case, the request process functions are divided into a subject request process and an object request process. The subject request process contains the authentication module and host granularity authorization module. Communication between the subject request process and the object request process is via inter-process messages. The inter-process message facility of the nucleus distinguishes between messages addressed to a local process and those addressed to a process in another NSC. The nucleus is the only part of the system which is aware of the different location of the recipient, and it handles the differences in actual communication in a transparent manner.

Even if a request is processed entirely within a single NSC, the subject request process will communicate with the object request process via messages. The message mechanism is identical (from the process point of view) for both intra- and inter-NSC messages. The only awareness that a subject request process has of the location of the object request module is by its name. This allows easy migration of data from a single NSC to several. The request processes do not have to be aware of the move. Only the module that implements the message communication function must be changed.

The process structure presented here allows easy conversion to the multiple NSC architecture. If a single NSC is to be upgraded, only the request process must be changed, and inter-NSC communication added to the nucleus. If it is anticipated that a move will be made to a multi-NSC at some later time, the initial NSC can be built with the dual request process structure and simply maintain all modules within the same NSC.

5.3 Support Processes

The support process group provides various services for the multiple request processes. Functions which require a single point of control within the NSC are in this group, and there is a single instance of each of these processes.

The functions performed by this group include secondary storage I/O, communication management, cryptographic key generation, data base update, and auditing.

The communication manager process is the single control point in the NSC. The I/O line drivers in the nucleus communicate with the communication manager except when actually involved in a request dialogue. When a user first initiates a dialogue, the communication manager recognizes the start of a new dialogue and initiates a new instance of a user dialogue process and a request process. The logical I/O connection is assigned to the user dialogue process and all subsequent I/O traffic is with that process. When a request has completed processing, the user dialogue process and request process terminate, and the now idle I/O connection is reassigned to the communication manager. The communication manager may also recognize a special command prefix that will indicate that a dialogue is being initiated, since a process user will be communicating directly in the concise standard form expected by the request process.

As described in previous sections, there are two distinct types of I/O operations required to support a request process: user profile I/O and sector I/O.

The user profile I/O process will accept a subject identifier as a parameter and will perform the search of the data base to locate the first sector of the user profile for the specified user. When that sector is located, it is written into the address space of the appropriate request process. Thus, the user profile I/O process must have sufficient buffer space to read intermediate parts of the data structure while locating the desired profile, and must be aware not only of the logical structure of the data base but also how that logical structure is mapped onto secondary storage.

The sector I/O process is used to perform actual sector read operations. Both the user profile I/O process and the request processes will make use of the sector I/O process to search the data structure. The sector I/O process services the interface between the logical I/O requests generated by other processes and the specific details of the I/O device driver calls as provided by the nucleus (see Figure 5-1).

The cryptographic key process is responsible for generating distinct cryptographic keys for distribution to the Intelligent Cryptographic Devices (ICDs) at the subject and object. The cryptographic key process may have a pseudo or actual random process (hardware or software) available as part of the key generation mechanism.

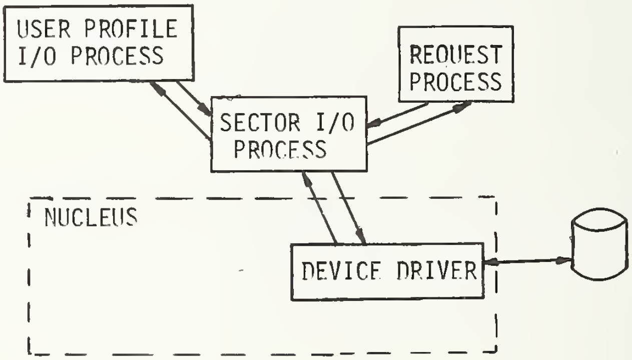


FIGURE 5-1. I/O Processing

In some systems, the actual key may not be distributed, but a key selector is used instead. In the distributed approach to cryptographic keying discussed in Section 2.2.4, a connection identifier is generated, rather than a cryptographic key.

