IEEE Std 802.10-1998
(Revision of IEEE Std 802.10-1992,
incorporating IEEE Std 802.10b-1992,
802.10e-1993, 802.10f-1993,
802.10g-1995, and 802.10h-1997)

IEEE Standards for Local and Metropolitan Area Networks: Standard for Interoperable LAN/MAN Security (SILS)

Sponsor

LAN MAN Standards Committee
of the
IEEE Computer Society

Approved 17 September 1998

IEEE SA-Standards Board

Abstract: IEEE 802.10 provides specifications for an interoperable data link layer security protocol and associated security services. The Secure Data Exchange (SDE) protocol is supported by an application layer Key Management Protocol (KMP) that establishes security associations for SDE and other security protocols. A security label option is specified that enables rule-based access control to be implemented using the SDE protocol. A method to allow interoperability with type-encoded Medium Access Control (MAC) clients is also provided, as well as a set of managed object classes to be used in the management of the SDE sublayer and its protocol exchanges.
Keywords: decipherment; encipherment; local area networks, security; metropolitan area networks, secure data exchange; security; security association

---

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017-2394, USA

ISBN 0-7381-1419-7

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

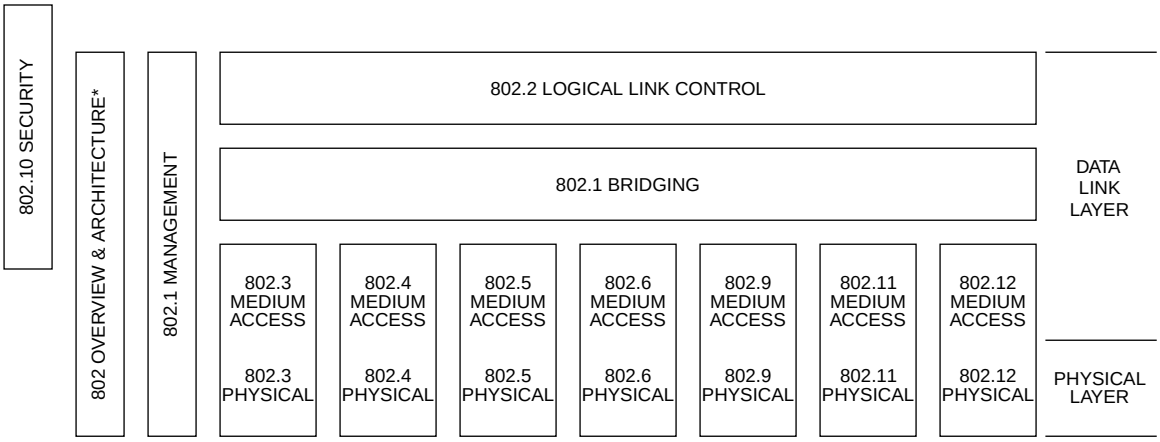
Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Introduction to IEEE Std 802.10-1998

This standard is part of a family of standards for local and metropolitan area networks. The relationship between the standard and other members of the family is shown below. (The numbers in the figure refer to IEEE standard numbers.)



* Formerly IEEE Std 802.1A.

This family of standards deals with the Physical and Data Link layers as defined by the International Organization for Standardization (ISO) Open Systems Interconnection (OSI) Basic Reference Model (ISO/IEC 7498-1 : 1994). The access standards define seven types of medium access technologies and associated physical media, each appropriate for particular applications or system objectives. Other types are under investigation.

The standards defining the technologies noted above are as follows:

