NISTIR 90-4250

SECURE DATA NETWORK SYSTEM (SDNS) NETWORK, TRANSPORT, AND MESSAGE SECURITY PROTOCOLS

Charles Dinkel
Editor

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards
and Technology
National Computer Systems Laboratory
Gaithersburg, MD 20899

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary

Lee Mercer, Deputy Under Secretary
for Technology

NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
John W. Lyons, Director

NIST

NISTIR 90-4250

SECURE DATA NETWORK SYSTEM (SDNS) NETWORK, TRANSPORT, AND MESSAGE SECURITY PROTOCOLS

Charles Dinkel
Editor

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards
and Technology
National Computer Systems Laboratory
Gaithersburg, MD 20899

February 1990

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
Lee Mercer, Deputy Under Secretary
for Technology
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
John W. Lyons, Director

Table of Contents

Page

Foreword iii

Introduction vii

SDNS Secure Data Network System
Security Protocol 3 (SP3) 1

SDNS Secure Data Network System
Security Protocol 4 (SP4) 39

SDNS Secure Data Network System
Message Security Protocol 63

SDNS Secure Data Network System
Directory Specifications for Utilization with
the SDNS Message Security Protocol 85

FOREWORD

The Secure Data Network System (SDNS) architecture and a set of associated specifications were developed through a multi-organizational project sponsored by the National Security Agency (NSA). They are presented here as a basis for standardization of security services in the Open Systems Interconnection (OSI) architecture. The National Institute of Standards and Technology (NIST) intends to encourage widespread adoption of the resulting standards and the implementation of these security services into a wide spectrum of vendor products.

NIST is publishing the specifications that resulted from Phase I of the SDNS project for review and comment from potential government and commercial users of security products. The specifications are not complete or totally consistent, either internally or with a number of other security projects in the National and International Standards arena. Readers of these documents should recognize that these specifications are subject to modification for various reasons as they progress through the standards process. The sponsor and participants in the SDNS project are acknowledged for the work accomplished and their support in developing and releasing these specifications.

The SDNS project was initiated by NSA to investigate methods of implementing security in a distributed computer network. The results of this project include a set of specifications that include security services, protocols and mechanisms for protecting user data in networks that are based on the OSI computer network model. Productive security services that protect user data are specified and supportive security services, such as key management and access control, are also provided. No cryptographic algorithms are included in these specifications.

NIST is working with NSA and industry to identify and develop a framework of base standards for network security. In 1989, NIST established the OSI Security Laboratory where interested researchers from government and industry develop and demonstrate new ideas in network security. The major goals of NIST's network security activities are to:

- Identify and develop security standards for open systems

- Specify a key management system that supports these security standards

- Encourage the development of interoperable equipment

The documents resulting from Phase I of the SDNS project are as follows:

SDN.301 -	Security Protocol 3 (SP3)
SDN.401 -	Security Protocol 4 (SP4)
SDN.601 -	Key Management Profile - Communication Protocol Requirements for Support of the SDNS Key Management Protocol
SDN.701 -	Message Security Protocol
SDN.702 -	SDNS Directory Specifications for Utilization with the SDNS Message Security Protocol
SDN.801 -	Access Control Concepts Document
SDN.802 -	Access Control Specification
SDN.902 -	Key Management Protocol - Definition of Services Provided by the Key Management Application Service Element
SDN.903 -	Key Management Protocol - Specification of the Protocol for Services Provided by the Key Management Application Service Element
SDN.906 -	Key Management Protocol - SDNS Traffic Key Attribute Negotiation

Because of the wide spread interest in the SDNS project, NIST is publishing these ten documents as three Reports entitled: **Security Protocols, Key Management,** and **Access Control.** The following diagram shows the relationship and contents of these reports.

NIST REPORT				
SECURITY PROTOCOLS	SDN.301 SECURITY PROTOCOL 3 (SP3)	SDN.401 SECURITY PROTOCOL 4 (SP4)	SDN.701 MESSAGE SECURITY PROTOCOL	SDN.702 DIRECTORY SPECS FOR USE WITH MSP
KEY MANAGEMENT	SDN.601 KEY MANAGEMENT PROFILE	SDN.902 KMP DEFINITION OF SERVICES PROVIDED BY KM ASE	SDN.903 KMP SERVICES PROVIDED BY KM ASE	SDN.906 KMP TRAFFIC KEY ATTRIBUTE NEGOTIATION
ACCESS CONTROL	SDN.801 ACCESS CONTROL CONCEPT DOCUMENT	SDN.802 ACCESS CONTROL SPECIFICATION		

INTRODUCTION

NISTIR 90-4250 includes four security protocol documents developed by the National Security Agency (NSA) as output from the Secure Data Network System (SDNS) project. **SDN.301** provides the framework for security at layer 3 (SP3) of the OSI model. The SP3 protocol provides connectionless network service with confidentiality, integrity, or both. **SDN.301.1**, an addendum to **SDN.301**, discusses an optional mechanism for providing routing flexibility in internetworks.

Cryptographic techniques to provide data protection for transport connections or for connectionless-mode transmission are described in **SDN.401.** Two classes of Transport layer security (SP4) are discussed in the document. SP4C is connection oriented protection in which each transport connection is individually protected with a different cryptographic key. End system to end system protection in which all connections between a pair of end systems are protected with the same cryptographic key is provided by SP4E.

Specifications for message security service and protocol are contained in **SDN.701.** This document describes the additions to the CCITT X.400 Message Handling System Recommendations that permit any type of message to be sent and received securely. The Message Security Protocol provides writer to reader confidentiality, integrity, data origin authentication, non-repudiation with proof of origin, access control for message transfer, and request of a signed receipt of the received message.

Directory System Specifications for Message Security Protocol are covered in **SDN.702.** This document specifies additions to the Directory System described in the 1988 X.500 series of CCITT Recommendations to support some key management functions for use by X.400 messages protected by SDNS Message Security Protocol. New attribute types and object classes for inclusion in the Directory Information Base (DIB) in support of these functions are discussed.

The security protocols addressed in **NISTIR 90-4250** are heavily dependent on a cryptographic management and access control service. The SDNS specifications for these services are contained in **NISTIR 90-4262** and **NISTIR 90-4259**, respectively.

Comments and feedback are solicited by NIST.

SPECIFICATION SDN.301
Revision 1.5 1989-05-15

SDNS
Secure Data
Network System

Security Protocol 3
(SP3)

Forward:

The SDNS architecture, and its associated specifications, were developed as a cooperative project between government and industry. The project was sponsored by the National Security Agency, and supported by the National Institute of Standards and Technology (formerly the National Bureau of Standards) and the Defense Communications Agency. Twelve leading U.S. companies in computers and telecommunications made significant contributions of their technical talents and development resources. The combined efforts of these organizations produced the basis for improved security technology, interoperable security standards, and cost-effective security for computers and telecommunications.

Introductory note:

This document provides the framework for the SDNS Security Protocol at layer three (SP3). This document is being circulated for comment and approval. It is subject to change during the development phase of SDNS.

TABLE OF CONTENTS

0 INTRODUCTION 1

1 SCOPE AND FIELD OF APPLICATION 2

2 REFERENCES 2

3 DEFINITIONS 3
- 3.1 Reference Model Definitions 3
- 3.2 Service Convention Definitions 3
- 3.3 Network Layer Architecture Definitions 3
- 3.4 Network Layer Addressing Definitions 3
- 3.5 Connectionless Network Protocol Definitions 4
- 3.6 Additional Definitions 4

4 SYMBOLS AND ABBREVIATIONS 4
- 4.1 Data Units 4
- 4.2 Protocol Data Unit Fields 4
- 4.3 Parameters 4
- 4.4 Miscellaneous 4

5 OVERVIEW OF THE PROTOCOL 5
- 5.1 Definition 5
- 5.2 Addressing 5
- 5.3 Services Provided 6
  - 5.3.1 Explicit Parameters 6
  - 5.3.2 Security Services 7
- 5.4 Services Assumed 8
- 5.5 Addressing Modes 8

6 PROTOCOL FUNCTIONS 9
- 6.1 Outline Functions 9
- 6.2 Checks 9
- 6.3 Keys and Attributes 9
- 6.4 Transmission Functions 11
  - 6.4.1 Initial Checks 11
  - 6.4.2 Identification of the Key 11
  - 6.4.3 Format of the Protected SP3 Header 11
  - 6.4.4 Encipherment and Calculation of ICV 12
  - 6.4.5 Key Identifier 13
  - 6.4.6 Network Request 13
- 6.5 Reception Functions 13
  - 6.5.1 Identification 13
  - 6.5.2 Decryption and ICV Checking 13
  - 6.5.3 Direction Check 13
  - 6.5.4 Padding 13
  - 6.5.5 Address Mode Determination 14
  - 6.5.6 Presence of Security Label 14
  - 6.5.7 Value of Security Label 14
  - 6.5.8 Other Checks 14
  - 6.5.9 Other Processing 14
  - 6.5.10 Parameters of the SP3 Indication 14

7 STRUCTURE AND ENCODING OF PDUS 16
7.1 Clear Header 17
7.1.1 LI 17
7.1.2 SE 17
7.1.3 KEY-ID 17
7.2 Protected Header 17
7.2.1 LI 18
7.2.2 Flags 18
7.2.3 LABEL 19
7.2.4 PAD 20
7.2.5 SRCNSAP 20
7.2.6 DSTNSAP 20
7.2.7 CLNPHDR 21
7.2.8 IPHDR 21

8 MINIMUM ESSENTIAL REQUIREMENTS 21
8.1 Addressing Modes 21
8.2 Security Services 21
8.3 Padding 22

9 ILLUSTRATIONS OF SP3 OPERATION 22
9.1 SP3 Operation at an End System 22
9.1.1 Address Mode N 22
9.1.2 Address Mode A 22
9.1.3 Address Mode I 22
9.1.4 Address Mode D 23

9.2 SP3 Operation at an Intermediate System 23
9.2.1 Address Mode A 24
9.2.2 Address Mode I 24
9.2.3 Address Mode D 25

0 Introduction

The SP3 Protocol is one of a number of protocols defined under the United States Government SDNS (Secure Data Network System) activity. The purpose of this protocol is to provide various security services, through the use of cryptographic mechanisms. This document defines the services provided by the protocol, the assumptions made by the protocol, the formats of the Protocol Data Units (PDUs) used by the protocol, and the functions performed by the protocol.

SP3 is a SNICP (subnetwork independent convergence protocol) as defined in ISO 8648. The SP3 protocol provides connectionless network service with confidentiality, integrity, or both. The basic mode of operation of SP3 is encapsulation of NSDUs (network service data units). In certain modes of operation, SP3 can also independently encapsulate network layer N-PDU fragments.

The service which SP3 assumes from the sublayer below it is the connectionless network service, as defined by ISO 8348/DAD1. Thus, SP3 might, as an example, interface with the ISO CLNP protocol (ISO 8473) at its lower boundary.

An SP3 entity may be located either at an end system, or, with certain assumptions, at an intermediate system.

The SP3 protocol is heavily dependent on an external key service for supply of cryptographic keys and the attributes associated with those keys. The attributes associated with a key include not only security attributes of the data which may be sent using that key, but also the SP3 header format options to be used.

Figure 0-1 below indicates the interfaces of SP3 to the transport layer above it and to the network sublayer below it. It also shows the Traffic Encipherment Key Management Information Base (TEK MIB) and indicates the external service which maintains it.

Note: The SP3 protocol is defined in ISO terms. However, a few details of the protocol are intended to support use with the USA DoD protocol suite, in particular interaction with the DoD IP protocol. Use with the DoD protocol suite is essentially identical with use with the ISO suite.

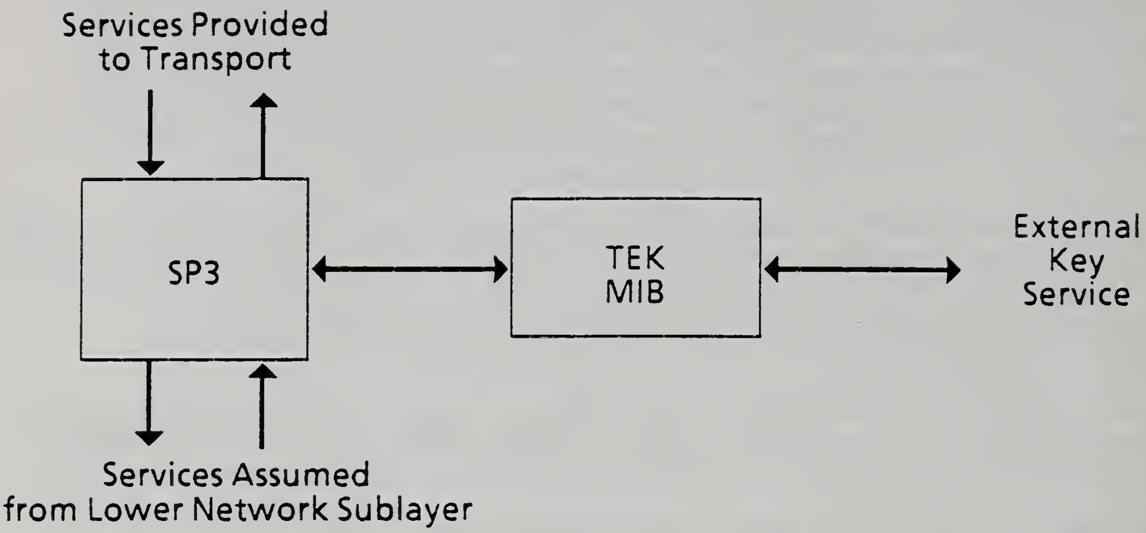

Services Provided to Transport
SP3
TEK MIB
External Key Service
Services Assumed from Lower Network Sublayer

Figure 0-1. SP3 Interfaces

1 Scope and Field of Application

This document specifies the SP3 security service and protocol. These exten to the connectionless network servie, described in ISO 8348/AD1, and the connectionless network protocol, described in ISO 8473. The purpose of SP3 is to provide various security services in the network layer, through the use of cryptographic mechanisms.

2 References

ISO 8473	01 March 1986 Final Text of DIS 8473, Protocol for Providing the Connectionless-mode Network Service
ISO 8348	15 July 1987 Final Text of DIS 8348: Information processing systems - Data communications - Network service definition
ISO 7498	15 Oct. 1987 Information Processing Systems - Open Systems Interconnection - Basic Reference Model
ISO 7498/2	March 1987 Information Processing Systems - Open Systems Interconnection - Security Architecture
ISO 8348/DAD1	Information Processing Systems - Data 07 Aug. 1985 Communications - Network Service Definition - Addendum 1
ISO 8348/DAD2	Information Processing Systems - Data 29 Feb. 1986 Communications - Network Service Definition - Addendum 2 Covering Network Layer Addressing

ISO 8648	Information Processing Systems - Data Communications - Internal Organization of the Network Layer
ISO TR 8509	Technical Report 8509, OSI Service Conventions
RFC 986	Guidelines for the use of Internet-IP addresses in the ISO Connectionless - Mode Network Protocol
MIL-STD-1777	Military Standard Internet Protocol 12 Aug 1983 -

3 Definitions

3.1 Reference Model Definitions

This document makes use of the following terms as defined in ISO 7498:

a. End System
b. Network Entity
c. Network Layer
d. Network Protocol
e. Network Protocol Data Unit
f. Network Relay
g. Network Service
h. Network Service Data Unit
i. Network Service Access Point
j. Network Service Access Point Address
k. Routing
l. Service
m. Service Data Unit

3.2 Service Convention Definitions

This Protocol document makes use of the following terms as extracted from the OSI Service Conventions (ISO TR 8509):

n. Service provider
o. Service user

3.3 Network Layer Architecture Definitions

The following term was extracted from the Internal Organization of the Network Layer (ISO 8648)

p. Intermediate System
q. Relay System
r. Subnetwork

3.4 Network Layer Addressing Definitions

This document makes use of the following terms that were extracted from ISO 8348/DAD2 Information Processing Systems - Data Communications - Network Service Definition Addendum 2 Covering Network Layer Addressing:

s. Network addressing domain

t. Network protocol address information
u. Subnetwork point of attachment

3.5 Connectionless Network Protocol Definitions

This document makes use of the following concepts from DIS 8473, Data Communications Protocol for Providing the Connectionless-Mode Network Service:

v. initial PDU - a protocol data unit carrying the whole of the user data from an N-UNITDATA request.
w. local matter - a decision made by a system concerning its behavior in the Network Layer that is not prescribed or constrained by ISO 8473.
x. reassembly - the act of regenerating an initial PDU from two or more derived PDU's.
y. segment - a distinct unit of data consisting of part or all of the user data provided in the N-UNITDATA request and delivered in the N-UNITDATA indication.

3.6 Additional Definitions

For the purpose of this document, the following definitions apply:

1. An NSAP is said to be "served by" an SP3 entity at an intermediate system if the NSAP is on a subnetwork reachable via that intermediate system.

2. Pairwise key: a key generated for two specific users and unavailable to other users.

4 Symbols and Abbreviations

4.1 Data Units

NSDU	Network Service Data unit
PDU	Protocol Data Unit

4.2 Protocol Data Unit Fields

DA	Destination Address
LI	Length Indicator
SA	Source Address

4.3 Parameters

DA	Destination Address
QOS	Quality of Service
SA	Source Address

4.4 Miscellaneous

CLNP	Connectionless-mode Network Protocol
ICV	Integrity Check Value
NS	Network Service
NSAP	Network Service Access Point
SN	Subnetwork

5 Overview of the Protocol

5.1 Definition

The SP3 protocol is a SNICP (subnetwork independent convergence protocol). SP3 provides at its upper boundary a secure connectionless network service and assumes at its lower boundary a connectionless network service. SP3 uses, in a subnetwork independent fashion, the services at its lower boundary to provide an improved service at its upper boundary.