The update request manager process is the single point through which all updates are controlled. This process must perform authorization checks before performing an update, and is responsible for issuing appropriate locks and controls on various portions of the data base to insure that updates do not leave the data base in an inconsistent state, and that interrogation requests are not allowed to access temporarily inconsistent portions of the data.

The audit process is the central repository for audit data recorded by the request processes. Whenever a request process takes any action (such as approving or denying an access request), a record of that action is sent to the audit process. The audit process will record, on secondary storage, a journal of all audit messages. It will also interpret and select certain information for analysis. For example, the audit process will keep a record of authentication failures and access request denials over a reasonable time interval. The information will be organized on the basis of user identifier, terminal identifier, object, etc. This will enable the audit process to determine if simple penetration attempts are in progress. The real-time audit can be as complex and sophisticated as desired, within the limits of program space and execution overhead.

5.4 Nucleus

The nucleus in the NSC provides the low level support necessary to implement the process structure. Such machine dependent functions as memory management and allocation, process scheduling, interrupt handling, and I/O drivers are performed in the nucleus.

The nucleus implements the inter-process message communication facility. In a multiple NSC network, the inter-process communication facility must distinguish between messages addressed to local processes and messages addressed to processes in other NSCs. Non-local messages must be transmitted through the communication system to other NSCs, where the nucleus inter-process communication facility will deliver the message to the appropriate process. The nucleus will mask all differences between inter- and intra-NSC messages from the point of view of the processes.

The nucleus provides I/O drivers for the peripherals attached to the NSC. Secondary storage, communication lines and cryptographic devices must all be supported by the nucleus. If the NSC site is to distribute cryptographic keys to all remote NCDs, the nucleus must contain drivers that interface with a special Network Cryptographic Device (NCD) of the NSC. In addition to normal cryptographic protection of logical I/O connections, the special NCD must distribute cryptographic keys to NCDs at other sites. These keys are distributed on command from a request process in the NSC when an access request has been approved. If the distributed approach to cryptographic keying is adopted, the NSC must be able to instruct the remote NCDs that a connection is authorized and that they may establish a key.

The nucleus will also provide a logical I/O structure for the communication system logical connections. A logical connection will be allocated or assigned to the appropriate user dialogue process or request process servicing the request coming in on that connection. All logical connections not allocated to an active request process will be automatically allocated to the communication manager process.

The nucleus is not to be confused with the concept of a security kernel [9,10]. A security kernel isolates all security relevant code in a system into a kernel that interprets and enforces all security-relevant actions. The entire function of the NSC is security-relevant, thus the concept of a security kernel isolating all security-relevant code does not apply.

5.5 Error Control

The process structure presented has a great deal of error control inherent in the design. The use of separate processes to service each user, separate process address spaces, separate code and data segments, hardware address protection enforcement, and a centralized update process, all contribute to error control. Inadvertent errors are isolated and restricted in scope.

Other techniques are available to provide a greater degree of error control. Redundancy checks (such as a cyclic redundancy check or checksum) can be included as part of all inter-process messages. The secondary storage hardware can incorporate similar checks to detect transmission errors. Stored data can include redundancy checks or simple parity checks to detect inadvertent modification of the data.

More sophisticated techniques are possible, but the cost is greater. For example, each request could be processed by two separate request processes, and the results compared. (Of course, the reliability of the comparator is always an issue, but it is presumably much less complex than the request process, and thus there is more confidence that it does not fail.) However, this only guards against dynamic data errors. Since each instantiation of a request process uses the same re-entrant code, running dual request processes does not guard against program errors. If a transient error changes the program code, all instances of the process are affected. Only request processes running in separate NSCs would get around that problem.

A less effective, but less costly approach will probably suffice in all but the most demanding applications. Redundancy checks can be incorporated into the object code for each module or process. Since re-entrant code is used, there will be no variable values to disrupt such an approach. At selected intervals, an audit process can check the object code, as it exists in main memory, to determine that no changes have occurred.