- IEEE Std 802 — *Overview and Architecture.* This standard provides an overview to the family of IEEE 802 Standards.
- ANSI/IEEE Std 802.1B and 802.1k [ISO/IEC 15802-2] — *LAN/MAN Management.* Defines an OSI management-compatible architecture, and services and protocol elements for use in a LAN/MAN environment for performing remote management.
- ANSI/IEEE Std 802.1D [ISO/IEC 15802-3] — *Media Access Control (MAC) Bridges.* Specifies an architecture and protocol for the interconnection of IEEE 802 LANs below the MAC service boundary.
- ANSI/IEEE Std 802.1E [ISO/IEC 15802-4] — *System Load Protocol.* Specifies a set of services and protocol for those aspects of management concerned with the loading of systems on IEEE 802 LANs.
- ANSI/IEEE Std 802.1F — *Common Definitions and Procedures for IEEE 802 Management Information*
- ANSI/IEEE Std 802.1G [ISO/IEC 15802-5] — *Remote Media Access Control (MAC) Bridging.* Specifies extensions for the interconnection, using non-LAN communication technologies, of geographically separated IEEE 802 LANs below the level of the logical link control protocol.
- ANSI/IEEE Std 802.2 [ISO/IEC 8802-2] — *Logical Link Control*
- ANSI/IEEE Std 802.3 [ISO/IEC 8802-3] — *CSMA/CD Access Method and Physical Layer Specifications*
- ANSI/IEEE Std 802.4 [ISO/IEC 8802-4] — *Token Passing Bus Access Method and Physical Layer Specifications*

- ANSI/IEEE Std 802.5 [ISO/IEC 8802-5] — *Token Ring Access Method and Physical Layer Specifications*
- ANSI/IEEE Std 802.6 [ISO/IEC 8802-6] — *Distributed Queue Dual Bus Access Method and Physical Layer Specifications*
- ANSI/IEEE Std 802.9 [ISO/IEC 8802-9] — *Integrated Services (IS) LAN Interface at the Medium Access Control (MAC) and Physical (PHY) Layers*
- ANSI/IEEE Std 802.10 — *Interoperable LAN/MAN Security*
- ANSI/IEEE Std 802.11 [ISO/IEC DIS 8802-11] — *Wireless LAN Medium Access Control (MAC) and Physical Layer Specifications*
- ANSI/IEEE Std 802.12 [ISO/IEC 8802-12] — *Demand Priority Access Method, Physical Layer and Repeater Specifications*

In addition to the family of standards, the following is a recommended practice for a common Physical Layer technology:

- IEEE Std 802.7 — *IEEE Recommended Practice for Broadband Local Area Networks*

The following additional working group has authorized standards projects under development:

- IEEE 802.14 — *Standard Protocol for Cable-TV Based Broadband Communication Network*

Conformance test methodology

An additional standards series, identified by the number 1802, has been established to identify the conformance test methodology documents for the 802 family of standards. Thus the conformance test documents for 802.3 are numbered 1802.3.

IEEE Std 802.10-1998

The IEEE 802.10 Working Group was formed in May of 1988 to address the security LANs and MANs. It is sponsored by the IEEE LAN/MAN Standards Committee (LMSC). The working group currently has representation from vendors and users of security technology, and previously has also had representation from the government and general interest communities. The standard is an interoperability standard that is compatible with the existing IEEE 802 and OSI architectures.

Data networks, especially LANs and MANs, have become widespread. LANs and MANs are used by both industry and government for transferring vast amounts of information in the course of daily operations. Because of their ever-increasing use in the private and public sectors, the capabilities of these networks are being expanded to encompass more and more performance requirements. As a result, there is the growing need to standardize network protocols wherever feasible, to ensure that data networks will interoperate effectively.

As standardization practices evolve, several key areas will become critically important. One of these areas is network security. Many LANs and MANs require the capability to exchange data in a secure manner. This is especially important in cases where disclosure of operational information to unauthorized parties would severely undermine an organization’s effectiveness. It is often as critical to protect the integrity of the data as it is to prevent disclosure of operating information.

Financial and government institutions have traditionally been most aware of the importance of security. However, recent widely publicized cases of computer fraud and related crimes have made security a goal for many other industries as well. As the need for security on LANs and MANs gains recognition, the need for a standardized approach to providing such a capability also becomes a priority. Much security standardization has already been started. Where applicable, this standard attempts to incorporate this work.