Note: if the sublayer below SP3 provides a connection-oriented service, the fundamental encapsulation function of SP3 could very likely be used to provide a connection-oriented communication service on its upper boundary. The means by which this might be done remain for further study.

Figure 5-1 shows SP3 at the top of the network layer, both in an end system and in an intermediate system. In this Figure, SP3 is used to provide security services between end system A and intermediate system B. Some other method (such as physical protection of the subnetwork) must be used to provide security between B and C if required.

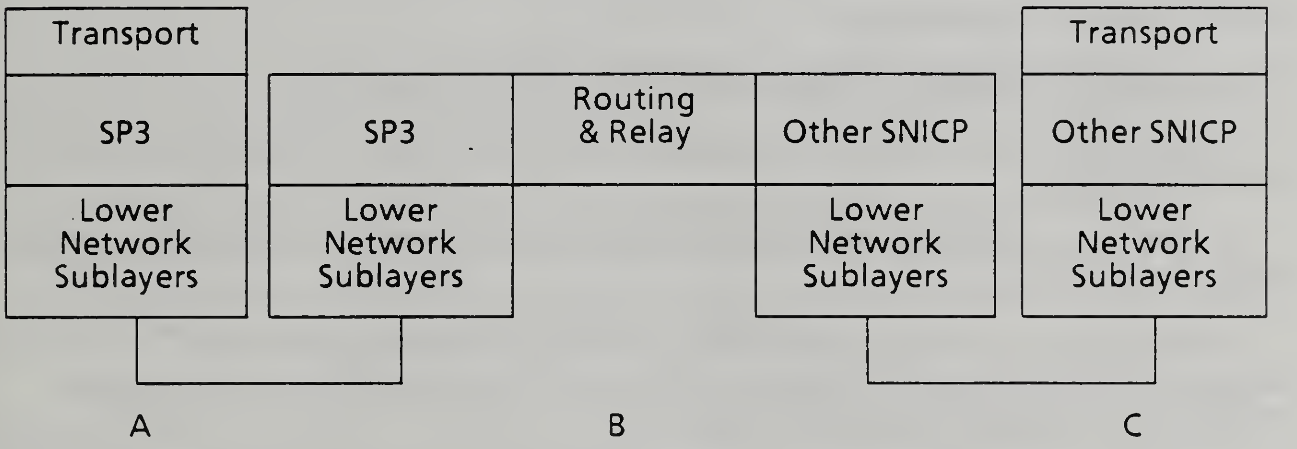


Figure 5-1. SP3 at Top of Network Layer

Note that the SNICP in end system C does not have a peer in system A; instead its peer is at intermediate system B. Thus the protocol conducted by the SNICP at C ends at B. Intermediate system B is responsible for performing a convergence function which allows the secure connectionless service to be maintained.

5.2 Addressing

Addresses referred to in this protocol are NSAP addresses. Their syntax and semantics are defined in ISO 8348/DAD2.

When using SP3 with the DoD protocol suite, the ISO NSAP address is functionally replaced by the concatenation of the DoD Internet Address with the IP "Next Protocol" field as defined in RFC 986 and GOSIP.

5.3 Services Provided

The service provided by SP3 will be referred to with the prefix "SP3". The primitives are:

Primitives		Parameters
SP3_UNITDATA	Request Indication	SP3_Destination_Address SP3_Source_Address SP3_Quality_of_Service SP3_Userdata

The services provided are the same as those provided by the ISO CLNP, (ISO 8348), with some additional Quality of Service parameters.

5.3.1 Explicit Parameters

This section gives a more complete list of the parameters of the Request and Indication, emphasizing the parameters associated with security. "Boolean" indicates a true or false condition.

5.3.1.1 Request Parameters

The parameters associated with the SP3_UNITDATA Request are:

1. SP3_Destination_Address.. End system NSAP address (N_Destination_Address) as in ISO 8348.

2. SP3_Source_Address. End system NSAP address (N_Source_Address) as in ISO 8348.

3. SP3_Quality_of_Service. As in ISO 8348/DAD1 (plus the addition of Security_Label_Requested) (N_Quality_of_Service). The following security-related quality of service parameters are supported.

   a. Integrity_Requested: boolean. A TRUE setting indicates data integrity is requested.

   b. Confidentiality_Requested: boolean. A TRUE setting indicates confidentiality is requested.

   c. Security_Label_Requested. Not always present. If present, this is the security label associated with the data, which must be indicated with integrity to the destination address along with the data.

4. SP3_Userdata. An ordered multiple of octets, as in ISO 8348/DAD1 (N Userdata).

5.3.1.2 Indication Parameters

The parameters associated with the SP3_UNITDATA Indication are:

1. SP3_Destination_Address. End System NSAP (N_Destination_Address) as in ISO 8348.

2. SP3_Source_Address. End System NSAP (N_Source_Address) as in 8348.

3. SP3_Quality_of_Service. As in ISO 8348. (N_Quality _of_Service).

    a. Integrity_Indicated: boolean. A TRUE setting indicates data integrity provided.

    b. Confidentiality_Indicated: boolean. A TRUE setting indicates data confidentiality provided.

    c. Security_Label_Indicated: Not always present. If present, this is the security label associated with the data.

4. SP3_Userdata. An ordered multiple of octets, as in ISO 8348 (N_Userdata).

5.3.2 Security Services

Of the security services provided by SP3, some are always provided, while others are requested as part of the SP3_UNITDATA Request.

1. SP3 provides integrity of the SP3_Userdata (Connectionless Integrity) if requested via the Integrity_Requested parameter. NOTE: In many cases, SP3 provides integrity even if not requested.

2. SP3 provides confidentiality (Connectionless Confidentiality) of the SP3 Userdata, if requested via the Confidentiality_Requested parameter.

3. If requested via the Security_Label_Requested parameter, SP3 will accept and validate a security label and deliver it with the SP3_Userdata.

4. SP3 never explicitly provides for the integrity of the non-security-related SP3 QOS parameters. Integrity of these parameters may be obtained by using the SP3-D or SP3-I addressing modes (see Sections 9.2 and 9.3), in an implementation which encapsulates these parameters into the SP3 PDU.

5. SP3 indicates, via the Integrity_Indicated parameter, whether connectionless data integrity was supplied for this unit.

6. SP3 indicates, via the Confidentiality_Indicated parameter, whether confidentiality was supplied for this unit.

7. SP3 supports data origin authentication. If SP3 is used between end systems, then this authentication is supplied via key management techniques. If SP3 is concatenated with another SNICP at an intermediate system, then the data origin authentication is supplied by a combination of key management and explicit protection of source and destination NSAP addresses.

When the peer SP3 is located at an intermediate system, then either the SP3-A, SP3-I, or SP3-D mode is used. In this case, the origin of the data is identified by the Source Address within the protected header.The peer SP3 is authenticated by pairwise keying. The source end system is authenticated to be a member of the set of addresses served by the peer SP3 entity.

8. SP3 supplies in the Indication a representation (Security_Label_Indicated) of a security label which has been carried with integrity.

9. SP3 supports access control. The key service external to SP3 may choose, for access control reasons, not to make a key available for communication between two SP3 entities. This decision by the key service implies an access control service at the upper interface of SP3, since SP3 will not allow communication if an appropriate key is not present. In addition, if a security label is provided for the data, then SP3 will make an access control check against this label on both transmit and receive.

5.4 Services Assumed

The service assumed by SP3 on its lower boundary will be referred to with the prefix "BN_" (for "Black Network"). The primitives are:

Primitives		Parameters
BN_UNITDATA	Request	BN_Destination_Address
	Indication	BN_Source_Address
		BN_Quality_of_Service
		BN_Userdata

The services assumed are the same as those defined in the ISO CLNP (ISO 8348/DAD1).

5.5 Addressing Modes

SP3 has four possible modes of conveying addressing information from SP3 source to SP3 destination. They utilize different SP3 header options. The modes are:

1. SP3-N -- no explicit addressing information is contained in the SP3 header. This mode is used between two end systems.

2. SP3-A -- the SP3 header contains the NSAP addresses of the source and destination end systems. This mode can be used at end systems and at intermediate systems. SP3-A always provides integrity of the SP3_Userdata and the addresses, whether or not requested by the user. Confidentiality of the SP3_Userdata is determined by the SP3_UNITDATA Request SP3_Quality_of_Service parameter

3. SP3-I -- the SP3 header includes an ISO CLNP header, in the format specified in ISO 8473. This CLNP header includes the addresses of the source and destination end systems, as well as other communications parameters. The SP3-I mode can be used at end systems and at intermediate systems. It allows independent encapsulation of CLNP fragments. SP3-I always provides integrity of the Userdata and CLNP header whether or not requested by the user. Confidentiality of the SP3

Userdata is determined by the SP3_UNITDATA Request SP3_Quality_of_Service parameter.

4. SP3-D -- the SP3 header includes a DoD IP header, in the format specified by MIL-STD-1777. This IP header includes the addresses of the source and destination end systems, as well as other communications parameters. The SP3-D mode can be used at end systems and intermediate systems in networks which use the DoD IP protocols. SP3-D allows independent encapsulation of IP fragments. SP3-D always provides integrity of the Userdata and IP header whether or not requested by the user. Confidentiality of the SP3_Userdata is determined by the SP3_UNITDATA Request SP3_Quality_of_Service parameter.

6 Protocol Functions

6.1 Outline of Functions

The SP3 entity has access to a "Traffic Encryption Key Management Information Base" (TEK MIB). Associated with each key in the TEK MIB are a number of attributes describing how the key is to be used. In response to an SP3_UNITDATA Request, the SP3 attempts to identify a key with proper attributes. If one is found, the SP3 constructs a protected data unit containing the SP3_Userdata of the Request and some protocol control information, uses the key to encrypt and/or compute an Integrity Check Value (ICV) for the protected data, and adds some unprotected control information to form a PDU. The resulting PDU is then passed to the next lower sublayer as the BN_Userdata of a BN_UNITDATA Request. BN_Destination Address, BN_Source_Address, and BN_Quality_of_Service are passed to the next lower sublayer in the BN_UNITDATA Request.

In response to a BN_UNITDATA Indication, an SP3 entity identifies the key from the key identifier carried in the PDU. The SP3 uses the key to decrypt and/or verify the ICV of the protected data. The SP3_Userdata portion of the PDU is extracted, and passed to the next layer above in the SP3_UNITDATA Indication. SP3_Destination Address, SP3_Source_Address, SP3_Quality_of_Service are derived from the related BN UNITDATA Indication parameters or protocol control information and are passed to the next layer above in the SP3_UNITDATA Indication.

6.2 Checks

At many points in the following descriptions, it is stated that the SP3 entity checks that some condition is satisfied. Unless otherwise specified, whenever such a check fails, the SP3 entity discards the data currently being processed. Optionally, the entity may also file an audit report. What failures are to be audited is a local matter.

6.3 Keys and Attributes

This protocol assumes that each cryptographic association is defined by a set of attributes at each end system. The following paragraphs describe these attributes and list the defined mnemonics used to refer to the attributes in this specification. SP3 uses the keys and key attributes available to the SP3 entity via the TEK MIB to determine processing characteristics of the user data. The attributes associated with each key are defined as follows.

1. The local identifier of the key.

2. The remote identifier of the key.

3. Whether this key is to be used to provide confidentiality services.

   a. confid : Boolean

4. Whether this key is to be used to provide integrity, and if so, what length of ICV is to be used.

   a. integ : Boolean

   b. ICV_Length: The ICV length to be used if integ is TRUE.

5. Whether an explicit security label is required to be used with this key, and, if so, the nature ("Defining Authority") of that label. Other labels may be defined in the future and added to this list.

   a. ppl_abs : Explicit security label absent.

   b. ppl_DoD : DoD security label must be present.

The processing descriptions below describe only the handling of DoD security labels. The processing of other labels is analogous.

6. The addressing mode to be used with this key.

   a. ad_none : No address options are to be used, SP3-N.

   b. ad_A : Both the source and destination NSAP address options are to be used, SP3-A.

   c. ad_CLNP_hdr : The encapsulated CLNP header option is to be used, SP3-I.

   d. ad_IP_hdr : The encapsulated DoD IP header option is to be used, SP3-D.

7. The set of security labels to be used with this key.

NOTE: The key service guarantees that this is either the set of security labels shared by the two holders of this key or exactly one label which is within that set.

8. The NSAP address of the peer SP3 entity.

9. All NSAP addresses served via the peer SP3 entity.

10. Whether this SP3 was Initiator or Responder in creation of this key.

11. The algorithms used to perform encipherment using this key, to form an ICV with this key, or both.

6.3.1 Relations Among Key Attributes

The attributes of keys must obey the following relations:

1. If not Integ, then ad_none.

2. If not Integ, then xsl_abs.

6.4 TRANSMISSION FUNCTIONS

The transmission functions are those involved in processing an SP3_UNITDATA Request.

6.4.1 Initial Checks

The SP3 checks that SP3_Source_Address is an NSAP served by this SP3 entity. The SP3 checks that the security label, if any, associated with the request lies within the range of security labels that may be legitimately processed by this SP3 entity.

6.4.2 Identification of the Key

The SP3 entity identifies among the keys available to it a key whose attributes satisfy the following conditions:

1. The SP3_Destination_Address of the Request is one of the NSAPs served by the peer SP3 entity.

2. If Confidentiality_Requested then confid.

3. If Integrity_Requested then integ.

4. The Security_Label_Requested, if any, must lie in the set of security labels associated with the key. If Security_Label_Requested is present, then either:

   a. The set of security labels associated with the key must consist of precisely one value, or

   b. xsl_abs is false.

5. If xsl_DoD, then Security_Label_Requested must be a DoD security label.

6. If xsl_abs, then it must be permissable to transmit the data of the Request without an explicit security label.

The procedure to be followed if more than one key satisfies these conditions, or if no key satisfies these conditions, is a local matter.

6.4.3 Format of the Protected SP3 Header

The format of the protected portion of the SP3 header depends on the selected addressing mode. If the addressing mode is N or A, then each parameter in the header is separately formatted. If the addressing mode is I or D, then the source and destination addresses and the security label, if one is present, are incorporated in a CLNP or IP header, which is included as a single field of the SP3 header.

6.4.3.1. Addresses

If ad_A, then the SP3_Source_Address parameter is placed in the SRCNSAP address field of the PDU, and the SP3_Destination_Address parameter is placed in the DSTNSAP address field of the PDU.

6.4.3.2 Encapsulated Headers

If ad_CLNP_hdr, then a CLNP header is constructed for encapsulation in the SP3 header, with the SP3_Source_Address parameter as source NSAP address, and the SP3_Destination_Address parameter as destination NSAP address. This header conforms to ISO 8473. It includes the security relevant SP3_Quality_of_Service (SP3 QOS) parameters to the extent that they can be carried in the CLNP header in the SP3_UNITDATA Request and may also include fragmentation information. Record routing and source routing are Quality of Service (QOS) parameters that may be included in the protected header, depending on local policy.

If ad_IP_hdr, then a DoD IP header is constructed for encapsulation in the SP3 header, with the SP3_Source_Address parameter as IP source address and the SP3 Destination_Address parameter as IP destination address. This header conforms to the IP specification. It includes the SP3_QOS parameters to the extent that they can be carried in the IP header and may also include fragmentation information. Record routing and source routing are QOS parameters that may be included in the protected header, depending on local policy.

6.4.3.3 Security Label Option

If xsl_DoD, then the DoD security label associated with the Request is placed in the protected header. If the address mode is N or A, then the security label is placed in the security label option and the Defining Authority is set to DoD. If the address mode is I or D, then the security label is placed in the CLNP or IP header, in the standard header format. If xsl_abs, then no security label shall be placed in the SP3 protected header.

6.4.3.4 Padding

The use of padding is optional. Whether the padding option is used by a SP3 entity, and how much padding is present, on a per PDU basis, is a local matter. The amount of the padding is limited by the maximum header size.

6.4.3.5 Direction Flag

If this SP3 was Initiator in creation of this key, then the Initiator-to-Responder Flag in the PDU is set to True. Otherwise it is set to False.

6.4.4 Encipherment and Calculation of ICV

The PDU shall be enciphered if and only if the key attribute confid is TRUE. An ICV shall be calculated if and only if the key attribute integ is TRUE. The length of the ICV shall be determined by the key attribute ICV_length. Note that the only address mode which may be used with key attribute integ FALSE is ad_none.

6.4.5 Key Identifier

The remote identifier of the key shall be placed in the Key Identifier field of the PDU.

6.4.6 Network Request

The SP3 PDU is passed to the next lower protocol as the BN_Userdata parameter of a BN_UNITDATA Request.

The BN_Source_Address is the NSAP address of this SP3 entity. The content of non-security relevant BN_Quality_of_Service (BN_QOS) parameters is determined by local policy but may be obtained from the SP3_QOS.

The BN_Destination_Address is set to the NSAP address of the peer SP3 entity.

NOTE: If record route and source route parameters are in SP3_QOS parameters and are not passed as BN_QOS parameters, then the specified QOS may not be provided for the part of the route between source and destination SP3 entities.

6.5 Reception Functions

The reception functions are those involved in processing a BN_UNITDATA Indication.

6.5.1 Identification

The SP3 entity identifies among the keys available to it a key with local identifier equal to the Key Identifier in the received PDU. Descriptions of other reception functions refer to attributes of this key, as recorded in the TEK MIB.

Note: It is a serious error for there to be more than one key with the same local identifier. If no key with this Identifier is found, a security relevant event is indicated to the Layer Management Entity.