These techniques are appropriate for controlling inadvertent errors. Errors introduced by a sophisticated penetrator or by a Trojan Horse built into the system cannot generally be detected by these methods. We assume that sufficient physical protection of the NSC site will preclude physical attacks on the NSC hardware, and proper procedural and personnel controls during software production and maintenance will preclude the introduction of Trojan Horses.

5.6 An Alternative

Although it is felt the process structure described here has many advantages, such as modularity, understandability, cleanness of design, and error limitation, it is realized that in some circumstances the overhead will be unacceptable. An alternative structure in which all requests are processed by a single request module will be briefly described.

A good control structure for handling many simultaneous multiple requests is a table-driven finite state machine. As each new request enters, it is assigned a new table entry. As processing of the request proceeds, the current state of the request (for example, pending user-profile I/O) is stored, along with the next action to be performed.

When a pending condition completes, the appropriate next action is initiated, the new state recorded, and the next action specified. Each "new action" will consist of a test of the result of the previous action, some processing, and specification of a next action (which is probably a function of the result of the previous action). When the request is completely processed, the entire table entry is cleared.

A very simple description of the request module is that it is simply a finite state machine, accepting input conditions, performing some action, determining which state transition to execute, and entering a new state. The request module must have available to it the same range of service modules described for the process structure. Although this alternative approach is certainly feasible, and does maintain some degree of simplicity and understandability, it does not have the advantages of the process structure presented previously. This approach does not separate the mechanism that processes a request from the mechanism that maintains isolation of separate requests. The finite state machine implementation is suggested as a feasible way of maintaining some degree of control and clarity of design when a single module must process simultaneous multiple requests. The state representation of an iterative process is often a less obvious and straightforward way of describing the processing flow for a single request. The advantages of small, simple, independent modules are lost, and thus problems of implementation, testing, modification, and certification are increased.

6.0 MESSAGE CONTENT AND FLOW

This section defines the content of various messages involved in operation of the process-oriented NSC. These message definitions specify generic message content and message relationships. It would be inappropriate to specify the messages in terms of field ordering, exact field sizes, or message lengths. The message definitions are not exact "bit patterns," but will reflect message semantics.

6.1 User Dialogue Messages

The messages between a user and the user dialogue process are the most free form and flexible of the messages in the NSC system. The messages from the user to the user dialogue process will be free format character text. Imbedded blanks between parameters will be ignored, and delimiters that are recognized by the user dialogue process will be flexible (usually a comma and/or blank at a minimum). The various parameters will be expected in the order specified, but may be sent in separate messages, waiting for an appropriate prompt each time, or they may be grouped together in a single message, or in any combination desired by the user. The user dialogue module will determine what prompting is necessary, based on which parameters were received so far.

```
USER                                              USER DIALOG PROCESS

Attention Signal →

                                                  ← Prompt for USER ID

USER ID →

                                                  ← Prompt for AUTHENTICATOR

AUTHENTICATOR →

                                                  ← Prompt for ACCESS REQUEST

ACCESS REQUEST →

     HOST NAME   }  may be a single
     OBJECT NAME }  structured name

     ACCESS TYPE
```

FIGURE 6-1. User Dialogue Messages

The suppression of printing an authenticator can be accomplished only when the user dialogue process can prompt for the authenticator. If the user groups the authenticator with the user identifier, the user dialogue module should attempt to overprint the authenticator, if that is possible on the type of terminal in use.

6.2 Request Process Messages

When the user dialogue process has collected the information as specified in Section 6.1, the process will format an input message to the request process. Process users in the network can utilize these formats directly, using a special control prefix to indicate to the communication manager that no user dialogue process is needed. Note that in these and all subsequent message formats a check character (denoted CC) is specified. This is a redundancy check such as a cyclic redundancy check or checksum, to detect transmission errors. Human users cannot be expected to generate or respond to check characters, so user dialogue messages do not contain them. All messages except user dialogue messages are fixed length, the fields being characters or codes.