Participants

At the time the revision of standard was completed, the 802.10 LAN/MAN Security Working Group had the following membership:

Kenneth G. Alonge, *Chair and Technical Editor* **Russell Housley,** *Vice Chair*
Joseph Maley, *Recording Secretary* **Richard McAllister,** *Executive Secretary*

The following persons were on the balloting committee:

William B. Adams
Don Aelmore
Kenneth G. Alonge
Alan Arndt
Kit Athul
Thomas W. Bailey
Kathleen L. Briggs
Peter K. Campbell
James T. Carlo
David E. Carlson
Paul Chen
John Cronican
Robert S. Crowder
Philip H. Enslow
Changxin Fan
Mark R. M. Ferguson
Harvey A. Freeman
Gautam Garai
Patrick S. Gonia
Julio Gonzalez-Sanz
Russell D. Housley
Walter K. Hurwitz
Raj Jain
Gary C. Kessler
Stephen Barton Kruger
Kenneth C. Kung
Lanse M. Leach
Pil Joong Lee
Randolph S. Little
Joseph G. Maley
Peter Martini
Bennett Meyer
David S. Millman
Warren Monroe
John E. Montague
Shimon Muller
Paul Nikolich
Robert O'Hara
Donal O'Mahony
John M. Osepchuk
Roger Pandanda
Lucy W. Person
Ronald C. Petersen
Vikram Punj
Edouard Y. Rocher
James W. Romlein
Floyd E. Ross
Christoph Ruland
Norman Schneidewind
Donald A. Sheppard
Leo Sintonen
William R. Smith
Efstathios D. Sykas
Patricia Thaler
Geoffrey O. Thompson
Mark-Rene Uchida
John Viaplana
Elfed T. Weaver
Donald F. Weir
Qian-li Yang
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 16 September 1998, it had the following membership:

Richard J. Holleman, *Chair* **Donald N. Heirman,** *Vice Chair*
Judith Gorman, *Secretary*

Satish K. Aggarwal
Clyde R. Camp
James T. Carlo
Gary R. Engmann
Harold E. Epstein
Jay Forster*
Thomas F. Garrity
Ruben D. Garzon
James H. Gurney
Jim D. Isaak
Lowell G. Johnson
Robert Kennelly
E. G. "Al" Kiener
Joseph L. Koepfinger*
Stephen R. Lambert
Jim Logothetis
Donald C. Loughry
L. Bruce McClung
Louis-François Pau
Ronald C. Petersen
Gerald H. Peterson
John B. Posey
Gary S. Robinson
Hans E. Weinrich
Donald W. Zipse

*Member Emeritus

Kristin M. Dittmann
IEEE Standards Project Editor

Contents

1. Overview and model 1

1.1 Scope and purpose 1
1.2 References 1
1.3 Definitions and acronyms 3
1.4 Architecture 9

2. Secure Data Exchange (SDE) 9

2.1 Overview 9
2.2 Definitions 10
2.3 SDE security services 10
2.4 SDE service specifications 11
2.5 SDE PDU structure 13
2.6 SDE procedure 16
2.7 Minimum Essential Requirements (MERs) 26
2.8 SDE sublayer management 27

3. Key Management 54

4. Bibliography 54

Annex 2A (informative) Service rationale 55

Annex 2B (informative) Example 65

Annex 2C (informative) Objectives of SDE 72

Annex 2D (informative) Rationale for placement 73

Annex 2E (normative) Fragmentation 77

Annex 2F (normative) ASN.1 encodings 82

Annex 2G (normative) Allocation of object identifier values 85

Annex 2H (informative) Recommended practice for SDE with IEEE 802.3 Type-encoded frames 91

Annex 2I (normative) Secure data exchange security label 94

Annex 2J (informative) Security label set registration service 102

Annex 2K (informative) Basic processing rules for security labels 103

Annex 2L (normative) Secure Data Exchange (SDE) Protocol Implementation Conformance Statement (PICS) proforma 106

IEEE Standards for Local and Metropolitan Area Networks:

Standard for Interoperable LAN/MAN Security (SILS)

1. Overview and model

1.1 Scope and purpose

This standard was developed to provide security in IEEE 802 Local Area Networks (LANs) and Metropolitan Area Networks (MANs).

The model for providing security services is described in Clause 1. A Secure Data Exchange (SDE) protocol for IEEE 802 LANs and MANs is defined in Clause 2. Key management in IEEE 802 LANs and MANs is described in Clause 3 (published separately as IEEE Std 802.10c-1998). While SDE is independent of any key management or system management implementation, the security services described in this standard depend on management information provided by management entities.

1.2 References

This standard shall be used in conjunction with the following references:

NOTE—Additional references are listed in 1.3 of IEEE Std 802.10c-1998.