6.5.2 Decryption and ICV Checking

If the key attribute config is TRUE, the PDU shall be deciphered. If the key attribute ineg is TRUE, the PDU ICV shall be checked.

6.5.3 Direction Check

If the Initiator-to-Responder Flag in the PDU is True, then the SP3 checks that the SP3 was Responder in creating the key. If the Initiator-to-Responder Flag in the PDU is False, then the SP3 checks that the SP3 was Initiator in creating the key.

6.5.4 Padding

The SP3 removes any padding present in the PDU.

6.5.5 Address Mode Determination

If ad_none, the SP3 checks that no addresses or network header options are present in the PDU.

If ad_A, the SP3 checks that the source and destination NSAP addresses are present in the PDU and that no network headers are present.

If ad_CLNP_hdr, the SP3 checks that the CLNP header option is present and that the NSAP address options and other header options are not present.

If ad_IP_hdr, the SP3 checks that the DoD IP header option is present and that the NSAP address options and other header options are not present.

6.5.6 Presence of Security Label

If ppl_abs, the SP3 checks that the security label is absent from the PDU. In the N or A address modes, this means that the security label option in the SP3 header is not present. In the I or D address modes, this means that the security option in the encapsulated CLNP or IP header is not present.

If ppl_DoD, the SP3 checks that the security label is present. In address modes N or A, the security label option must be present in the SP3 header and the "Defining Authority" must be DoD. In address modes I or D, the security option must be present in the encapsulated header and the format must be correct.

6.5.7 Value of Security Label

If a security label is present in the PDU, the SP3 checks that the value of the label lies within the set of security labels associated with the key.

6.5.8 Destination Address Check

The SP3 checks that SP3_Destination_Address is an NSAP served by this SP3 entity.

6.5.9 Other Processing

If the address mode is I, the SP3 processes the encapsulated header according to ISO 8473 procedures. If the address mode is D, the SP3 processes the encapsulated header according to DoD IP procedures.

6.5.10 Parameters of the SP3 Indication

The data field (after decipherment, if required) is delivered to the SP3 User as the Userdata of an SP3_UNITDATA Indication. The other parameters of this indication are as follows:

6.5.10.1 Address Parameters

If the address mode is A, then SP3_Source_Address is the value from the SRCNSAP option, and SP3_Destination_Address is the value from the DSTNSAP option.

If the address mode is I or D, then SP3_Source_Address is the source address in the encapsulated CLNP or IP header, and SP3_Destination_Address is the destination address in the encapsulated CLNP or IP header.

Otherwise, the addressing mode is N and the values are taken from the BN Indication parameters:

SP3_Source_Address = BN_Source_Address and

SP3_Destination_Address = BN_Destination_Address.

The SP3 entity shall check that the SP3_Source_Address of the Indication is the address of an NSAP reachable via the peer SP3 entity.

6.5.10.2 QOS

If the address mode is A or N, the SP3_QOS is copied from the BN_Indication QOS for the non-security-relevant QOS parameters. If the address mode is I or D, the SP3 QOS is copied from the QOS parameters in the encapsulated CLNP or IP header.

6.5.10.2.1 Integrity_Indicated

Integrity_Indicated is set equal to the value associated with the key attribute integ

6.5.10.2.2 Confidentiality_Indicated

Confidentiality_Indicated is set equal to the value associated with the key attribute config.

6.5.10.2.3 Security_Label_Indicated

1. If a security label is present in the PDU, either in the label field of the protected header or in a security option within an encapsulated CLNP or IP header, then Security_Label_Indicated is set to the value of this label.

2. If the set of security labels associated with this key (section 6.3, item 7) consists of a single value, then Security_Label_Indicated is set to this label.

3. Otherwise, the Security_Label_Indicated parameter shall not be present in the indication.

7 Structure and Encoding of PDUs

The SP3 protocol uses a single PDU type.

Any PDU contains an integral number of octets. The octets in a PDU are numbered starting from one (1) and increasing in the order in which they are put into the Userdata parameter of a BN_UNITDATA Request. When consecutive octets are used to represent a binary number, the lower octet number has the most significant value. The bits in an octet are numbered from one (1) to eight (8), where bit one (1) is the low-order bit.

When the encoding of a PDU is represented using a diagram in this section,

1. Octets are shown with the lowest numbered octet to the left or above.

2. Within an octet, bits are shown with bit eight (8) to the left and bit (1) to the right.

The notations below a box show the length of each field in octets; "var" indicates that the field length is variable.

The format of a PDU is shown in Figure 7-1.

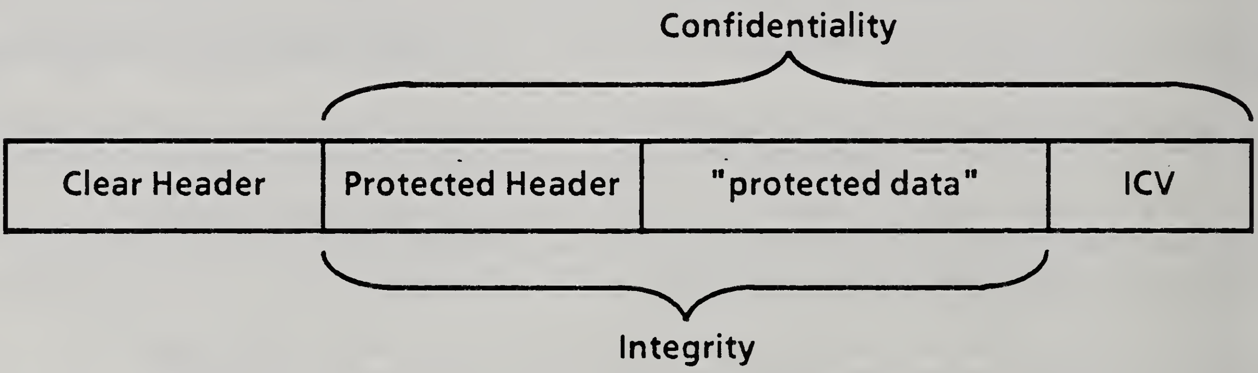


Figure 7-1. PDU Structure

The fields must appear in the order shown.

Note that (as specified in detail in section 6) the presence or absence of an "optional" field is specified by the parameters associated with the key. The optional fields are optional in that a given key will require the presence of certain fields and the absence of other fields. Once the key is decided, the presence or absence of each field is not optional, but determined.

7.1 Clear Header

Figure 7-2 shows the clear header. All fields are required and must appear in the order shown.

LI	SE	KEY_ID
1	1	var

Figure 7-2. Clear Header

7.1.1 LI

The length indicator field (LI) contains the length of the Clear Header in octets, excluding the length indicator field.

7.1.2 SE

This field contains the value 72 (48 hex). It may be interpreted as a PDU type code, indicating a Security PDU.

7.1.3 KEY_ID

Figure 7-3 shows the KEY-ID field.

Value of Key_ID
var

Figure 7-3. KEY_ID

The KEY_ID field contains the key identifier.

7.2 Protected Header

The contents of the protected header is determined from the key attribute addressing mode. Figures 7-4, 7-5, 7-6, and 7-7 show the protected header format for the ad_none (SP3-N), ad_A (SP3-A), ad_CLNP_hdr (SP3-I), and ad_IP_hdr (SP3-D) addressing modes.

The LI and FLAGS fields are required. All other fields are optional. Fields must occur in the order shown, if used.

LI	FLAGS	LABEL	PAD
1	1	var	var

Figure 7-4 SP3-N. Protected Header

LI	FLAGS	LABEL	PAD	SRCNSAP	DSTNSAP
1	1	var	var	var	var

Figure 7-5 SP3-A. Protected Header

LI	FLAGS	PAD	CLNPHDR
1	1	var	var

Figure 7-6 SP3-I. Protected Header

LI	FLAGS	PAD	IPHDR
1	1	var	var

Figure 7-7 SP3-D. Protected Header

7.2.1 LI

The length indicator field (LI) contains the length of the Protected Header in octets, excluding the length indicator field. It has a maximum value of 254 (FE hex).

7.2.2 Flags

Bit 1 of the flags field is the Initiator-to-Responder flag. Value 0 means false; value 1 means true. True indicates the PDU is sent from the Initiator to the Responder (of the creation of the key). Bits 2-8 are reserved for future use.

7.2.3 Label

The optional LABEL field is used to define a security label of a PDU. Figure 7-8 shows the LABEL field.

C0 Hex	Length	Value of LABEL
1	1	var

Figure 7-8. LABEL

The Length indicates the length of the Value of LABEL. The Value of LABEL has the following structure:

Defining Authority	Content of LABEL
1	var

Figure 7-9. Value of LABEL

The structure and interpretation of the Contents of the LABEL are defined by various Defining Authorities. The value 1 (one) indicates Defining Authority DoD. For Defining Authority DoD, the Content of LABEL has the following structure:

Basic Security Option	Extended Security Option
var	var
<---------- Appears ------ > exactly once	< ----- Appears zero ----- > or more times

Figure 7-10. Contents of DoD Security Label

The structure and interpretation of the Basic and Extended Security Options are those of the revised DoD IP Security Options. Within Content of LABEL, the Basic Security Option must appear before the Extended Security Options, if any.

7.2.4 PAD

Figure 7-11 shows the PAD field.

C1 Hex	Length	Value of PAD
1	1	var

Figure 7-11. PAD

The Length indicates the length of the Value of PAD. The value is arbitrary.

7.2.5 SRCNSAP

This field is only used in addressing mode A. Figure 7-12 shows the SRCNSAP field.

C2 Hex	Length	Value of SRCNSAP
1	1	var

Figure 7-12. SRCNSAP

The Length indicates the length of the Value of SRCNSAP. The Value of SRCNSAP is the source NSAP, expressed with the same syntax and semantics as used in the ISO CLNP protocol (ISO 8473).

7.2.6 DSTNSAP

This field is only used in addressing mode A. Figure 7-13 shows the DSTNSAP field.

C3 Hex	Length	Value of DSTNSAP
1	1	var

Figure 7-13. DSTNSAP

The Length indicates the length of the Value of DSTNSAP. The Value of DSTNSAP is the destination NSAP, expressed with the same syntax and semantics as used in the ISO CLNP protocol (ISO 8473).

7.2.7 CLNPHDR

This field is only used in addressing mode I. Figure 7-14 shows the CLNPHDR field.

C4 Hex	Length	Value of CLNPHDR
1	1	var

Figure 7-14. CLNPHDR

The length is the length of the value of CLNPHDR. The value of CLNPHDR is the encapsulated ISO CLNP header, in the format specified in ISO 8473.

7.2.8 IPHDR

This field is only used in addressing mode D. Figure 7-15 shows the format of IPHDR.

C5 Hex	Length	Value of IPHDR
1	1	var

Figure 7-15. IPHDR

The length is the length of the value of IPHDR. The value of IPHDR is the encapsulated DoD IP header, in the format specified by MIL-STD-1777.

8 Minimum Essential Requirements

Minimum essential requirements are capabilities which must be present in each SP3 implementation. The options used by SP3 are negotiated for each key, and are fixed for each key.

8.1 Addressing Modes

The addressing mode is a negotiated option. Every SP3 running at an end system must implement SP3-N and SP3-A. Other addressing modes are optional. Every SP3 running at an intermediate system must implement SP3-A. SP3-N is not suitable for use by intermediate systems. SP3-I and SP3-D are optional.

8.2 Security Services

Security services are negotiated options.

Every SP3 implementation must be capable of providing both integrity and confidentiality. Integrity-only and confidentiality-only are optional services.

Type I SP3 implementations must be capable of providing the security-label service. Labels are optional for Type II.

8.3 Padding

The use of padding is not negotiable. Every SP3 implementation must be capable of receiving and stripping the PAD option. Generation of padding is an optional capability.

9 Illustrations of SP3 Operation

This section describes the operation of SP3 using various addressing modes. SP3 is defined to provide a connectionless network service. Its request and indication parameters are consistent with that definition.

9.1 SP3 Operation at an End System

This section describes SP3 operation at an end system, using different addressing modes. The transmission functions are as described in Section 6.4, and the reception functions are as described in Section 6.5. SP3 is defined to provide a connectionless network service. Its request and indication parameters are consistent with that definition. At end systems, a transmitting SP3 always operates on NSDUs, and a receiving SP3 delivers NSDUs to the layer above it.

9.1.1 Address Mode N

This mode is identical to SP4E. It is used only from end-system to end-system. SP3N encapsulates complete NSDUs. It does not include any addressing information in the encapsulated header. Network address information passed to SP3N by the transport layer is used in selecting the appropriate key and are then passed to the underlying network communications protocol in the SP3 service request. Network address information passed to SP3N in the network indication are validated against the key-ID and are passed to the transport layer in the SP3 indication.

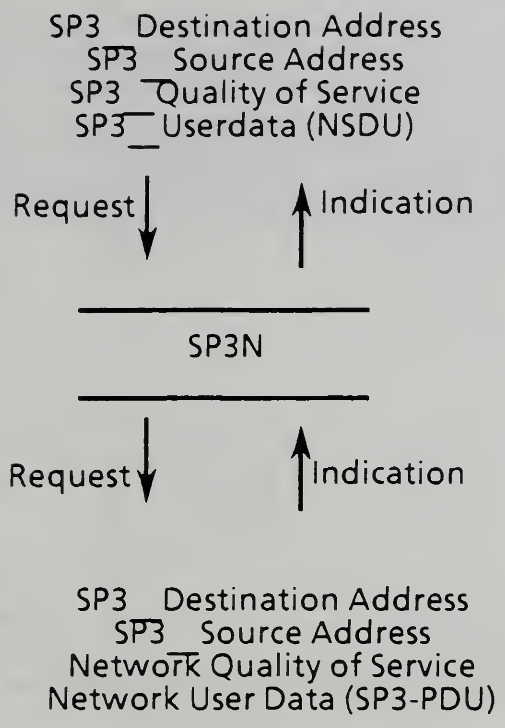
9.1.2 Address Mode A

SP3A also encapsulates complete NSDUs. The addressing information included in the encapsulated header is that passed to the source SP3 in the service request, namely source and destination end-system NSAP addresses. The source address passed in the SP3 service is the NSAP address associated with the source SP3. The destination address passed with the service request is the NSAP address associated with the destination SP3.

The network service indication passed to the receiving SP3 includes the NSAP address of the sending SP3. This is validated against the key-ID. The end-system NSAP addresses received in the encapsulated data are passed to the transport layer with the SP3 indication.

9.1.3 Address Mode 1

SP3I encapsulates complete NSDUs or fragments. The transmitting SP3I constructs a CLNP header for the entire NSDU. This header includes the source and destination NSAP addresses from the service request and other QOS parameters. If fragmentation is necessary before encapsulation, this is done. The fragments (or



complete data unit) are them encapsulated and passed to the lower layer for transmission. The source address passed in the SP3 service request is the NSAP address associated with the source SP3. The destination address passed with the service request is the NSAP address associated with the destination SP3.

The network service indication passed to the receiving SP3 includes the NSAP address of the sending SP3. This is validated against the key-ID. The key is used to remove the encapsulation. If the received data is a fragment, it must be reassembled. Complete NSDUs are passed to the higher layer, along with the end-system NSAP addresses received in the encapsulated data.

9.1.4 Address Mode D

This is identical to SP3I, except that the DoD IP format and protocol rules are used.

9.2 SP3 Operation at an Intermediate System

Parameters which are associated with each separate PDU can be carried through a network by a connectionless network protocol. These are processed by the intermediate system and used to construct the SP3 PDU. On reception at an intermediate system SP3, the information in the SP3 PDU is used to construct a PDU for the destination network.

9.2.1 Address Mode A

This mode uses protocol conversion at the intermediate system. Figure 9-1 shows the operation of SP3-A between B and C. SP3 carries explicit source and destination addresses and (optionally) security label. The network protocol carries the other communications parameters. SP3-A operates only on complete NSDUs. If the SNICP in the network between A and B fragments the PDUs, then intermediate system B must reassemble the fragments before encapsulating them. SP3-A is the intermediate system mode which is closest to being protocol-independent of the other SNICP in the network.

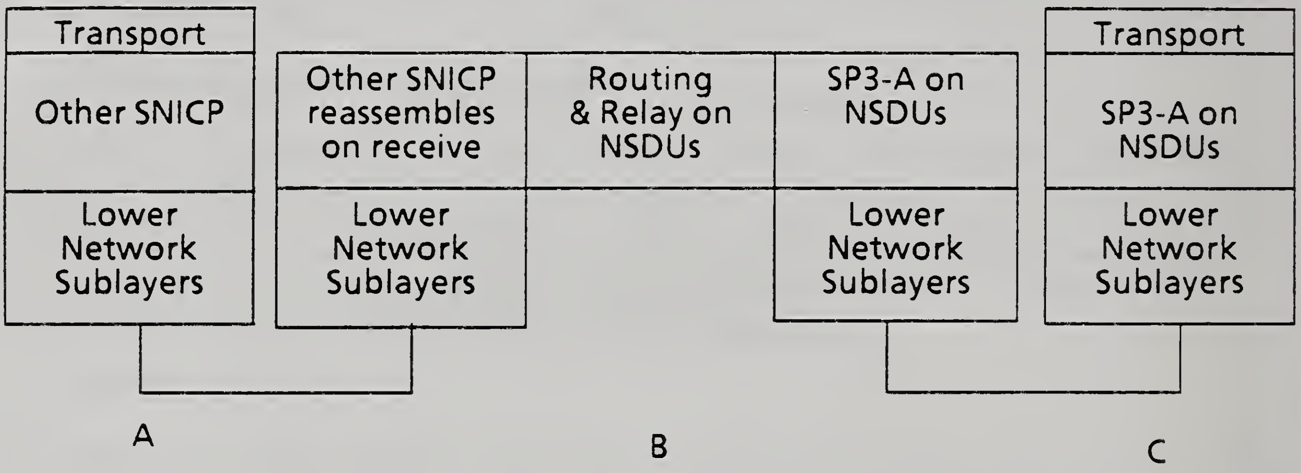


Figure 9-1. SP3-A at an intermediate system

9.2.2 Address Mode I

Address mode I uses header encapsulation at an intermediate system. Figure 9-2 shows the operation of this mode between B and C. SP3-I carries the source and destination addresses, optional security label and communications parameters in an encapsulated CLNP header. It can operate on complete NSDUs or fragments. SP3-I incorporates CLNP processing in order to operate on the CLNP header.