USER DIALOG PROCESS (or User Process) sent to REQUEST PROCESS

PROCESS ID	USER ID	AUTHENTICATOR	ACCESS REQUEST	CC

FIGURE 6-2. Request Process Messages

6.3 I/O Requests

I/O requests come in two types, the user profile request and sector request.

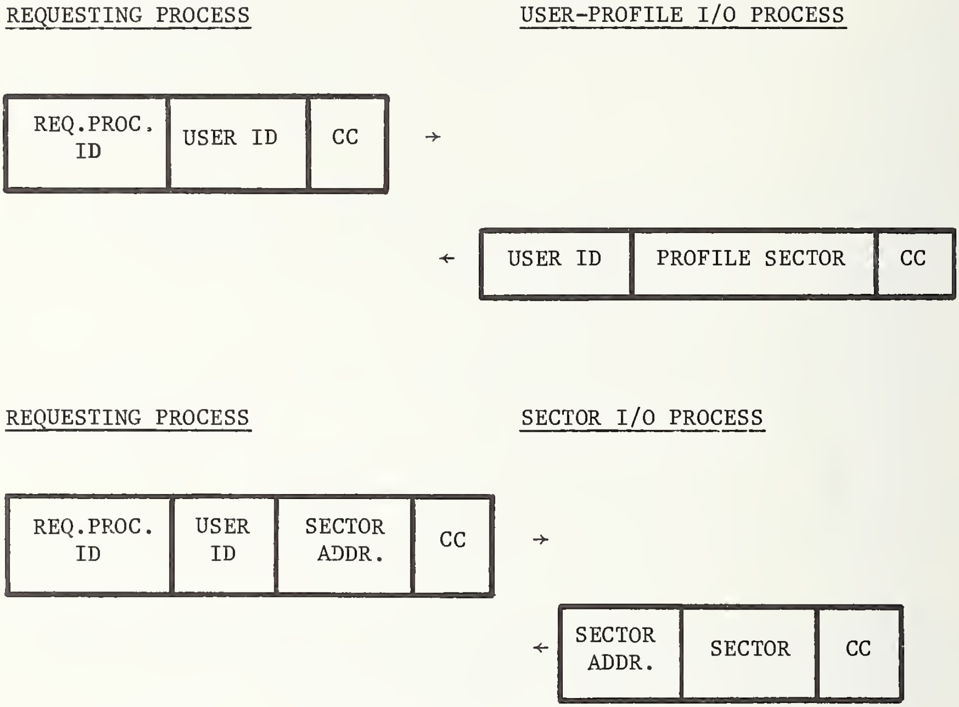


FIGURE 6-3. I/O Request Messages

Note that in both cases, the user identifier is included in the I/O request. It may be desirable in certain circumstances to journal (probably on a dedicated mag tape) all I/O requests, indicating on which user's behalf a sector was accessed. This is a substantial burden for normal NSC operation, but the journal facility may be switched on to provide a trace when a full audit is desired to locate a suspected problem. It costs little to provide the necessary information as part of the I/O request, if such a facility is desired.

6.4 Cryptographic Key Request

When a request process determines that an access request is authorized, it will obtain from a key generation process a cryptographic key to distribute to the Network Cryptographic Devices (NCD).