ANSI X9.17-1985 (R1991), Financial Institution Key Management (Wholesale).[1]

FIPS PUB 188: 1994, Standard Security Label for Information Transfer (SSL).[2]

IEEE Std 802-1990, IEEE Standards for Local Area Networks: Overview and Architecture.[3]

---

[1]ANSI publications are available from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA (http://www.ansi.org/).

[2]FIPS publications are available from the National Technical Information Service (NTIS), 5285 Port Royal Road, Springfield, VA 22161 USA, tel. (703) 605-6000.

[3]IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA (http://standards.ieee.org/).

IEEE Std 802.1F-1993 (Reaff 1998), IEEE Standards for Local and Metropolitan Area Networks: Common Definitions and Procedures for IEEE 802 Management Information.

Internet RFC 1457: 1993, R. Housley, Security Labeling Framework for the Internet.[4]

ISO/IEC 7498-1: 1994, Information technology—Open Systems Interconnection—Basic Reference Model—Part 1: The Basic Model.[5]

ISO 7498-2: 1989, Information processing systems—Open Systems Interconnection—Basic Reference Model—Part 2: Security Architecture.

ISO/IEC 7498-4: 1989, Information processing systems—Open Systems Interconnection—Basic Reference Model—Part 4: Management framework.

ISO/IEC 8802-2: 1998 [ANSI/IEEE Std 802.2, 1998 Edition], Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 2: Logical link control.[6]

ISO/IEC 8824: 1990, Information technology—Open Systems Interconnection—Specification of Abstract Syntax Notation One (ASN.1) (Provisionally retained edition).

ISO/IEC 8825: 1990, Information technology—Open Systems Interconnection—Specification of basic encoding rules for Abstract Syntax Notation 1 (ASN.1) (Provisionally retained edition).

ISO/IEC 9595: 1991, Information technology—Open Systems Interconnection—Common management information service definition.

ISO/IEC 9596-1: 1991, Information technology—Open Systems Interconnection—Common management information protocol—Part 1: Specification.

ISO/IEC 9596-2: 1993, Information technology—Open Systems Interconnection—Common management information protocol—Part 2: Protocol Implementation Conformance Statement (PICS) proforma.

ISO/IEC 10040: 1992, Information technology—Open Systems Interconnection—Systems management overview.

ISO/IEC 10164-1: 1993, Information technology—Open Systems Interconnection—Systems Management—Part 1: Object Management Function.

ISO/IEC 10165-1: 1993, Information technology—Open Systems Interconnection—Structure of management information—Part 1: Management Information Model.

ISO/IEC 10165-2: 1992, Information technology—Open Systems Interconnection—Structure of management information—Management information—Part 2: Definition of management information.

ISO/IEC 10165-4: 1992, Information technology—Open Systems Interconnection—Structure of management information—Part 4: Guidelines for the definition of managed objects.

---

[4]Internet Requests for Comments (RFCs) are available from the DDN Network Information Center, SRI International, Menlo Park, CA 94025 USA. They are also available on the World Wide Web at the following URL: http://www.internic.net/ds/rfc-index.html.

[5]ISO and ISO/IEC publications are available from ISO, Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse (http://www.iso.ch/). ISO publications are also available in the United States from ANSI.

[6]ISO [ANSI/IEEE] publications are available from the ISO Central Secretariat, 1 rue de Varembé, Case Postale 56, CH-1211, Genève 20, Switzerland/Suisse. They are also available in the US from the IEEE.

ISO/IEC 10165-6: 1997, Information technology—Open Systems Interconnection—Structure of management information—Part 6: Requirements and guidelines for implementation conformance statement proforma associated with OSI management.

ISO/IEC 15802-1: 1995, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 1: Medium Access Control (MAC) service definition.

ISO/IEC 15802-2: 1995 [ANSI/IEEE Std 802.1B and IEEE Std 802.1k, 1995 Edition], Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 2: LAN/MAN management, service and protocol.

NISTIR 5308: 1993, N. Nazario, General Procedures for Registering Computer Security Objects.[7]

1.3 Definitions and acronyms

1.3.1 Definitions

This subclause contains the definitions that are applicable to all clauses of this standard. Sources for the definitions are indicated. When no source is indicated, this standard is the source.