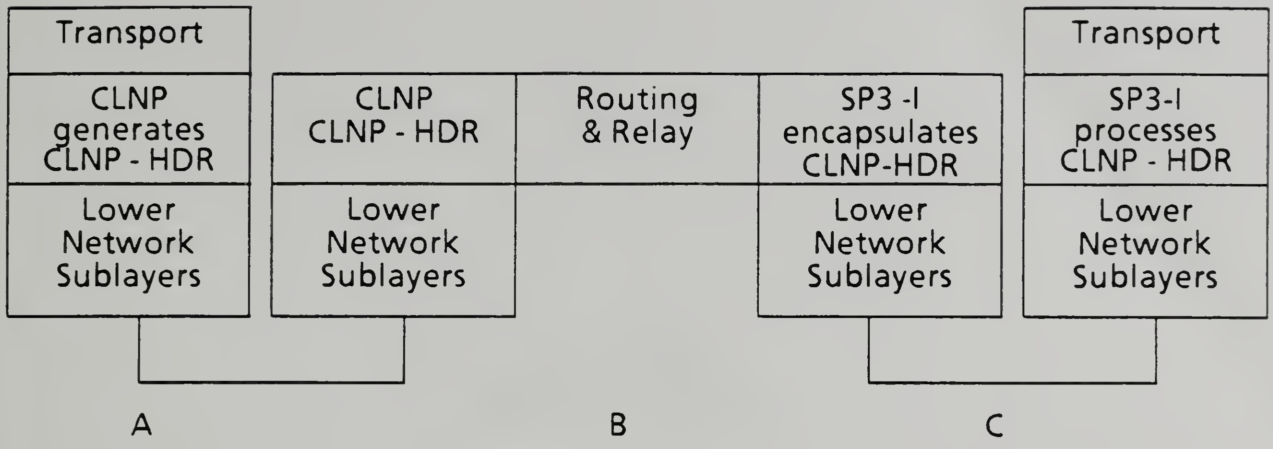


Figure 9-2. SP3-I at an Intermediate System

9.2.3 Address Mode D

This mode is similar to address mode I, except that it incorporates the DoD IP protocol instead of CLNP. Figure 9-3 shows the operation of SP3-D between B and C.

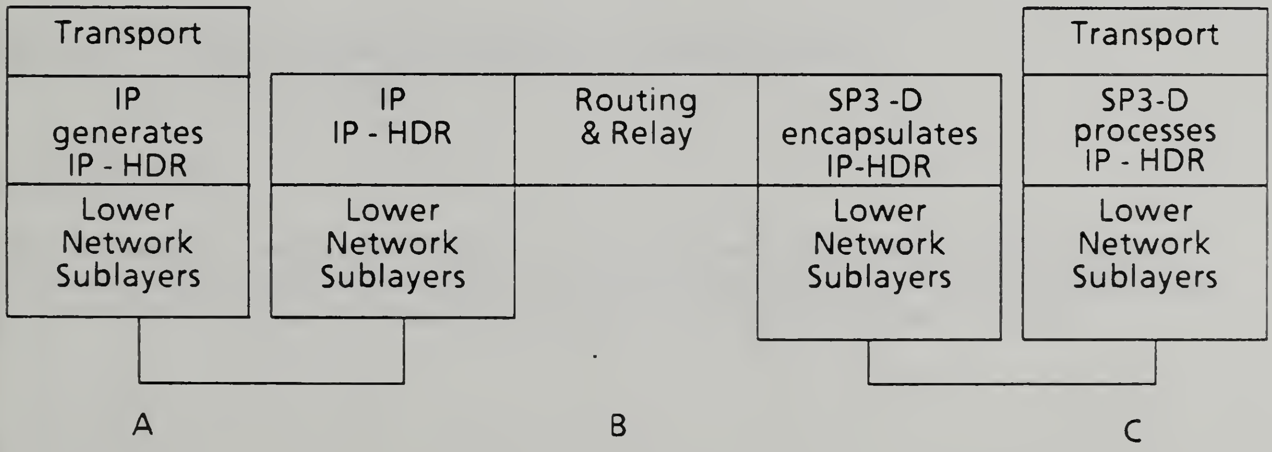


Figure 9-3. SP3-D at an Intermediate System

SPECIFICATION SDN.301.1
Revision 1.2 1988-07-12

SDNS
Secure Data
Network Systems

Security Protocol 3
(SP3)
Addendum I

Cooperating Families

Source: SDNS Protocol and Signaling Working Group
SP3 Sub-Group

Introductory note:

This document provides the framework for Addendum I of the SDNS Security Protocol at layer three (SP3). This document is being circulated for comment and approval. It is subject to change during the development phase of SDNS.

TABLE OF CONTENTS

0 INTRODUCTION. 1

1 DEFINITIONS 2

2 KEYS AND ATTRIBUTES 2

3 NETWORK REQUEST 3

COOPERATING FAMILIES

0 Introduction

This addendum describes cooperating families, an optional mechanism for providing routing flexibility in internetworks. It describes:

a. additional key attributes (referenced in SDN.301 section 6.3) and

b. modification to the processing of Section 6.4.6.

which must be implemented if the SP3 entity wishes to take advantage of a possibility that a peer entity is a member of a cooperating family.

The cooperating family mechanism allows an SP3 PDU to be received and processed by multiple SP3 entities at intermediate systems. Members of cooperating families share traffic encryption keys by a mechanism outside the scope of this document. The family may also have a mechanism to reassemble network layer fragments delivered to different members. The data sharing mechanism for fragments and the reassembly mechanisms are also outside the scope of this document.

Consider the situation of figure 0-1.

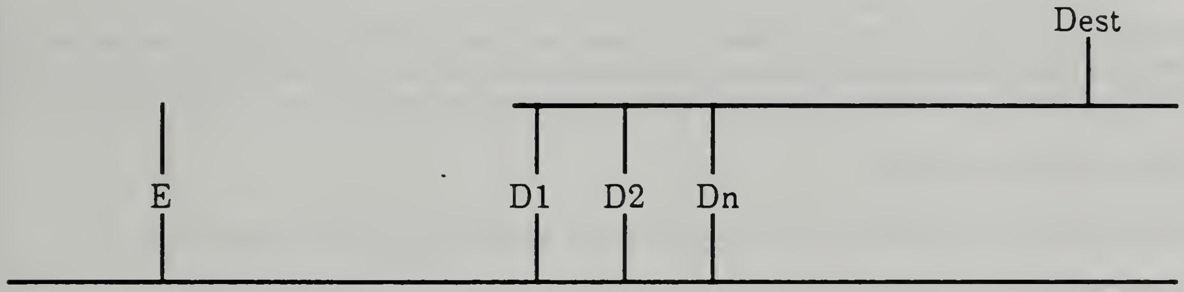


Figure 0-1

This figure shows SP3 entities E and D1, D2, ... Dn. D1, D2 ... Dn are located at intermediate systems. E may be at either an intermediate system or an end system. Consider a unit encrypted at E, whose ultimate destination is the end system Dest.

In the absence of SP3, the unit would be routed through one of the intermediate systems D1 ... Dn. If one of the Ds is unavailable, traffic will be re-routed through another in a transparent fashion.

With SP3, in the absence of the special cooperating family capabilities, E will encrypt the unit in some key which is held by some particular D, say D2. The encrypted unit must be sent to D2, since it cannot be processed by any other of the Ds.

Using cooperating family capabilities D1 ... Dn share keys via some pre-arranged mechanism such that any key held by one will be held by all. When this is the case, any one of them can decrypt a unit received by any member of the cooperating family.

The encrypted unit can be addressed (BN_Destination_Address) to any of the Ds, to Dest, or to any location served via the Ds; in fact in any way that causes it to be routed to or through one of the Ds. (After decryption, the address of Dest is recovered from the SP3 protected header.) This mechanism recovers the routing transparency that was present before the addition of SP3.

However, there is a second difficulty: Due to the nature of cryptographic algorithms. what is encrypted as one unit must be decrypted as one unit; fragments cannot be decrypted. So, suppose D1 ... Dn share keys. Suppose a unit encrypted by E is fragmented while encrypted, and the fragments are delivered to different Ds. These fragments cannot be processed. Therefore, in the absence of a special capability on the part of the Ds to handle this situation, the encryptor E must somehow arrange that all fragments of an encrypted unit arrive at the same decryptor.

The "reassemble fragments" capability on the part of the Ds means that the Ds are able to gather fragments of an encrypted unit arriving at multiple Ds together at a single one of the Ds and reassemble and process them.

The SP3 protocol does not define the mechanisms by which D1 ... Dn either share keys or reassemble fragments.

1 Definitions

A cooperating family of intermediate systems is a set of SP3 entities which share a common cryptographic key. The mechanism for sharing keys and data among cooperating family members is outside the scope of this document.

2 Keys and Attributes

The following key attributes are in addition to those listed in Section 6.3.:

1. Whether the peer SP3 entity is a member of a cooperating family.

2. Whether the cooperating family can reassemble fragments delivered to more than one member of the family.

3. NSAP addresses of the members of the family.

4. NSAP addresses which can be used to reach members of the family

In explanation of item 4: As described in the introduction, an encrypted unit to be decrypted at a cooperating family may be addressed (BN_Destination_Address) in any way that causes it to be routed to or through any member of the family. The list of 'NSAP addresses which can be used to reach members of the family' is the list of values to which BN_Destination_Addresses may be set. This may include end system addresses on networks behind the cooperating family.

NOTE: Items 3 and 4 above, plus items 8 and 9 of Section 6.3, together comprise the following list of addresses. Each of the items is included in the next with the exception of c., which contains addresses also found in b. that are not included in d.

a. The NSAP address of the peer SP3 entity

b. NSAP addresses of the members of the family

c. NSAP addresses which can be used to reach members of the family

d. All NSAP addresses served via the peer SP3 entity

3 Network Request

The following modifies the specification of the setting of BN_Destination_Address, described in section 6.4.6.

During a transmission network request function (section 6.4.6), if the peer SP3 entity is a member of a cooperating family, then:

a. BN_Destination_Address may be set to the NSAP address of the peer SP3 entity, or

b. BN_Destination_Address may be set to the NSAP address of any member of the cooperating family,

c. BN_Destination_Address may be set to any NSAP address which can be used to reach members of the family. If the cooperating family does not reassemble fragments delivered to different members of the family, then the source entity must ensure that the SP3 PDU does not become fragmented. One mechanism for doing this is to request the lower sublayer not to allow fragmentation.

SPECIFICATION SDN.401
Revision 1.3 1989-05-02

SDNS
Secure Data
Network Systems

Security Protocol 4
(SP4)

Forward:

The SDNS architecture, and its associated specifications, were developed as a cooperative project between government and industry. The project was sponsored by the National Security Agency, and supported by the National Institute of Standards and Technology (formerly the National Bureau of Standards) and the Defense Communications Agency. Twelve leading U.S. companies in computers and telecommunications made significant contributions of their technical talents and development resources. The combined efforts of these organizations produced the basis for improved security technology, interoperable security standards, and cost-effective security for computers and telecommunications.

Introductory note:

This document provides the framework for the SDNS Security Protocol at layer four (SP4). This document is being circulated for comment and approval. It is subject to change during the development phase of SDNS.

TABLE OF CONTENTS

0 INTRODUCTION 1

1 SCOPE AND FIELD OF APPLICATION 2

2 REFERENCES 3

3 DEFINITIONS 3

4 SYMBOLS AND ABBREVIATIONS 4

5 OVERVIEW OF THE PROTOCOL 5
- 5.1 Transport Security Services 5
  - 5.1.1 Connection-Oriented Security Services 7
  - 5.1.2 Connectionless Security Services 7
- 5.2 Service Assumed of the Network Layer 8
- 5.3 Service Assumed of the Key Manager 8
- 5.4 Minimum Algorithm Characteristics 8
- 5.5 Security Encapsulation Function 8
  - 5.5.1 Data Encipherment Function 8
  - 5.5.2 Integrity Function 9
  - 5.5.3 Security Label Function 9
  - 5.5.4 Security Padding Function 9

6 ELEMENTS OF PROCEDURE 9
- 6.1 Concatenation and Separation 10
- 6.2 Cryptographic Confidentiality 11
  - 6.2.1 Purpose 11
  - 6.2.2 TPDUs and parameters used 11
  - 6.2.3 Procedure 11
- 6.3 Integrity Processing 11
  - 6.3.1 Integrity Check Value (ICV) Processing 11
  - 6.3.2 Direction Indicator Processing 12
  - 6.3.3 Connection Integrity Sequence Number Processing 13
- 6.4 Peer Address Check Processing 14
- 6.5 Security Labels for Cryptographic Associations 14
  - 6.5.1 Purpose 14
  - 6.5.2 TPDUs and parameters used 14
  - 6.5.3 Procedure 15
- 6.6 Pad Parameter 15
  - 6.6.1 Purpose 15
  - 6.6.2 TPDUs and parameters used 15
  - 6.6.3 Procedure 15
- 6.7 Connection Release 15
- 6.8 Key Replacement 15

7 PROTOCOL CLASSES 16

8 STRUCTURE AND ENCODING OF TPDUS 16
8.1 Structure 16
8.2 Security Encapsulation TPDU 17
8.2.1 Clear Header 17
8.2.2 Protected Header 18
8.2.3 Data 19
8.2.4 ICV 19

0 INTRODUCTION

The transport protocol specified in International Standard (ISO) 8073 provides the connection oriented transport service described in ISO 8072. The transport protocol specified in ISO 8602 provides the connectionless-mode transport service described in ISO 8072/DAD1. This document specifies optional extensions to ISO 8073 and ISO 8602 permitting the use of cryptographic techniques to provide data protection for transport connections or for connectionless-mode TPDU transmission.

ISO 8072 describes Transport Connection (TC) protection as the prevention of unauthorized monitoring or manipulation of Transport Service (TS) user data. The TS users specify TC protection qualitatively by selecting one of four TC protection quality of service options during the TC establishment phase:

a) no protection features;
b) protection against passive monitoring;
c) protection against modification, replay, addition or deletion;
d) both b and c.

DP 7498/2 on OSI Security Architecture uses the following terms for these security services:

a) no security services;
b) connection/connectionless confidentiality;
c) connection/connectionless integrity (with or without recovery); and
d) both connection/connectionless confidentiality and integrity.

The two subclasses of SP4 are distinguished on the basis of the granularity of cryptographic associations that are established. They are:

1. SP4C: Connection oriented protection in which each transport connection is individually protected with a different cryptographic key; can provide full connection integrity and confidentiality.

2. SP4E: End system to end system protection in which all connections between a pair of end systems are protected with the same cryptographic key; can provide connectionless integrity and confidentiality.

Table 1S summarizes the connectionless and connection-oriented security services provided when SP4 is used with a connection-oriented transport protocol.Table 2S

	Connectionless - SP4E	Connection-oriented - SP4C
Confidentiality	prevent cleartext disclosure	prevent cleartext disclosure
Integrity	detect TPDU modification	detect TPDU modification, replay, addition, or deletion

Table 1S: Security Services Available in Conjunction with ISO 8073

summarizes the connectionless security services provided when SP4 is used with a connectionless transport protocol.

	Connectionless
Confidentiality	prevent cleartext disclosure
Integrity	detect TPDU modification

Table 2S: Security Services Available in Conjunction with ISO 8602

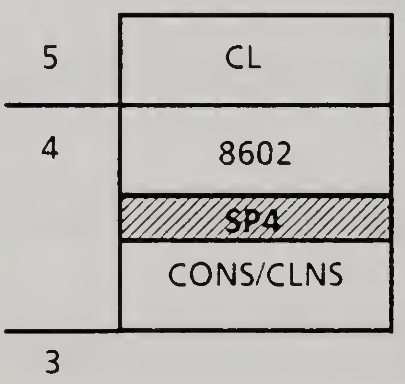
This document specifies protocol extensions for providing confidentiality and integrity data protection, including:

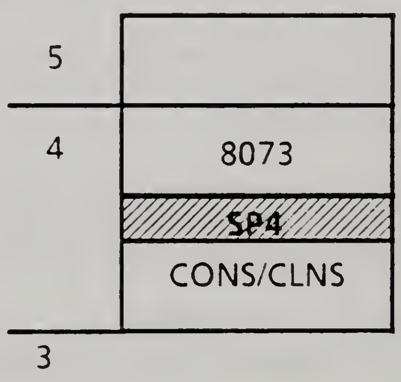
a) procedures incorporating cryptographic techniques in protocol processing,

b)the minimum characteristics of cryptographic algorithms with which these procedures can be used.

c) the structure and encoding of data units necessary to achieve interoperability.

The following figures show the location of SP4 in the seven layer ISO model.





1 SCOPE AND FIELD OF APPLICATION

The procedures specified in this document operate as extensions to those defined in ISO 8073 and ISO 8602 and do not preclude unprotected communication between transport entities implementing ISO 8073 or ISO 8602.