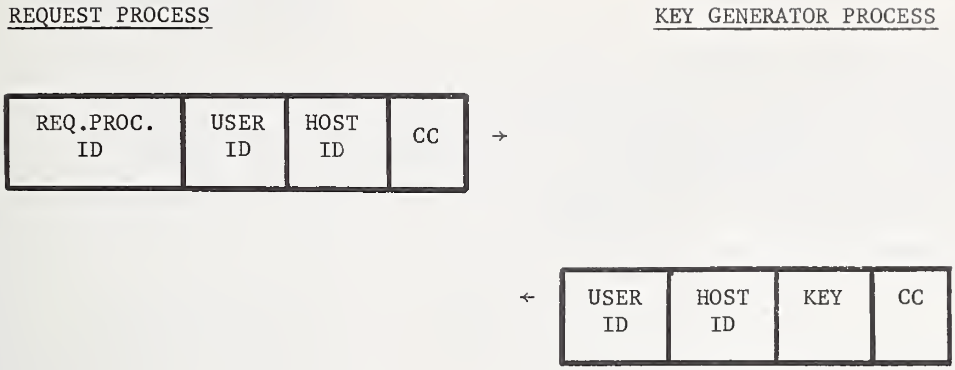


FIGURE 6-4. Cryptographic Key Request Messages

Note that the request and response both include user identifier and object host identifier. This is to provide a way to check responses, and also to permit a journal record, as discussed in Section 6.3, to be provided, if desired.

6.5 Cryptographic Keying

When a centralized key distribution approach is adopted, the cryptographic keying function is accomplished by passing the parameters (subject location, object location, key) to the NCD attached to the NSC. The nucleus performs the interface to the NCD, and is invoked by a call rather than by an inter-process message, but the format is the same.

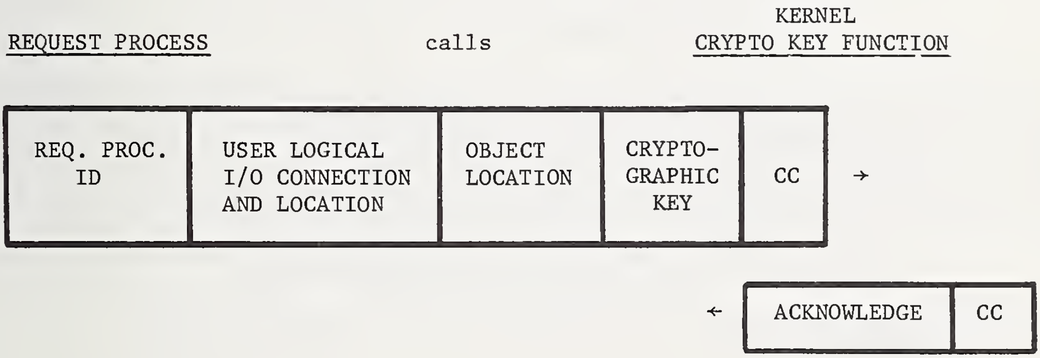


FIGURE 6-5. Cryptographic Keying Nucleus Call

6.6 Inter-NSC Messages

The inter-NSC messages are exactly the same as inter-process messages within a single NSC. The nucleus is responsible for providing this transparency. In multi-NSC operation the user messages are the same, but the function of the request process is distributed. The subject request process checks authentication and host granularity access authorization. The object request process performs finer granularity access authorization. The only additional complication in a multi-NSC situation is the intermediate message between the subject request process and the object request process. Other messages remain the same.

SUBJECT REQUEST PROCESS   OBJECT REQUEST PROCESS

REQ. PROC. ID	USER DIALOG PROC.ID	REMOTE NSC NAME	USER ID	ACCESS REQUEST (INCLUDES HOST NAME)	USER GROUP NAMES	CC	→

←	ACKNOWLEDGE	CC

FIGURE 6-6. Inter-NSC Messages

NOTE: User group names, which are essentially aliases for the user identifier, are discussed in Section 3.2. Their inclusion in the inter-NSC message is described in Section 3.2.6. The user dialogue process identifier is required for dialogue abort notification, as discussed in Section 6.8.