The degree of protection achieved will depend upon proper management of cryptographic keys. The procedures in this document assume that:

a) storage for cryptographic keys is available;

b) both the sending and receiving transport entities have the same cryptographic key available (i.e., symetric key for data protection use);

c) cryptographic keys are pairwise (i.e., shared only between two end-systems for data protection).

This document does not define how the cryptographic keys are created, updated or otherwise managed.

This protocol can support the access control service described in ISO 7498/2 using security labeling and using attributes associated with cryptographic keys. This protocol can support the peer entity authentication and data origin authentication services described in ISO 7498/2 using attributes associated with cryptographic keys.

NOTE

> The security label function provides access control enforcement (see section 6.5).

2 REFERENCES

ISO 7498	Information Processing Systems -Open Systems Interconnection - Basic Reference Model
ISO 7498/AD1	Information Processing Systems -Open Systems Interconnection - Basic Reference Model - Addendum 1: Connectionless Mode Transmission
ISO 7498-2	Information Processing Systems - Open System Interconnection - Security Architecture
ISO 8072	Information Processing Systems -Open Systems Interconnection - Transport Service Definition
ISO 8072/AD1	Information Processing Systems -Open Systems Interconnection - Addendum to the Transport Service Definition Covering Connectionless Mode Transmission
ISO 8073	Information Processing Systems -Open Systems Interconnection - Transport Protocol Specification
ISO 8602	Information Processing Systems -Open Systems Interconnection - Protocol for Providing the Connectionless-Mode Transport Service

3 DEFINITIONS

This document is based on the concepts developed in the Reference Model for Open Systems Interconnection (ISO 7498) including ISO 7498/2 on Security Architecture, and makes use of the following terms defined in the ISO 7498/2:

a) access control

b) ciphertext

c) cleartext

d) confidentiality

e) data integrity

f) data origin authentication

g) denial of service

h) end-to-end encipherment

i) key

j) key management

Additionally, this document uses the following definitions:

a) cryptographic association: a state between two entities which share a pairwise key. The entities associate common attributes with the key.

b) cryptoperiod: length of time or amount of information that a key is good.

c) pairwise key: a key generated for two specific transport entities and unavailable to any other users.

d) reflection protection: to detect that a message has been sent back.

4 SYMBOLS AND ABBREVIATIONS

This Addendum makes use of the following abbreviations from Clause 4 of ISO 8073:

CR TPDU	Connection request TPDU
DC TPDU	Disconnect confirm TPDU
DR TPDU	Disconnect request TPDU
DST-REF	Destination reference (field)
DT TPDU	Data TPDU
ED TPDU	Expedited Data TPDU
ED-TPDU-NRED	TPDU number (field)
ER TPDU	Error TPDU

LI	length indicator (field)
NSAP	Network service access point
SRC-REF	Source Reference (field)
TPDU	Transport protocol data unit
TPDU-NR	DT TPDU number (field)

Additionally, the following abbreviations are used in this Addendum:

FSN	Final Sequence Number (field)
ICV	Integrity Check Value
KEY-ID	Key Identifier
LABEL	Security Label
LME	Layer Management Entity
NSDU	Network Service Data Unit
PAD	Padding (field)
SE TPDU	Security Encapsulation TPDU

5 OVERVIEW OF THE PROTOCOL

5.1 Transport Security Services

The specific SP4 processing options used in an instance of communications are determined by the attributes associated with the pairwise cryptographic key. SP4 assumes that two transport entities using the same pairwise key will associate identical sets of attributes with that key. Thekey identifier, KEY_ID, points to the appropriate set of attributes for the pairwise key.

The following paragraphs define these attributes and list the defined mnemonics used to refer to the attributes in this specificaition. Note that the selections under each attribute are mutually exclusive - only one can be active for each cryptographic association.

- KEY GRANULARITY

A SP4 entity shall support one or more of the following key granularities:

kg_tc A separate cryptographic key is used for each transport connection

kg_esp A separate cryptographic key is used for each end system pair.

kg_esp_sr a separate cryptographic key is used for each end system pair and security level set

- CONFIDENTIALITY

The confidentiality attribute specifies whether a confidentiality service is to be provided with this cryptographic key for the cryptographic association.

conf_yes	confidentiality is to be provided
conf_no	confidentiality is not to be provided

- CONFIDENTIALITY ALGORITHM

This attribute identifies the algorithm to be used, and all of its associated parameters, if the confidentiality attribute specifies that confidentiality is to be supplied under this key (conf_yes).

- INTEGRITY

The integrity attribute specifies whether integrity services are in effect for the key.

integ_yes	integrity is to be provided
integ_no	integrity is not to be provided

- INTEGRITY ALGORITHM

This attribute identifies the algorithm to be used, and all of its associated parameters, if the integrity attribute specifies that integrity is to be supplied under this key (integ_yes).

- EXPLICIT SECURITY LABEL

This attribute specifies whether a security label is included in every TPDU exchanged on this cryptographic association. The possible values for this attribute are:

ppl_abs	Security label never used on TPDUs
ppl_xxx	xxx security label used on every TPDU

NOTE

Explicit security labels must be used when the cryptographic association supports more than one security level. They are optional when the cryptographic association supports only one security level.

- SECURITY LEVEL SET

This attribute specifies the set of allowable security levels for the cryptographic association.

- FINAL SEQUENCE NUMBER

This attribute specifies whether the final sequence number procedure (6.3.3.2) is to be used with this cryptographic association. The possible values for this attribute are:

fsn_yes	Final sequence number procedure is used
fsn_no	Final sequence number procedure is not used

- INITIATOR

This attribute specifies whether this end-system was the initiator of the cryptographic association.

- REMOTE IDENTIFIER

This attribute contains the key identifier used by the peer entity for this cryptographic association.

- PEER ADDRESS

This attribute contains the address of the peer for the cryptographic association. When the per end-system (kg_esp) or per end-system and security level (kg_esp_sr) keying is used, the address is the NSAP of the peer transport entity. When per connection (kg tc) keying is used the peer address information identifies the connection in use via the local and remote transport reference number.

5.1.1 Connection-Oriented Security Services

SP4C is used to provide connection oriented security services. The transport entity shall associate a key with each protected transport connection(kg_tc). The key shall be created explicitly for each protected transport connection. The security services to be provided on the connection are those associated with the key. All TPDUs sent or received over a protected transport connection shall be protected according to the services associated with the key. It should be noted that in this case, transport connection and cryptographic association are the same.

If connection-oriented integrity is desired, the security services associated with the key shall include Integrity Check Value (ICV) processing (integ_yes) and connection truncation protection (fsn_yes).

NOTE

> If session integrity is desired, the session entity shall not reuse transport connections.

5.1.2 Connectionless Security Services

SP4E is used to provide connectionless security services. The transport entity shall associate a key with either:

- each transport entity pair (kg_esp)
- each transport entity and security level set pair (kg_esp_sr)

The sending transport entity shall protect each TPDU according to the services associated with the key and shall place the key identifier in the KEY-ID parameter of the SE TPDU. Upon receiving an SE TPDU, the key specified in the KEY-ID parameter shall be used to decipher the TPDU set and/or to verify its ICV. Any improperly protected TPDU received shall be discarded.

5.2 Service Assumed of the Network Layer

Security services provided by the SP4 protocol are independent of any security services that may be used by the network layer.

5.3 Service Assumed of the Key Manager

The protocol specified in this addendum to the ISO 8073 requires the availability of cryptographic keys prior to an instance of protected communication. Keys are established by a combination of system, layer and security management functions. Any specific procedure for establishing keys is outside the scope of this document. The specific procedures for maintaining cryptographic key storage as well as for associating keys with specific TPDUs, are considered a local matter.

5.4 Minimum Algorithm Characteristics

Both the sending and receiving transport entities must use the same cryptographic algorithm or algorithms. The assumptions regarding cryptographic algorithms are as follows:

1) The same algorithm may be used for providing both confidentiality and integrity services.

2) It is beyond the scope of this document to specify a particular algorithm or to assess the security strengths or weaknesses of particular algorithms.

3) Encipherment and decipherment is performed in multiples of octets.

4) Cryptographic synchronization or initialization is realized on an individual TPDU basis.

5.5 Security Encapsulation Function

Encapsulation is used in conjunction with the encipherment and/or cryptographic check function to provide the connection or connectionless confidentiality and integrity services. When used by the sending entity, encapsulation is applied subsequent to all other protocol processing functions as described in ISO 8073 and ISO 8602. Further concatenation (in accordance with the concatenation rules described in 6.1) may occur after encapsulation. Decapsulation is applied by the receiving entity prior to any other protocol processing functions.

5.5.1 Data Encipherment Function

An encipherment mechanism provides data confidentiality. Each SE TPDU contains sufficient information for decipherment independent of information in any other SE TPDU. This includes identification of the cryptographic key (KEY-ID) to be used for decipherment as well as any cryptographic synchronization or algorithm initialization sequences.

5.5.2 Integrity Function

An integrity function provides data and/or data stream integrity. The elements of integrity and the mechanisms used to provide them are:

- Modification protection is provided by a ICV computed over the protected header and encapsulated TPDU.
- Insertion protection is provided by the use of the ICV and the transport sequence numbers.
- Deletion protection is provided by the use of the ICV and the transport sequence numbers.
- Connection truncation protection is provided by the transmission of final sequence numbers during connection release (fsn_yes).
- Connection replay protection is provided by the use of a separate key per transport connection (kg_tc).
- Protection against replay of a PDU is provided by the use of a separate key per transport connection (kg_tc) and the use of unique sequence numbers under each key.
- Reflection protection is provided by the use of a direction indicator (FLAGS field) in each SE TPDU (see 8.2.2.2).

5.5.3 Security Label Function

Security labeling is an optional function which can be used to associate a security label with each encapsulated TPDU set. The label indicates the sensitivity of the data. The security label supports access control mechanisms and helps meet computer security labeling requirements.

5.5.4 Security Padding Function

Security padding is an optional function which can be used to extend the length of an encapsulated TPDU set as needed. This supports cryptographic algorithm requirements.

6 ELEMENTS OF PROCEDURE

The elements of procedure are as specified in Clause 6 of the Connection-oriented Transport Protocol specification (ISO 8073) and Clause 6 of the Protocol for Providing the Connectionless-mode Transport Service (ISO 8602), with the additions in the following sections.

The protocol mechanisms described below are those used for data encapsulation. A SE TPDU contains:

a) a clear text header;

b) a protected header; if confidentiality is not used, this header is also cleartext;

c) a single TPDU or set of TPDUs concatenated according to the rules in ISO 8073;

d) an ICV parameter field, if integrity protection is used.

A TPDU shall be protected based on the attributes of the cryptographic association and encapsulated in a SE TPDU. On receipt of a SE TPDU, the transport entity shall verify that all the protection specified by the key attributes is present. An improperly protected TPDU shall be discarded.

NOTE

> This is a security relevant event and shall be reported to the layer management entity.

6.1 Concatenation and Separation

The procedure for concatenation and separation is as specified in sub-clause 6.4 of the Connection-oriented Transport Protocol specification (ISO 8073), with the following changes:

a. Concatenation may take place both prior to and subsequent to encapsulation. Any TPDU defined in ISO 8073 may be transferred after being encapsulated within an SE TPDU. Only TPDUs which are to be protected under the same cryptographic key may be concatenated.

b. If the final sequence number option is specified (fsn_yes) for the cryptographic key, at most one TPDU from the following list may be present in a set of concatenated TPDUs: DC, DR, ER.

c. An encapsulated SE TPDU may itself be concatenated according to the concatenation rules which apply to a DT TPDU type; that is, there shall be at most one SE within a set of concatenated TPDUs and, if present, it shall always be placed last in the set of concatenated TPDUs.

NOTE

> Concatenation following encapsulation is only of use when a mix of protected and unprotected transport connections are in use for the same end system.

d. A SE TPDU shall never itself be encapsulated within another SE TPDU.

NOTE

> This procedure is not used with the connectionless transport service (ISO 8602).

6.2 Cryptographic Confidentiality

6.2.1 Purpose

Cryptographic confidentiality is used in all classes of transport protocol for end-to-end protection of user and control data in transit between communicating transport entities.

6.2.2 TPDUs and parameters used

The procedure makes use of the following TPDU and parameters:

- SE TPDU;
  - key-id.

6.2.3 Procedure

If confidentiality is specified for a cryptographic association (conf_yes), then all TPDUs shall be protected by being encapsulated within an SE TPDU. All octets following the key-id (protected header and TPDU) shall be enciphered.

The cryptographic algorithm is an attribute of the cryptographic association which is identified by the key identifier (KEY-ID).

Upon receipt of a SE TPDU the transport entity uses the key identified by the key identifier in the SE TPDU to identify the security service and to decipher the TPDU. If the key is not available, the SE TPDU is discarded.

NOTE

> This is a security relevant event and shall be reported to the layer management entity.

6.3 Integrity Processing

The following procedures are used to provide connectionless and connection-oriented integrity services.

6.3.1 Integrity Check Value (ICV) Processing

6.3.1.1 Purpose

ICV processing is used in all classes to detect unauthorized modification of user and control data while in transit between communicating transport entities.

6.3.1.2 TPDUs and parameters used

The procedure makes use of the following TPDU and parameters:

- SE TPDU;
  - key-id

- ICV.

6.3.1.3 Procedure

If data integrity is specified (integ_yes) for a cryptographic association, then an ICV shall protect every SE TPDU. The message authentication code (MAC) is carried in the ICV parameter and occurs as the last field in the SE TPDU. The ICV is computed over the protected header and encapsulated TPDU. If confidentiality is specified (conf_yes) in addition to integrity, the manipulation detection code (MDC) is computed prior to encipherment.

The integrity check function and ICV field length are attributes of the cryptographic association.

Upon receiving a SE TPDU on a cryptographic association with integrity protection, the ICV field shall be verified by computing a test Integrity Check Value over the protected header and encapsulated TPDU set. If the key is not available or the test Integrity Check Value is not equal to the ICV field, then the entire SE TPDU shall be discarded.

NOTES

> This is a security relevant event and shall be reported to the layer management entity.
>
> If decipherment is also required, the testing of the Integrity Check Value shall be performed subsequent to decipherment.

6.3.2 Direction Indicator Processing

6.3.2.1 Purpose

The purpose of the direction indicator is to provide reflection protection.

6.3.2.2 TPDUs and parameters used

The procedure makes use of the following TPDU and parameters:

- SE TPDU;
  - FLAGS.

6.3.2.3 Procedure

Each SE TPDU must contain the direction indicator bit (FLAGS field) indicating the sender of the TPDU. When a SE TPDU is sent by the initiator of the cryptographic association, the direction indicator bit is set to 1. When a SE TPDU is sent by the responder of the cryptographic association, the direction indicator bit is set to 0. Upon receipt of a SE TPDU the transport entity shall validate the direction indicator bit. If a SE TPDU is received with an incorrect direction indicator the TPDU shall be discarded.

NOTE

This is a security relevant event and shall be reported to the layer management entity.

6.3.3 Connection Integrity Sequence Number Processing

Replay, insertion, and deletion detection requires that each TPDU in a cryptographic association have a unique sequence number. When connection-oriented integrity is specified for a connection (kg_tc and integ_yes), this is provided using a key per connection in conjunction with the unique sequence number procedure (6.3.3.1) and the final sequence numbers procedure (6.3.3.2).

6.3.3.1 Unique Sequence Numbers

6.3.3.1.1 Purpose

The purpose of unique sequence numbers is to uniquely identify each DT and ED TPDU within a connection.

6.3.3.1.2 Procedure

If the connection-oriented integrity service is specified for a transport connection (kg_tc and integ_yes), each TPDU shall have a unique sequence number in a CA. Neither transport entity shall transmit a new DT or ED TPDU bearing a sequence number (either TPDU-NR or ED-TPDU-NR) which was previously used with that key. Retransmissions as part of normal error control and recovery may repeat the sequence number under the original key or use a new key.When either the DT or ED sequence number space is exhausted on a particular connection, a different cryptographic key than any previously used to protect data using that connection identifier (DST-REF) shall be used for transmitting any further data TPDUs. The key replacement procedure (6.8) shall be invoked. The new key shall exist prior to this procedure being employed. If no such key exists, the connection shall be released. Upon receipt of a DT or ED TPDU which duplicates a previously received sequence number on the current cryptographic key the transport entity shall discard the TPDU.

NOTE

This procedure is not used with the connectionless transport service (ISO 8602).

6.3.3.2 Final Sequence Numbers

6.3.3.2.1 Purpose

The final sequence number is used in transport classes 2 (except when the non-use of explicit flow control option is selected), 3, and 4. Its purpose is to detect connection truncation, the deletion of the final PDUs of a connection.

6.3.3.2.2 TPDUs and parameters used

The procedure makes use of the following TPDU and parameter:

- SE TPDU encapsulating a DR, DC or ER TPDU
  - FSN.

6.3.3.2.3 Procedure

If the connection truncation protection service is specified (kg_tc and fsn_yes) for a transport connection, then the FSN field of the SE TPDU shall be included when encapsulating the DR, DC, and ER TPDUs. The sequence numbers of the final DT and ED TPDUs sent and received on the connection shall be placed in the LAST SENT and LAST RECEIVED subfields of the FSN field.

Upon receipt of a DR, DC, or ER TPDU on a connection for which the connection truncation protection service is specified, the transport entity shall compare the final sent and received sequence numbers with the sequence numbers of the final sent and received DT and ED TPDUs.

NOTE

> The handling of a mismatch of sequence numbers is a local matter. This is a security relevant event and reporting to an audit authority is recommended.

This procedure is not used with the connectionless transport service (ISO 8602).