6.7 Audit Messages

The audit process will collect various information as transmitted to it by the other processes. The audit process will time stamp all information before recording it. In all cases the audit process will respond with an acknowledgement, which will not be shown in the following description.

from USER DIALOG PROCESS

PROC. ID	USER ID	"DIALOG STARTED"	CC	→

FIGURE 6-7. Audit Message-User Dialogue Process

NOTE: "Dialog Started" is a concise code rather than a character string.

from REQUEST PROCESS

PROC. ID	USER ID	AUTHENTICATED YES OR NO	ACCESS REQUEST (INCLUDES HOST NAME)	DISPOSITION	CC	→

FIGURE 6-8. Audit Message-Request Process

NOTE: The actual authenticator is not recorded. If the authenticator does not check, the dialogue may have been terminated before the access request was gathered. In this case, a null code will be present for access request.

from SUBJECT REQUEST PROCESS

PROC. ID	REMOTE NSC NAME	USER ID	AUTHENTICATED YES OR NO	ACCESS REQUEST (INCLUDES HOST NAME)	USER GROUP NAMES	CC	→

FIGURE 6-9. Audit Message-Subject Request Process

NOTE: This is simply the message to the OBJECT REQUEST PROCESS, with the addition of the disposition of authentication. As in the single request process audit message, if the authentication check fails, the access request field (and remote NSC name) may contain a special null code to indicate that no access request was collected prior to ending the dialog.

from OBJECT REQUEST PROCESS

PROC. ID	USER ID	SUBJECT NSC NAME	ACCESS REQUEST	DISPOSITION	CC	→

FIGURE 6-10. Audit Message-Object Request Process

6.8 Dialogue Termination

Dialogues that establish a properly authorized connection require no additional messages; the connection is simply created. However, when a dialogue must be aborted, due to failed authentication or denied access request authorization, the user must be informed. Whichever process (single request process, subject request process, or object request process) aborts, the dialogue will inform the user dialogue process which will, in turn, format a user oriented response to the user. When there is no user dialogue process, the coded response will be returned to the user process.

from ABORTING PROCESS

USER DIALOG PROCESS OR USER PROCESS

USER ID	ABORT CODE	CC	→

from USER DIALOG PROCESS

USER

Text String Error Message →

FIGURE 6-11. Dialogue Termination Messages

7.0 SUMMARY AND CONCLUSION

The Network Security Center (NSC) is a separate, secure network component which performs the functions of identification/authentication of subjects and access request authorization for subjects requesting access to objects. The NSC acts in close cooperation with intelligent Network Cryptographic Devices using the NBS Data Encryption Standard to enforce access control and provide end-to-end cryptographic protection for message transmission and mutual authentication of the communicating entities. The NSC is designed for implementation on a current-generation minicomputer.

The NSC can take on several differing roles in network security. The NSC performs authentication of the identity of subjects and provides authorization for subjects' access requests to objects. By controlling creation of cryptographic connections, the NSC can enforce access control to the granularity of complete host computer systems. The NSC can provide interpretation of access requests to finer granularity, with enforcement carried out by the hosts.

Implementation of NSC data structures was considered and it was shown that a minicomputer implementation of an NSC is appropriate for a wide range of network sizes, with I/O service being the most critical limiting factor for a single NSC. The design is oriented to use multiple NSCs, if required, so there is adequate growth potential for very large networks.

The recommended implementation focuses on three major concerns: (1) the ability for the NSC to interface to both human and computerized requests for network connections, (2) a modular structure made up of simple components, and (3) the ability to separate processes into different physical machine environments, permitting modular expansion to multiple NSCs and ensuring proper interprocess interaction since all interprocess communication is explicit.

The process structure presented has a great deal of error control inherent in the design. The use of separate processes for each user request, separate address spaces, separate code and data segments, hardware address protection enforcement, and a centralized update process all contribute to error control. Inadvertent errors are isolated and restricted in scope.

The NSC approach appears to provide a unique solution to the network security problems since it can effectively control network access, provide audit data collection, and provide protection against tampering or modification of the access control data base. Since multiple NSCs can operate together, issues such as modular expandability, regional subnets and local control over resources can also be addressed by this solution approach. With the introduction of the NBS Data Encryption Algorithm and remotely keyed cryptographic devices using this algorithm, the NSC can provide a viable solution to the problems caused by interconnecting computers into network configurations.