6.4 Peer Address Check Processing

Upon receipt of a TPDU, the peer address associated with the cryptographic key shall be compared to the source address of the TPDU. If the addresses do not match, the SE TPDU shall be discarded.

NOTE

> This is a security relevant event and shall be reported to the layer management entity.

6.5 Security Labels for Cryptographic Associations

6.5.1 Purpose

Security labels are used in all classes to provide support for access control and to provide support for data separation based on sensitivity.

6.5.2 TPDUs and parameters used

The procedure makes use of the following TPDU and parameter:

- SE TPDU;
  - key-id
  - label.

6.5.3 Procedure

When a cryptographic asociation specifies use of an explicit security label on every TPDU, the label shall be sent in the LABEL field of the protected header of each SE TPDU. Upon receipt of a SE TPDU containing the LABEL parameter, the transport entity shall verify that the LABEL parameter falls within the set of acceptable security levels for the cryptographic association. If a SE TPDU is received with an improper LABEL, the TPDU shall be discarded.

NOTE

> This is a security relevant event and shall be reported to the layer management.

6.6 Pad Parameter

6.6.1 Purpose

Pad parameter processing is used in all classes for cryptographic algorithms which process data in blocks of specific sizes.

6.6.2 TPDUs and parameters used

The procedure makes use of the following TPDU and parameter:

- SE TPDU encapsulating any TPDUs defined in ISO 8073 or ISO 8602;
  - pad.

6.6.3 Procedure

A received pad parameter value is discarded.

6.7 Connection Release

If the connection-oriented service (kg_tc) is in use, the key associated with a connection shall be deleted as part of the connection release procedure.

6.8 Key Replacement

The key replacement procedure is used if the cryptoperiod of a key expires. When the connection-oriented service is in use (kg_tc) it is also used when the sequence number spaces have been exhausted (see section 6.3.3.1).

Key replacement associates a new cryptographic key with an ongoing transport connection. The new key shall exist prior to key replacement and shall have attributes which are identical to the old key. If no such key exists, the layer management entity shall be notified and the original cryptographic key shall be discarded.

Following a key replacement, unacknowledged DT and ED TPDUs requiring retransmission shall be sent under the new key.

7 PROTOCOL CLASSES

Table 6 gives an overview of which elements of procedure are included in each class. This table applies if cryptographic protection is implemented. This table is part of this addendum to the international standard.

NOTE

The key to Table 6 in ISO 8073 applies. Part of this table is reproduced below, with the additions applicable to the addendumto IS 8073.

KEY TO TABLE 6

* Procedure always included in class

Not applicable

m Negotiable procedure whose implementation in equipment is mandatory

o Negotiable procedure whose implementation in equipment is optional

Protocol mechanism	Reference	0	1	2	3	4	CLTS
Cryptographic Confidentiality	6.2	m	m	m	m	m	m
ICV Processing	6.3.1	m	m	m	m	m	m
Direction Indicator Processing	6.3.2	*	*	*	*	*	*
Unique Sequence Nos.	6.3.3.1			o	o	o	
Final Sequence Nos.	6.3.3.2			o	o	o	
Peer Address Check Processing	6.4	*	*	*	*	*	*
Security Labels for Cryptographic Associations	6.5	o	o	o	o	o	o
Pad Parameter	6.6	*	*	*	*	*	*
Connection Release	6.7			o	o	o	
Key Replacement	6.8	m	m	m	m	m	m

TABLE 6: SP4 Elements of Procedure

8 STRUCTURE AND ENCODING OF TPDUS

8.1 Structure

The structure is defined in Section 13.2 of ISO 8073.

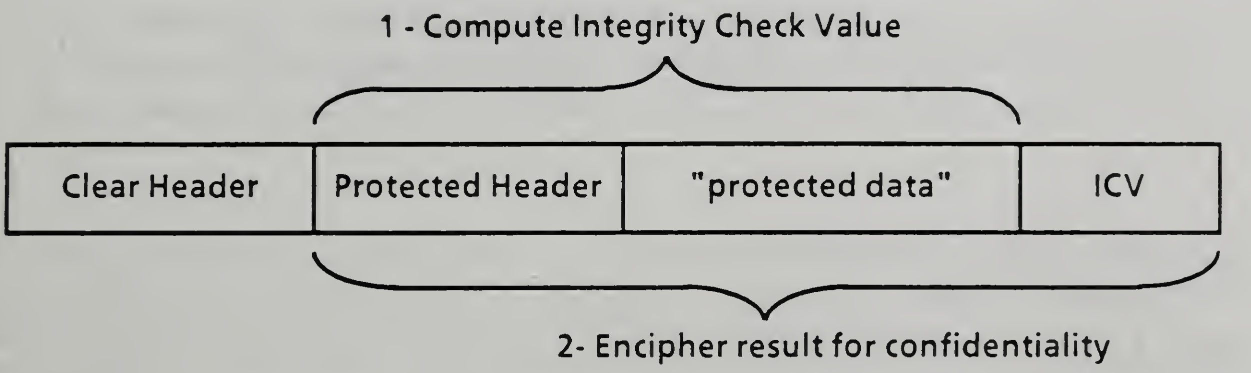
All the transport protocol data units (TPDUs) shall contain an integral number of octets. The octets in a TPDU are numbered starting from 1 and increasing in the order they are put into an NSDU. The bits in an octet are numbered from 1 to 8, where bit 1 is the low-order bit.

When consecutive octets are used to represent a binary number, the lower octet number has the most significant value.

For each fixed length field the number of octets for the field is listed below the field in the following figures.

8.2 Security Encapsulation TPDU

The structure of the SE TPDU shall be as follows:


1 - Compute Integrity Check Value
Clear Header
Protected Header
"protected data"
ICV
2- Encipher result for confidentiality

8.2.1 Clear Header

LI	SE	KEY-ID
1	1	var

8.2.1.1 LI

The length indicator field (LI) contains the length of the Clear Header in octets, excluding the length indicator field.

8.2.1.2 SE

This field contains the TPDU code. It is used to define the structure of the remaining header. The value of the SE TPDU code is: 0100 1000.

8.2.1.3 KEY-ID

The key identifier field (KEY-ID) identifies the cryptographic key used to protect the TPDU.

8.2.2 Protected Header

LI	FLAGS	LABEL	FSN	PAD
1	1	var	12	var

The LABEL, FSN, and PAD fields are optional. When present, they must appear in the order shown above.

8.2.2.1 LI

The LI contains the length of the Protected Header in octets, excluding the LI field. It has a maximum value of 254 (1111 1110).

8.2.2.2 FLAGS

Value
1

The currently defined bits in this field are:

- bit 1 direction indicator
  0 = responder to initiator
  1 = initiator to reponder

8.2.2.3 LABEL

C0 Hex	Length	Defining Authority	Value
1	1	1	var

The format of the Value is defined by the Defining Authority.

8.2.2.4 FSN

C6 Hex	Length	Last DT Sent	Last DT Received	Last ED Sent	Last ED Received
1	1	4	4	4	4

Last DT Sent is the last sequence number sent in a DT-TPDU. Last DT Received is the last sequence number received in a DT-TPDU. Last ED Sent is the last sequence number sent in a ED-TPDU. Last ED Received is the last sequence number received in a ED-TPDU.

8.2.2.5 PAD

C1 Hex	Length	Value
1	1	var

The Value field contains arbitrary data.

8.2.3 Protected Data

The data field contains a TPDU or concatenated set of TPDUs as per ISO 8073 or ISO 8602.

8.2.4 ICV

The ICV field contains the Integrity Check Value.

SDNS
Secure Data
Network System

Message Security Protocol

Forward:

The SDNS architecture, and its associated specifications, were developed as a cooperative project between government and industry. The project was sponsored by the National Security Agency, and supported by the National Institute of Standards and Technology (formerly the National Bureau of Standards) and the Defense Communications Agency. Twelve leading U.S. companies in computers and telecommunications made significant contributions of their technical talents and development resources. The combined efforts of these organizations produced the basis for improved security technology, interoperable security standards, and cost-effective security for computers and telecommunications.

Introductory note:

This document provides a framework for the SDNS Message Security Protocol. This document is being circulated for comment and approval. It is subject to change during the development phase of SDNS.

Table of Contents

0. Introduction 1

1. Scope and Field of Application 1

2. References 1

3. Definitions 2
   - 3.1 Open System Interconnection 2
   - 3.2 Message Handling System 2
   - 3.3 Secure Data Network System 2
   - 3.4 Message Security 3

4. Symbols and Abbreviations 3

5. Overview of the Message Security Protocol 3
   - 5.1 Access Control 4
   - 5.2 Multiple Recipients 4
   - 5.3 Signatures and Receipts 5

6. Message Security Services Elements 6
   - 6.1 User Agent Security Services 6
   - 6.2 Required Services from the Message Transfer Service 7
   - 6.3 Required Services from the Directory User Agent 8

7. Message Security Abstract Service Primitive 8
   - 7.1 Secure-message-request parameters 8
   - 7.2 Secure-message-indicate parameters 10

8. Message Security Protocol 10
   - 8.1 Message Security Heading 10
   - 8.2 Elements of the Procedure. 13

0. Introduction

The requirement for secure Electronic Mail and secure messaging has resulted in the Secure Data Network System (SDNS) architecture developed in support of a secure version of the X.400 Message Handling System. This document describes the additions to the CCITT X.400 Recommendations (either 1984 or 1988) that permit any type of message (including interpersonal messages) to be sent and received securely. ANSI has defined X.400 Message Transfer System conventions for Electronic Data Interchange (EDI). Using the ANSI conventions and the SDNS security additions, EDI messages can be exchanged securely. This protocol is known as the Message Security Protocol (MSP).

1. Scope and Field of Application

This document specifies the Message Security services and protocol that will be implemented in SDNS MHS components. The User Agent provides these services by encapsulating the message content and adding a Message Security Protocol heading before submission to the Message Transfer System. SDNS MSP is transparent to the X.400 MTS.

The Message Security Protocol provides writer to reader confidentiality, integrity, data origin authentication, non-repudiation with proof of origin, access control for message transfer, and request for a signed receipt of the received message.

2. References

X.200|ISO 7498 Information Processing Systems - Open Systems Interconnection - Basic Reference Model.

DIS 7498/2 Information Processing Systems - Open Systems Interconnection - Security Architecture.

X.208|ISO 8824 Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1).

X.209|ISO 8825 Information Processing Systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).

X.400|ISO 8505-1 Information Processing Systems - Text Communications - Message Handling: Service and System Overview.

X.411|ISO 8883-1 Information Processing Systems - Text communications - Message Handling: Message Transfer System [part 1] Abstract Service Definition and Procedures.

X.419|ISO 8505-2 Information Processing Systems - Text Communications - Message Handling: Protocol Specification.

X.420|ISO 9065	Information Processing Systems - Text Communications - Message Handling: Interpersonal Messaging System.
X.501/DIS 9594	Information Processing Systems - Open Systems Interconnect - The Directory - Models.
SDN.702	SDNS Directory Specifications for Utilization with the SDNS Message Security Protocol.
SDN.801	SDNS Access Control Concept Document.
SDN.802	SDNS Access Control Specification

3. Definitions

3.1 Open System Interconnection

This document uses the following terms contained in the Basic Reference Model for Open Systems Interconnection (IS 7498), including the Security Architecture (IS 7498/2):

Access control
Connectionless confidentiality
Connectionless integrity
Data origin authentication
Non-repudiation with proof of delivery
Non-repudiation with proof of origin

3.2 Message Handling System

This document uses the following terms contained X.400|ISO 8505-1 Information Processing Systems - Text Communications - Message Handling: Service and System Overview:

Content type
Distribution List (DL)
Interpersonal Messages (IPM)
Message Transfer Agent (MTA)
Message Transfer System (MTS)
O/R Name
P2
P3
SUBMIT
User Agent (UA)

3.3 The Directory

This document uses the following term contained in X.501/DIS 9594 Information Processing Systems - Open Systems Interconnect - The Directory - Models:

Directory Information Base (DIB)

3.4 Secure Data Network System

This document uses the following terms from the SDNS Access Control System Specification:

Key Material IDentifier (KMID)
Key Management System (KMS)

3.4 Message Security

For the purposes of this document the following definitions apply:

Directory service (DS): A directory server containing information (e.g., certificates, auxiliary vectors, and user keying material) corresponding to recipient UAs.

Message Security Protocol (MSP): The SDNS protocol for X.400 message security. MSP is a content protocol, and is implemented within the originator and recipient UAs. MSP processing occurs prior to submitting a message to the MTS and after accepting delivery of a message from the MTS. MSP provides security for a content protocol (e.g. IPM, EDI), but is independent of the content protocol.

Originator UA: The user agent process which originates a message.

Recipient UA: The user agent process which receives a message.

4. Symbols and Abbreviations

DIB	Directory Information Base
DS	Directory Service
DL	Distribution List
IPM	Interpersonal Messages
KMID	Key Material Identifier
KMS	Key Management System
MSP	Message Security Protocol
MS	Message Store
MTA	Message Transfer Agent
MTAE	Message Transfer Agent Entity
MTS	Message Transfer System
UA	User Agent
UKM	User Keying Material

5. Overview of the Protocol

The Message Security Protocol (MSP) operates on messages of any type, including, but not restricted to, Interpersonal Messages (IPMs) as defined in X.420. Thus, the term "message" is used throughout this document to refer to any content type defined for transfer by the Message Handling System, reflecting the wide applicability of this protocol. The set of security services provided by this protocol are grouped as indicated below:

- message confidentiality, integrity, data origin authentication and access control
- message non-repudiation with proof of origin
- request for a signed receipt of the received message (only available if non-repudiation with proof of origin is selected)

The protocol operates by encapsulating a message content in a security heading; these together form a new content type designated ProtectedContent. This new content carries the original message content (encrypted if message confidentiality is requested), plus various security parameters required by recipients to decrypt and/or validate the message upon receipt. Also included are parameters which specify the algorithms employed to perform encryption, integrity checking, and signature generation/validation, as appropriate for each service. Optionally, selected fields from the original message can be included in the security heading for perusal by the recipient prior to decryption. This facility is initially defined only for the P2 content type, but other content types can be added as needed. (Invocation of this facility may be restricted based on a particular security doctrine.) The resulting encapsulated content is submitted to the Message Transfer System (MTS) for further processing.

5.1 Access Control

The access control function is integrated within the SDNS MSP implementation. MSP operates in conjunction with the user agent (UA) process responsible for originating and receiving messages. It is technically feasible to reflect information resulting from the access control decision process to the user associated with the UA, whether the user is a human or another process. Access control implementation details are a local matter. It should be noted that access control applies only to messages for which confidentiality, integrity, and data origin authentication services are selected.

5.2 Multiple Recipients

The MHS is a store-and-forward message system where messages can be addressed to multiple recipients. MSP accommodates these characteristics in the distribution of keys for use by cipher functions, in the use of several distinct encryption and integrity algorithms, and in the structure of the MSP heading.

The originator of a message assigns the key to be used to protect the message content. The originator obtains, (e.g., from a directory server), the certificate and the UKM for each intended recipient. In preparing a message for submission, the originator collects a key for each recipient. The originator takes the message key, sensitivity label, a one-way hash on the message content and other security control information and encrypts it under this pairwise key to form a protected token. The originator then places the protected token for each recipient in the MSP heading. During this process, the originator performs access control checks to ensure that each recipient is authorized to receive the message. Each token is tagged by the originator for identification by the proper recipient. The originator's certificate and UKM are placed in the MSP heading to provide each recipient with the necessary data to process the message.

Upon receipt of the message, each recipient selects its token from the security heading using the tag as a search key. Each recipient then collects his pairwise key and decrypts his token to provide access to the message key, one way hash, and other security control information as described above. The security label is checked to enforce access control policy, the message key is used to decrypt the message, and the one way hash is used to verify the integrity of the message.

Figure 1 depicts the structure of a secure message containing an envelope, a security heading, and a message content.