REFERENCES

[1] Cole, G. D., "Design Alternatives for Computer Network Security," NBS Special Publication 500-21, Volume I.

[2] Cotton, I.W. and Meissner, P., "Approaches to Controlling Personal Access to Computer Terminals." Proceedings of 1975 Symposium COMPUTER NETWORKS: Trends and Applications, Gaithersburg, Maryland, June 1975.

[3] Branstad, D. K., "Encryption Protection in Computer Data Communications," Fourth Data Communications Symposium, Quebec City, Canada, October 1975.

[4] Branstad, D. K., "Security Aspects of Computer Networks," AIAA Computer Network Conference, Huntsville, Alabama, April 1973.

[5] Kaufman, D. J., "A Secure National System for Electronic Funds Transfer," Proceedings of 1976 National Computer Conference, New York, New York, June 1976.

[6] Lampson, B. W., "Dynamic Protection Structures," Fall Joint Computer Conference, 1967.

[7] Knuth, D. E., The Art of Computer Programming, Vol. 1, (Addison-Wesley, Reading, Mass., 1968).

[8] Knuth, D. E., The Art of Computer Programming, Vol. 3, (Addison-Wesley, Reading, Mass., 1973).

[9] Popeck, G., "Protection Structures," Computer, June 1974.

[10] Schroeder, M. D., "Engineering a Security Kernel for Multics," Op. Sys. Review, Vol. 9, No. 5, November 1975.

[11] Rosenthal, R., "Network Access Techniques - A Review," National Computer Conference, 1976.

[12] Rosenthal, R., "Access to Computer Networks," 1975 IEEE Intercon Conference Record, April 1975.

NBS-114A (REV. 7-73)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO NBS SP 500-21, Volume 2	2. Gov't Accession No.	3. Recipient's Accession No.

4. TITLE AND SUBTITLE

COMPUTER SCIENCE & TECHNOLOGY:

The Network Security Center: A System Level Approach to Computer Network Security

5. Publication Date
January 1978

6. Performing Organization Code

7. AUTHOR(S)
Frank Heinrich

8. Performing Organ. Report No.

9. PERFORMING ORGANIZATION NAME AND ADDRESS

System Development Corporation
2500 Colorado Avenue
Santa Monica, California 90406

10. Project/Task/Work Unit No.
6401112

11. Contract/Grant No.
NBS 5-35934

12. Sponsoring Organization Name and Complete Address *(Street, City, State, ZIP)*

Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, D. C. 20234

13. Type of Report & Period Covered
Final

14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

Library of Congress Catalog Card Number: 77-608304

16. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

This report describes a unique approach to the solution of computer network security problems, and provides guidance in the areas of network security architectural issues and implementation options. The approach is based on a network resource, called a Network Security Center (NSC), which performs the functions of user identification/authentication and access request authorization. The NSC works in concert with Network Cryptographic Devices (NCDs) to enforce access control policy through the creation or denial of logically separate cryptographic connections between subjects (users) and objects (resources). The use of a NSC in a network permits effective control over network access, provides for audit data collection, and provides protection against tampering or modification of the access control data base. The architecture presented permits multiple NSCs to operate together, thus addressing issues such as modular expandability, regional subnets, and local control over resources. Network Cryptographic Devices that use the NBS Data Encryption Standard algorithm and are capable of being remotely keyed are a vital part of the NSC security approach. NCDs provide end-to-end cryptographic message protection, source-destination authentication of identity and, through the remote keying capability, the enforcement mechanism for NSC access control decisions. Implementation options for an NSC are presented, covering the areas of data structures, I/O structure, control structure, and size and performance limitations.