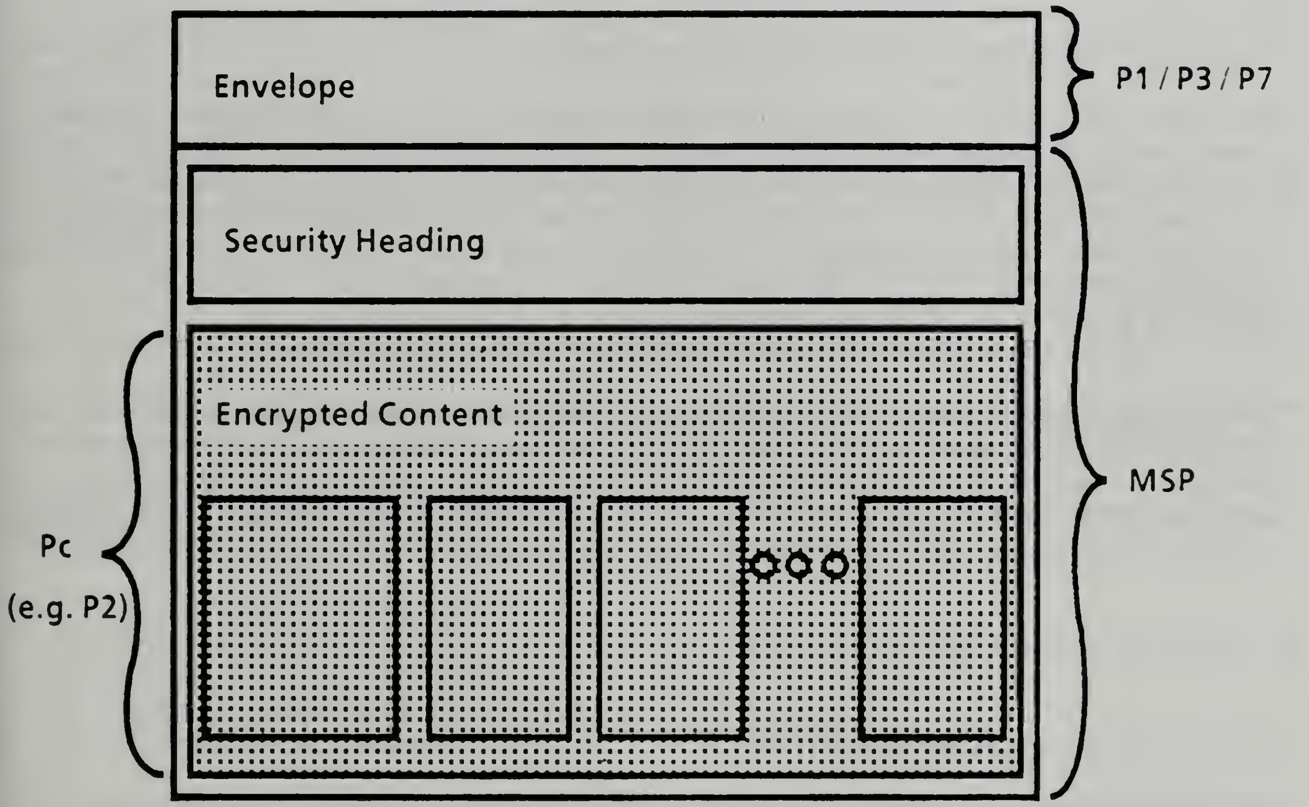


Figure 1. Secure Message Structure

5.3 Message Processing for Signatures and Signed Receipts

If the originator selects the non-repudiation with proof of origin service, a SignatureBlock is included in the MSP heading. This data item includes an identifier for the signature algorithm, the originator's signatureCertificate, controlInformation, and a signatureValue. To sign a message, the originator calculates a one-way hash on the (original) message content, as above, then continues the calculation to include the SignatureInformation. This field includes an encapsulatedContentType and a contentIdentifier. The resulting hash value is signed and included as the signatureValue in the SignatureBlock.

If the originator elects to request signed receipts from any or all of the recipients, this is indicated in the receiptsIndicator field of the SignatureInformation. If all recipients are requested to return such receipts, a flag is set; if a subset of the recipients are requested to return receipts, a list of their O/RNames is included in the field. An originator might maintain a database, (e.g., indexed by contentIdentifier), indicating the recipients from whom signed receipts for a specified message (and a copy of the message plus the hash value) are expected. This data will be required in support of later receipt processing. The database is a local matter and thus outside the scope of this protocol specification.

A recipient validates a message signature in a two-step process. First, the one-way hash on the received message content and the SignatureInformation is calculated. Then he performs calculations to verify the supplied signatureValue using the information contained in the signatureCertificate.

If the recipient has been requested to return a signed receipt, as noted above, he may generate a receipt or not. To return a receipt, the recipient constructs a new SignatureBlock. This block will contain the recipient's signatureCertificate. The receiptInformation field of the ControlInformation identifies the block as a signed receipt (versus a signed message). The original hash value is extended to include the ControlInformation for the receipt and is signed by the recipient. A message consisting solely of an MSP heading containing a version number and SignatureBlock is transmitted to the originator of the signed message.

The originator of a message requesting signed receipts will process the receipt message using the stored message hash from the original message. This value is extended to include the ControlInformation of the receipt. The computed value is compared to the verified signature value of the receipt message. If the values match, the returned SignatureBlock serves as a signed receipt for the original message.

6. Security Service Elements

6.1 User Agent Security Services

Connectionless confidentiality protects data from unauthorized disclosure. This service is provided by an encryption mechanism that is applied to the message content.

Connectionless integrity protects data from modification. This service is provided by a one-way hash applied to the plaintext message content.

Data origin authentication provides corroboration to the application process that the source of the message is the claimed originator. The key used to encrypt the message content is separately encrypted for each recipient using a token key. Successful decryption of the message key and successful verification of the one-way hash provides data origin authentication.

Access control for message transfer is a service to both originator and recipient. The originator is prohibited from submitting, and the recipient is prohibited from receiving messages that violate the security policy.

Non-repudiation with proof of origin is a service to originator and recipient. The originator is assured that the signed message did not admit tampering. The recipient of a signed message is assured that the message sent cannot be denied. A digital signature placed in the security heading of the message binds the identity of the originator to the message content. The recipient of the message can subsequently prove to a third party that only the originator could have produced the signature.

Request For Signed Receipt of the received message asks the recipient to digitally sign a hash of the message and return the signature. If the recipient performs this action and the verified receipt hash value supplied by the recipient matches that computed by the originator using the signature verification process, then the originator is provided a form of non-repudiation with proof-of-delivery service.

The Request For Signed Receipt may be used only in conjunction with the non-repudiation with proof-of-origin service. The originator of a message selects which recipients are to return a signed receipt. This list is contained within the signature block in the MSP security heading. The originator's MSP process must include the list of recipients requested to return a signed receipt in the calculation of the hash used to produce the digital signature for the non-repudiation with proof of origin service.

6.2 Required Services from the Message Transfer Service

The basic Message Transfer Service (MTS) enables a user agent (UA) to submit and receive messages. If a message cannot be delivered, the originating UA is informed.

The following Message Transfer Service (MTS) user facilities must be selected when MSP is invoked:

- Content return requested.
- Conversion prohibited.
- Physical delivery prohibited.
- Recipient redirection prohibited.
- DL expansion prohibited.

A site or community specific security doctrine may further restrict possible use of the following optional user facilities:

- Content identifier.
- Priority.
- Disclose recipients.
- Deferred delivery time.
- Latest delivery time.
- Originator report request.

6.3 Required Services from the Directory User Agent

In order to support key distribution for SDNS message security, the Directory Information Base (DIB) must store for each MSP user: certificate, a signature certificate, a set of UKMs with associated tags, and any auxiliary vectors associated with the user. The certificate, signature certificate, and any auxiliary vector(s) contain the O/R name of the user and additional identifying information. The UKMs are submitted by the user and employed in the keying process. The tags identify each UKM and a user-generated signature binds each UKM to its tag. The originator of a message accesses the directory service to obtain address information, the certificate, auxiliary vector, and tagged UKM for each recipient. {Details of these and other key management support functions provided by the directory service are contained in SDN.702 SDNS Directory Specifications for Utilization with the SDNS Message Security Protocol.}

7. Message Security Abstract Service Primitive

Secure-message is the service primitive for use with SDNS X.400 message security. This primitive permits a UA to select one or more security services:

a. Confidentiality, integrity, data origin authentication, and access control;
b. Non-repudiation with proof of origin;
c. Request for signed receipt .

Service c, Request for signed receipt, may be selected only in combination with service b, non-repudiation with proof of origin.

Secure-message service is provided by the UA. The originator of a message requests the UA to process a message content. This request is based on the arguments supplied to invoke the submit operation. The recipient of a message containing a ProtectedContent type invokes the UA to process the message content.

7.1 Secure-message-request parameters

The Secure-message-request primitive requires all of the parameters required by the SUBMIT operation (as defined in CCITT X.411). In addition, the following arguments are required:

```
ProtectionFlags
    conf-integ-doa-ac
    nonrepud
    requestForSignedReceipt
```

If the conf-integ-doa-ac flag is set then the following arguments are required:

originatorCertificate
originatorAuxVector
originatorUKM
confidentialityAlgorithm
integrityAlgorithm
tokenConfidentialityAlgorithm
tokenIntegrityAlgorithm
Sensitivity

If the nonrepud flag is set the following arguments are required:

signatureCertificate
signatureAlgorithm

If the requestForSignedReceipt flag is set, the following arguments are required:

contentIdentifier
receiptsIndicator

The ProtectionFlags are BOOLEANS that indicate which of the sets of security services the originator desires. Note that if the requestForSignedReceipt flag is set, then the nonrepud flag must be set.

The originatorCertificate is the unique identification phrase of the originator. The originatorAuxVector is additional access control information required by the security policy in force. The originatorUKM is a portion of the originator's contribution to the token key used to cover the ProtectedToken. The.signatureCertificate is used to verify the signature applied to this message.

The confidentialityAlgorithm, integrityAlgorithm,tokenConfidentialityAlgorithm, tokenIntegrityAlgorithm and signatureAlgorithm identify the algorithms used by encryption functions and include any parameters necessary for the operation of the function.The Sensitivity indicates the security level of the message.

The Secure-message-request primitive causes the UA to form a ProtectedContent. This is then input as a Content to the Message Transfer SUBMIT operation.

If an MSP implementation permits the use of blind carbon copies (bcc) and conf-integ-doa-ac is selected, then the MSP process must generate separate copies of the ProtectedContent to input to the SUBMIT operation. One copy should contain the protected tokens for all recipients not on the bcc list and should include these recipients in the SUBMIT argument list. If the identity of each bcc recipient is to be concealed from each other recipient (including other bcc recipients), then a SUBMIT operation should be invoked for each bcc recipient and the protected content should contain a protected token for only the designated recipient. This restriction is not needed if the message is only signed.

The requestForSignedReceipt flag indicates that the originator has requested one or more recipients to return a signed copy of the received message. The contentIdentifier is an octet string used to uniquely identify the message among all those submitted by the originator. The ReceiptsIndicator is either an integer with value 1, indicating that ALL recipients should be requested to return a signed receipt, a sequence of O/R names identifying the recipients who are to be explicitly requested to respond with a signed receipt, or an integer with value 0, indicating that no receipts are requested. As above, in order to provide maximum privacy in the context of blind carbon copies, multiple submissions will be required to prevent a ReceiptsIndicator sequence from disclosing identities of bcc recipients.

7.2 Secure-message-indicate parameters

The Secure-message-indicate primitive provides all of the arguments provided by the DELIVER operation as defined in (CCITT X.411). In addition, the following arguments are provided:

a. SignatureBlock
b. msgIntegrityValidity
c. sensitivity
d. tokenIntegrityValidity

The SignatureBlock is present if the originator requested the message to be signed. The SignatureBlock contains the signatureAlgorithm, the signatureCertificate, the signatureControlInformation and the signatureValue. The signatureAlgorithm identifies the algorithm. The signatureCertificate is used in the signature verification process. The signatureValue is the result of the signature function calculated for this message. The controlInformation contains the signatureType, the encapsulatedContentType, the signatureContentIdentifier, and the receiptsIndicator.

The signatureValidity, msgIntegrityValidity, and tokenIntegrityValidity arguments are BOOLEANS that indicate the success or failure of the signature, message integrity, and token integrity calculations respecitvely. The Sensitivity argument specifies the security labeling of this message.

8. Message Security Protocol

The confidentiality, integrity, data origin authentication, access control, non-repudiation with proof of origin, and the request for signed receipt services are provided within the User Agent (UA) through the addition of a security heading. The content provided by the UA and the security heading constitute a new content type.

8.1 Message Security Heading

The ProtectedContent is a sequence of the version, OriginatorSecurityData, SignatureBlock, recipient-security-data, bypassed-content-data, and encapsulatedContent. The bypassed-content-data is content dependent data that is unprotected and intended for the recipient to use prior to the invocation of security. processing. The encapsulatedContent is the original message content, after the UA applies the security services, (e.g. the encrypted content if confidentiality has been requested).

OriginatorSecurityData comprises the originator's originatorCertificate, originatorUKM, originatorAuxVector, confidentialityAlgorithm, integrityAlgorithm, tokenConfidentialityAlgorithm, and tokenIntegrityAlgorithm. The originatorCertificate is the unique identification phrase of the originator. The originatorUKM is the originator key material. The originatorUKM and originatorCertificate are combined with the recipient's posted (to a directory service) UKM and certificate to form the key used to protect the ProtectedToken. The confidentialityAlgorithm, integrityAlgorithm, tokenConfidentialityAlgorithm, and tokenIntegrityAlgorithm identify the algorithms used by encryption functions and include any parameters necessary for the operation of the function.

The SignatureBlock contains the digital signature used for non-repudiation with proof of origin. The SignatureBlock consists of the user's signatureAlgorithm, signatureCertificate (as presented in the Directory), the signatureValue, and the signatureControlInformation.

The PerRecipientToken comprises the recipient's Tag and ProtectedToken.

SignatureControlInformation comprises the signatureType, the encapsulatedContentType, the signedContentIdentifier, and the ReceiptsIndicator. The signatureType contains a zero (0) if the ProtectedContent is a signed message (signedMsg), a one (1) if the ProtectedContent is a signed message with signed receipts requested (sMsgReqRecp), and a two (2) if the ProtectedContent is a signed receipt (signedReceipt). The encapsulatedContentType is the ContentType of the original message being protected. The signedContentIdentifier identifies the message among all those submitted by the originator.

The ReceiptsIndicator is either an integer or a sequence of O/RNames. If ReceiptsIndicator is an integer, then a value of zero (0) indicates no receipts are requested (AllOrNone), and a value of one (1) indicates all recipients must return a receipt (ReceiptList). If ReceiptsIndicator is a sequence of O/RNames then the originator requests a signed receipt from each recipient listed.

The Tag is the identifier associated with the recipient's posted certificate and UKM, and includes the edition and effective period of the certificate and UKM. The Tag is not encrypted.

The ProtectedToken comprises the msgKey, msgHash, Sensitivity, encapsulatedContentType, signatureBlockIndicator and token-integrity-check. The msgKey is the key used by the encryption function identified by the confidentialityAlgorithm to encrypt the message body. The msgHash is the result of the checkfunction applied to the entire Content. The checkfunction is identified by the integrityAlgorithm. The Sensitivity indicates the security level of the encrypted portions of the message. The encapsulatedContentType identifies the type of the original content before MSP processing. The signatureBlockIndicator is a flag indicating to the recipient that the originator has requested a signature block. The token-integrity-check is the result of an integrity function applied to the concatenation of the msgKey, msgHash, sensitivity, encapsulatedContentType, and signatureBlockIndicator. The integrity function is identified by the tokenIntegrityAlgorithm. The Protected Token is encrypted using the token key.

```
MSP DEFINITIONS   ::=
      BEGIN
      -- the following types must be imported from other modules
      -- Content
      -- AlgorithmIdentifier
      -- ContentType
      -- ORName

ProtectedContent   ::= [48] SEQUENCE OF {
       version                        [0] IMPLICIT INTEGER,
       originatorSecurityData         [1] IMPLICIT OriginatorSecurityData
                                                     OPTIONAL,
       signatureBlock                 [2] IMPLICIT SignatureBlock
                                                     OPTIONAL,
       recipient-security-data        [3] IMPLICIT SET PerRecipientToken
                                                     OPTIONAL,
       bypassed-content-data          [4] IMPLICIT ANY OPTIONAL,
       encapsulatedContent            [5] IMPLICIT Content OCTETSTRING   }

OriginatorSecurityData   ::= SET {
      originatorCertificate           [0]  IMPLICIT OCTETSTRING,
      originatorUKM                   [1]  IMPLICIT OCTETSTRING,
      originatorAuxVector             [2]  IMPLICIT OCTETSTRING
                                                     OPTIONAL,
      confidentialityAlgorithm        [3]  IMPLICIT AlgorithmIdentifier,
      integrityAlgorithm              [4]  IMPLICIT AlgorithmIdentifier,
      tokenConfidentialityAlgorithm   [5]  IMPLICIT AlgorithmIdentifier,
      tokenIntegrityAlgorithm         [6]  IMPLICIT AlgorithmIdentifier   }

SignatureBlock ::= SET {
      signatureAlgorithm              [0]  IMPLICIT AlorithmIdentifer,
      signatureCertificate            [1]  IMPLICIT OCTETSTRING,
      signatureValue                  [2]  IMPLICIT OCTETSTRING,
      controlInformation              [3]  IMPLICIT ControlInformation }

ControlInformation::= CHOICE {
      signatureInformation            [0]  IMPLICIT SignatureInformation,
      receiptInformation              [1]  IMPLICIT ReceiptInformation }

PerRecipientToken ::= SET {
      tag                             [0]  IMPLICIT  Tag,
      protectedToken                  [1]  IMPLICIT  ProtectedToken }

SignatureInformation  ::= SEQUENCE OF {
      encapsulatedContentType         [0]  IMPLICIT ContentType OPTIONAL,
      signedContentIdentifier         [1]  IMPLICIT OCTET STRING,
      receiptsIndicator               [2]          ReceiptsIndicator OPTIONAL  }
```

```
ReceiptsIndicator ::= CHOICE {
      allOrNone                      [0]  IMPLICIT AllOrNone,
      receiptList                    [1]  IMPLICIT  ReceiptList }

AllOrNone ::= INTEGER {
                                     noReceipt (0),
                                     allReceipt (1)  }

ReceiptList ::=  SEQUENCE OF  ORName

ReceiptInformation  ::= SEQUENCE OF  {
      encapsulatedContentType        [0]  IMPLICIT ContentType OPTIONAL,
      signedContentIdentifier        [1]  IMPLICIT OCTET STRING,
      signatureValue                 [2]  OCTETSTRING    }

Tag  ::= SEQUENCE {
      kmid                           [0]  OCTETSTRING,
      edition                        [1]  INTEGER
      dateString                          UTCTime    }

ProtectedToken ::= SET {
      msgKey                         [0]  IMPLICIT OCTETSTRING,
      msgHash                        [1]  IMPLICIT OCTETSTRING,
      sensitivity                    [2]           Sensitivity,
      encapsulatedContentType        [3]  IMPLICIT ContentType,
      signatureBlockIndicator        [4]  IMPLICIT BOOLEAN,
      token-integrity-check          [5]  IMPLICIT OCTETSTRING  }

Sensitivity  ::= CHOICE {
      ccitt                          [0]  Ccitt,
      dod                            [1]  DoD }

Ccitt  ::= INTEGER {
                                     personal  (0),
                                     private   (1),
                                     companyConfidential  (2) }


DoD ::= INTEGER {
                                     unclassified(85),     -- 0101 0101
                                     confidential(122),    -- 0111 1010
                                     secret(173),          -- 1010 1101
                                     top-secret(222)       -- 1101 1110
                                        }

END  -- of MSP
```

8.2 Elements of the Procedure

8.2.1 Originating a Secure Message

8.2.1.1 Secure-message-request primitive issued

The originator UA presents, for MSP processing, a message content that is accompanied (implicitly or explicitly) by submission envelope information. If the UA and MTA reside in the same system, an explicit submission envelope may not be employed but equivalent information will be present as the message is transferred between the UA and the MTA. From the submit envelope, MSP processing uses the following data items:

- originator O/R name (if not implicitly provided)
- recipient O/R name list
- sensitivity
- ContentType designation for the message
- message Content

Message security label information is determined in one of two ways: (implicitly) based on local processing context, or (explicitly) based on the Sensitivity parameter. The UA may require some or all of the offered security services to be invoked for every message, or may allow a subset from this list.

8.2.1.2 Calculate Hash values

If the conf-integ-doa-ac flag is set, a msgHash is calculated. If the msgHash algorithm requires a key, the msgKey is used.

8.2.1.3 Sign message

If the nonrepud flag is set and the requestForSignedReceipt is not set, the signatureType is set to signedMsg (0) indicating that a signature is present. If the nonrepud flag is set and the requestForSignedReceipt flag is also set, signatureType is set to sMsgReqRecp (1) indicating that a signed message with a request for signed receipt is present. When the requestForSignedReceipt flag is set, the receiptsIndicator and contentIdentifier are included in the SignatureInformation. In either case, a signature hash is calculated over the message content and the SignatureInformation. This signature hash is called the signatureValue. The signatureAlgorithm, the signatureCertificate, the signatureValue, and the SignatureInformation form the SignatureBlock, which is included in the MSP message.

If the requestForSignedReceipt is set, a local database entry is contructed which includes the contentIdentifier, signatureValue, receiptsIndicator, and the message content. This database is used for later processing of signed receipts.

8.2.1.4 Encipher content

If the conf-integ-doa-ac flag is set, a message encryption key (msgKey) is generated and used to encipher the submitted message content.

8.2.1.5 Calculate token-integrity-check

If the conf-integ-doa-ac flag is set, the msgKey, the msgHash, the Sensitivity, the encapsulatedContentType, and the signatureBlockIndicator fields are concatenated and a token-integrity-check is calculated. The result is then appended to these fields all of which is referred to as the Token for this message.

8.2.1.6 Produce ProtectedToken

For each recipient in turn, the Token is encrypted using the token key. The recipient's key and the UKM associated with the encryption of the Token for a given recipient is the one which the recipient posted to the Directory System and which is designated by the Tag as being valid for the time at which the message processing began.

8.2.1.7 Produce PerRecipientToken

Each resulting ProtectedToken from step 6 is paired with the corresponding Tag of the intended recipient (the Tag is associated with the recipient's posted certificate and UKM to identify the UKM employed). The resulting set of data items is incorporated into the recipient-security-data contained in the MSP secure heading.

8.2.1.8 Place bypass data in the security heading

If the submitting entity designated any data to be bypassed, and if the implementation permits such bypass, this data is included in the MSP message. The content type of the encapsulated message is also included in the security heading. Since the data to be bypassed may differ as a function of the encapsulated content type, the recipient must examine the field which specifies the encapsulated content type to determine how to interpret this bypassed data. The bypassed data shall be represented in a fashion so that for a specified content type any intended recipient will be capable of interpreting the bypassed data.

8.2.1.9 Submit the secure message

The newly formed message is submitted to the MTA with the (validated) recipient list, SecureMessage ContentType (= 48), ConversionProhibited flag, and AlternateRecipientProhibited flag set.

8.2.2 Receiving a Secure Message

8.2.2.1 Recipient UA accepts delivery of message

The recipient requests the UA to accept delivery of a message from an MTA, or the UA retrieves a message from an MS. ContentType is 48, therefore MSP processing commences. If the submitting entity designated any data to be bypassed, and if the implementation permits such bypass, this data is included in the MSP heading. The content type of the encapsulated message is also included in the security heading. Since the data to be bypassed may differ as a function of the encapsulated content type, the recipient must examine the field which specifies the encapsulated content type to determine how to interpret this bypassed data. The bypassed data shall be represented in a fashion so that for a specified content type the recipient can interpret the bypassed data.

8.2.2.2 Select correct PerRecipientToken

MSP process identifies the correct PerRecipientToken for this recipient based on the Tag.

8.2.2.3 Decrypt ProtectedToken

The MSP process uses the token key to decrypt the ProtectedToken.

8.2.2.4 Verify token-integrity-check

A token-integrity-check is calculated and compared to the originator's token-integrity-check value. If the compare is unequal the recipient is notified (through local means) that the message has been modified.

If the signatureBlockIndicator is true, then the signature block should be present. If it should be and isn't then the originator should be notified that the message has been modified.

8.2.2.5 Decrypt content

If the conf-integ-doa-ac flag is set the content is decrypted using the msgKey.

8.2.2.6 Verify content integrity

The msgHash is calculated over the message content. If the calculated value does not equal the originator's msgHash, the recipient is notified through local means that the message has been modified.

8.2.2.7 Verify Signature

If the SignatureBlock is present in the security heading, a recipient validates the message signature in a two-step process. First, he calculates the one-way hash on the received message content and the SignatureControlInformation. Then he performs calculations to verify the supplied signatureValue using the information contained in the signatureCertificate.

If the signatureType indicates that a receipt is requested and the ReceiptsIndicator indicates that a return receipt is requested from all recipients or if it indicates selective receipting and this recipient's O/R name appears on the receiptList, the recipient is notified through local means of the request for signed receipt. Processing to generate a signed receipt is described in section 8.2.3.

If the signatureType indicates that this message is a signed receipt, the signatureValue is calculated using the supplied signatureCertificate. The contentIdentifier is used to search a local database to retrieve the signatureValue associated with the original message. A hash is calculated using this value as the initialization value and continuing over the ControlInformation in this receipt. If this value matches that from the receipt the receipt is deemed valid and the local database is updated accordingly, otherwise, the recipient is notified that receipt validation failed.

8.2.2.8 Deliver Secure Message

The Secure-message indication occurs and the arguments of the primitive, as specified in section 7.2, are supplied to the UA.

8.2.3 Generating a Signed Receipt

If, as described in section 8.2.2.7, a signed receipt is requested by the originator and the recipient elects to comply with the request, a SignatureBlock is constructed. The SignatureBlock contains SignatureCertificate of this recipient and the SignatureInformation specifies a signed receipt as the signatureType. That is, signatureType is set to signedReceipt (2). The signatureValue is computed by signing the signatureValue of the original message and the ControlInformation in the receipt. The resulting SignatureBlock is concatenated with the version to form an MSP message and is submitted to the originator of the received, signed message.

SPECIFICATION SDN.702
Revision 1.4   1989-08-01

SDNS

Secure Data Network System

SDNS Directory Specifications for Utilization with the SDNS Message Security Protocol

Source: SDNS Protocol and Signaling Working Group

Introductory note :

This document provides the framework for the SDNS Directory Server Specification. This document is being circulated for comment and approval. It is subject to change during the development phase of SDNS.

0. Introduction

This document specifies additions to the Directory System described in the 1988 X.500 series of CCITT Recommendations to support some key management functions, both for general use by SDNS components and, in particular, for use by X.400 messages protected by SDNS Message Security Protocol (MSP). This document describes the new attribute types and object classes for inclusion in the Directory Information Base (DIB) in support of these functions. These new attributes will be manipulated using the Directory Access Protocol (DAP), but no new operations are required to manipulate these attributes. None of these attributes are interpreted by the Directory, and none are used for naming.

1. Scope and Field of Application

In order to support key distribution for X.400 messages protected by MSP, the DIB must store some attributes, which are not currently provided as Directory attributes, in Directory entries associated with mail system users .

It is anticipated that, in normal operation, a Directory User Agent (DUA) would query a Directory System Agent (DSA), using the DAP to retrieve attributes associated with one or more entries, based on asserted attribute values that are sufficient to identify the intended recipient(s).

It is assumed that both private and administrative Directories may be employed in support of SDNS electronic messages and that both Type I and Type II users will be supported.

2. References

IS 7498/1 Information Processing Systems - Open Systems Interconnection - Basic Reference Model.

IS 7498/2 Information Processing Systems - Open Systems Interconnection - Security Architecture.

CCITT X.500	The Directory - Overview of Concepts, Models, and Services.
CCITT X.501	The Directory - Models.
CCITT X.509	The Directory - Authentication Framework.
CCITT X.511	The Directory - Abstract Service Definition.
CCITT X.518	The Directory - Procedures for Distributed Operation.
CCITT X.519	The Directory - Protocol Specifications.
CCITT X.520	The Directory - Selected Attribute Types.

CCITT X.521 The Directory - Selected Object Classes.

SDN.801 SDNS Access Control Concept Document.

SDN.802 SDNS Access Control Specification.

3. Definitions and Abbreviations

This document contains terms and abbreviations defined in CCITT X.500 and ISO DIS 9594 1-8. In addition, this document contains the following:

KMID - key material identifier.

SDNS - Secure Data Network System.

Ukm - User's individual keying material.

4. Specification of Attributes

4.1 Certificate

Each of these attributes consists of a certificate (Type I or II) associated with a message system user.

```
type1Certificate         ATTRIBUTE
                            WITH ATTRIBUTE-SYNTAX
                                   OCTETSTRING
                            MATCHES FOR EQUALITY
                            :: = {sdnsAttributeType 1}

type2Certificate         ATTRIBUTE
                            WITH ATTRIBUTE-SYNTAX
                                   OCTET STRING
                            MATCHES FOR EQUALITY
                            :: = {sdnsAttributeType 2}
```

4.2 Signature Certificate

These attributes consist of a certificate which can be used to validate digital signatures formulated by a user employing a signature key (Type I or II).

```
signature1Certificate         ATTRIBUTE
                              WITH ATTRIBUTE-SYNTAX
                                     OCTET STRING
                              MATCHES FOR EQUALITY
                              :: = {sdnsAttributeType 3}
```

```
signature2Certificate              ATTRIBUTE
                                   WITH ATTRIBUTE-SYNTAX
                                         OCTET STRING
                                   MATCHES FOR EQUALITY
                                   :: = {sdnsAttributeType 4}
```

4.3 Auxiliary Vector

This attribute contains information used to support the access control and authentication information contained in the Certificate.

```
auxVector                          ATTRIBUTE
                                   WITH ATTRIBUTE-SYNTAX
                                         OCTET STRING
                                   MATCHES FOR EQUALITY
                                   :: = {sdnsAttributeType 5}
```

4.4 Ukm

These attributes consist of values, each of which is a SignedUkm. The signature can be validated using the signature certificate stored with the Directory entry. The intent is that each Ukm is valid for a period of time and may be changed at the owner's discretion by modifying the appropriate attribute value. At any time, the user may delete the old attribute value; however, he should maintain the old Ukm locally for use in deciphering messages which were encrypted using the old Ukm. Users desiring to transmit secure mail should query the Directory and retrieve the appropriate SignedUkm.

```
signedUkm1                ATTRIBUTE
                             WITH ATTRIBUTE-SYNTAX
                                   SEQUENCE OF SignedUkm
                             MATCHES FOR EQUALITY
                             :: = {sdnsAttributeType 6}


signedUkm2                ATTRIBUTE
                             WITH ATTRIBUTE-SYNTAX
                                   SEQUENCE OF SignedUkm
                             MATCHES FOR EQUALITY
                             :: = {sdnsAttributeType 7}
```

5. Service Definitions

These additions to the DIB do not require any new services, but rather, specify new attributes which may be parameters to existing services as defined in X.511. Using these services, a DUA can retrieve the values of these attributes, modify the attribute values, and set the entry access control (A.C.) parameters associated with these attributes, subject to the A.C. restrictions. It is suggested that the default A.C. parameters of each of these new attributes be set to permit readItem access for all accessors, and modifyItem access for the owner(s) of each entry. Authorization to modify A.C. for these attributes should be granted to the Directory entry owner and/or system administrator.

Note: Access Control mechanisms for the Directory, as yet undefined by the International Standards, are a subject for further study.

6. Protocol Specification

These additions to the DIB require no new protocols. Rather, as in the service definitions above, the DAP is expected to operate on the newly defined attributes just as it would any other attribute. Since all of these attributes are represented as OCTET STRINGs in the Directory, no semantic processing is appropriate. Using DAP, a DUA can retrieve the values of these attributes, modify the attribute values. and set the entry A.C. parameters associated with these attributes, subject to the A.C. restrictions. It is suggested that the default A.C. parameters of each of these new attributes be set to permit readItem access for all accessors, and modifyItem access for the owner(s) of each entry. Authorization to modify A.C. for these attributes should be granted to the Directory entry owner and/or system administrator.

7. ASN.1 Notation

```
 SdnsAdditionsToDIB {48 3 9999}
 DEFINITIONS        :: =
 BEGIN

 IMPORTS

           ATTRIBUTE, ATTRIBUTE-SYNTAX, OBJECT-CLASS
              FROM InformationFramework {joint-ISO-CCITTds(5)modules(1)
                                                informationFramework(1)},


--SDNS ObjectIDs

   sdnsAttributeType  OBJECT IDENTIFIER :: = {49}

-- SDNS Attributes

type1Certificate          ATTRIBUTE
                             WITH ATTRIBUTE-SYNTAX
                                   OCTETSTRING
                             MATCHES FOR EQUALITY
                             :: = {sdnsAttributeType 1}

type2Certificate          ATTRIBUTE
                             WITH ATTRIBUTE-SYNTAX
                                   OCTET STRING
                             MATCHES FOR EQUALITY
                             :: = {sdnsAttributeType 2}

signature1Certificate        ATTRIBUTE
                             WITH ATTRIBUTE-SYNTAX
                                   OCTET STRING
                             MATCHES FOR EQUALITY
                             :: = {sdnsAttributeType 3}

signature2Certificate        ATTRIBUTE
                             WITH ATTRIBUTE-SYNTAX
                                   OCTET STRING
                             MATCHES FOR EQUALITY
                             :: = {sdnsAttributeType 4}

auxVector                    ATTRIBUTE
                             WITH ATTRIBUTE-SYNTAX
                                   OCTET STRING
                             MATCHES FOR EQUALITY
                             :: = {sdnsAttributeType 5}
```

```
signedUkm1              ATTRIBUTE
                           WITH ATTRIBUTE-SYNTAX
                                 SEQUENCE OF SignedUkm
                           MATCHES FOR EQUALITY
                           :: = {sdnsAttributeType 6}

signedUkm2              ATTRIBUTE
                           WITH ATTRIBUTE-SYNTAX
                                 SEQUENCE OF SignedUkm
                           MATCHES FOR EQUALITY
                           :: = {sdnsAttributeType 7}

SignedUkm               :: =  SEQUENCE {
                           day                INTEGER,
                           month              INTEGER,
                           year               INTEGER,
                           ukm                OCTET STRING,
                           signatureValue     OCTET STRING}

type1Sdns                  ATTRIBUTE
                           WITH ATTRIBUTE-SYNTAX
                                 SEQUENCE OF Type1Sdns
                           MATCHES FOR EQUALITY
                           :: = {sdnsAttributeType 8}

type2Sdns                  ATTRIBUTE
                           WITH ATTRIBUTE-SYNTAX
                                 SEQUENCE OF Type2Sdns
                           MATCHES FOR EQUALITY
                           :: = {sdnsAttributeType 9}

Type1Sdns                  :: =  SEQUENCE{
                           type1Certificate        OCTET STRING,
                           signature1Certificate   OCTET STRING,
                           signedUkm1            }

Type2Sdns                  :: =  SEQUENCE{
                           type2Certificate        OCTET STRING,
                           signature2Certificate   OCTET STRING,
                           signedUkm2            }


END
```

ST-114A
EV. 3-89)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NISTIR 90-4250

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
March 1990

TITLE AND SUBTITLE

AUTHOR(S)

Charles Dinkel - Editor

PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED

SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

National Security Agency - Information Security Applications Group
9800 Savage Road, SDNS SPO (C23)
Fort Meade, MD 20755-6000

SUPPLEMENTARY NOTES

□ DOCUMENT DESCRIBES A COMPUTER PROGRAM; SF-185, FIPS SOFTWARE SUMMARY, IS ATTACHED.

ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

The Secure Data Network System project, known as SDNS, implements computer to computer communications security for distributed applications. The internationally accepted Open Systems Interconnection (OSI) computer networking architecture provides the framework for SDNS. SDNS uses the layering principles of OSI to implement secure data transfers between computer nodes of local area and wide area networks. This publication includes four security protocol documents developed by the National Security Agency (NSA) as output from the SDNS project. SDN.301 provides the framework for security at layer 3 of the OSI Model. Cryptographic techniques to provide data protection for transport connections or for connectionless-mode transmission are described in SDN.401. Specifications for message security service and protocol are contained in SDN.701. Directory System Specifications for Message Security Protocol are covered in SDN.702

KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

computer security; message security; network layer security; SDNS; security protocols; transport layer security

AVAILABILITY

□ UNLIMITED
□ FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).
□ ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.
□ ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
88

15. PRICE
A05

CTRONIC FORM

IR 90-4251

RESTRICTED