17. KEY WORDS *(six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)*

Access authorization; access control; authentication; computer network security; cryptography; end-to-end encryption; inter-computer network; internetting; NBS Data Encryption Standard; Network Cryptographic Devices; Network Security Center

18. AVAILABILITY ☒ Unlimited

□ For Official Distribution. Do Not Release to NTIS

☒ Order From Sup. of Doc., U.S. Government Printing Office
Washington, D.C. 20402, SD Cat. No. C13.10:500-21, Vol. 2

□ Order From National Technical Information Service (NTIS)
Springfield, Virginia 22151

19. SECURITY CLASS (THIS REPORT)
UNCLASSIFIED

20. SECURITY CLASS (THIS PAGE)
UNCLASSIFIED

21. NO. OF PAGES
69

22. Price
$6.00 per set

ANNOUNCEMENT OF NEW PUBLICATIONS ON COMPUTER SCIENCE & TECHNOLOGY

Superintendent of Documents,
Government Printing Office,
Washington, D. C. 20402

Dear Sir:

Please add my name to the announcement list of new publications to be issued in the series: National Bureau of Standards Special Publication 500-.

Name ____

Company ____

Address ____

City ____________________ State ____________ Zip Code __________

(Notification key N-503)

☆ U.S. GOVERNMENT PRINTING OFFICE : 1978 O—253-749

NBS TECHNICAL PUBLICATIONS

PERIODICALS

JOURNAL OF RESEARCH—The Journal of Research of the National Bureau of Standards reports NBS research and development in those disciplines of the physical and engineering sciences in which the Bureau is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology, and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Bureau's technical and scientific programs. As a special service to subscribers each issue contains complete citations to all recent NBS publications in NBS and non-NBS media. Issued six times a year. Annual subscription: domestic $17.00; foreign $21.25. Single copy, $3.00 domestic; $3.75 foreign.

Note: The Journal was formerly published in two sections: Section A "Physics and Chemistry" and Section B "Mathematical Sciences."

DIMENSIONS/NBS (formerly Technical News Bulletin)—This monthly magazine is published to inform scientists, engineers, businessmen, industry, teachers, students, and consumers of the latest advances in science and technology, with primary emphasis on the work at NBS. The magazine highlights and reviews such issues as energy research, fire protection, building technology, metric conversion, pollution abatement, health and safety, and consumer product performance. In addition, it reports the results of Bureau programs in measurement standards and techniques, properties of matter and materials, engineering standards and services, instrumentation, and automatic data processing.

Annual subscription: Domestic, $12.50; Foreign $15.65.

NONPERIODICALS

Monographs—Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NBS, NBS annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a world-wide program coordinated by NBS. Program under authority of National Standard Data Act (Public Law 90-396).

NOTE: At present the principal publication outlet for these data is the Journal of Physical and Chemical Reference Data (JPCRD) published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements available from ACS, 1155 Sixteenth St. N.W., Wash., D.C. 20056.

Building Science Series—Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The purpose of the standards is to establish nationally recognized requirements for products, and to provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order **above** *NBS publications from: Superintendent of Documents, Government Printing Office, Washington, D.C. 20402.*

Order **following** *NBS publications—NBSIR's and FIPS from the National Technical Information Services, Springfield, Va. 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NBS Interagency Reports (NBSIR)—A special series of interim or final reports on work performed by NBS for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Services (Springfield, Va. 22161) in paper copy or microfiche form.

BIBLIOGRAPHIC SUBSCRIPTION SERVICES

The following current-awareness and literature-survey bibliographies are issued periodically by the Bureau:

Cryogenic Data Center Current Awareness Service. A literature survey issued biweekly. Annual subscription: Domestic, $25.00; Foreign, $30.00.

Liquified Natural Gas. A literature survey issued quarterly. Annual subscription: $20.00.

Superconducting Devices and Materials. A literature survey issued quarterly. Annual subscription: $30.00. Send subscription orders and remittances for the preceding bibliographic services to National Bureau of Standards, Cryogenic Data Center (275.02) Boulder, Colorado 80302.

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Washington, D.C. 20234

OFFICIAL BUSINESS

Penalty for Private Use, $300