1.3.1.1 access control: The prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner. (ISO/IEC 7498-2: 1989)[8]

1.3.1.2 attribute: A property of a managed object or a property of an association among OSI entities. An attribute has an associated value, which may have a simple or complex structure. (ISO/IEC 10040: 1992)

1.3.1.3 authentication: *See:* data origin authentication; peer entity authentication. *Note:* In this standard, the term “authentication” is not used in connection with data integrity; the term “data integrity” is used instead.

1.3.1.4 bootstrap SAIDs: Four SAID values that are reserved for the purpose of establishing initial communication with key management or system management when an SAID has not already been negotiated. These SAID values have a preestablished security association.

1.3.1.5 ciphertext: Data produced through the use of encipherment, the semantic content of which is not available. *Note:* Ciphertext may itself be input to encipherment, producing superenciphered data.

1.3.1.6 cleartext: Intelligible data, the semantic content of which is available. (ISO/IEC 7498-2: 1989)

1.3.1.7 compromise: A violation of the security of a system such that an unauthorized disclosure of sensitive information may have occurred. (NCSG-TG-005 Version-1 [B3][9])

1.3.1.8 computer security object: An information object used to maintain a condition of security in computerized environments. Examples include: representations of computer or communications systems resources, security label semantics, modes of operation for cryptographic algorithms, and one-way hashing functions. (NISTIR 5308: 1993)

---

[7]This publication is available from the National Technical Information Service (NTIS), U. S. Dept. of Commerce, 5285 Port Royal Rd., Springfield, VA 22161 USA.

[8] Information on references can be found in 1.2.

[9]The numbers in brackets preceded by the letter B correspond to those of the bibliography in Clause 4.

1.3.1.9 confidentiality: The property of information that is not made available or disclosed to unauthorized individuals, entities, or processes. (ISO/IEC 7498-2: 1989)

1.3.1.10 connectionless confidentiality: The protection of (N)-service data units from unauthorized disclosure during transmission from one (N+1)-entity to one or more (N+1)-entities, where each entity has an association with the physical layer, and no association is established for the transmission of data or for the application of the confidentiality service between the layer peer-entities themselves.

1.3.1.11 connectionless integrity: A service providing for the integrity of a single SDU. It may take the form of determining whether or not the received SDU has been modified.

1.3.1.12 connection-oriented confidentiality: The protection of all (N)-service data units from unauthorized disclosure during communications from one (N+1)-entity to one or more (N+1)-entities for which a security association is established for the transfer of data and for the application of confidentiality service between the entities themselves and between each entity and the physical layer.

1.3.1.13 connection-oriented integrity: A service providing for the integrity of all (N)-service data on a security association and detecting any modification, insertion, deletion, or replay of any data within an entire SDU sequence.

1.3.1.14 cryptographic checkvalue: Information that is derived by performing a cryptographic transformation on the data unit. *See:* cryptography. (ISO/IEC 7498-2: 1989)

1.3.1.15 cryptography: The discipline embodying principles, means, and methods for the transformation of data in order to hide its information content, prevent its undetected modification, and/or prevent its unauthorized use. (ISO/IEC 7498-2: 1989)

1.3.1.16 data deciphering key: A key used for the decipherment of an (N)-layer SDU. (It is not used to decipher other keys.)

1.3.1.17 data enciphering key: A key used for the encipherment of an (N)-layer SDU. (It is not used to encipher other keys.)

1.3.1.18 data integrity: The condition or state in which data has not been altered or destroyed in an unauthorized manner. (ISO/IEC 7498-2: 1989)

1.3.1.19 data origin authentication: The corroboration that the source of data received is as claimed. This service, when provided by the (N)-layer, provides the corroboration to an (N+1)-entity that the source of the data is the claimed peer (N+1)-entity. (ISO/IEC 7498-2: 1989)

1.3.1.20 decipherment: The reversal of a corresponding reversible encipherment. (ISO/IEC 7498-2: 1989)

1.3.1.21 encipherment: The cryptographic transformation of data to produce ciphertext. *See:* cryptography. (ISO/IEC 7498-2: 1989)

1.3.1.22 entity: An active element in an open system. (ISO/IEC 7498-1: 1994)

1.3.1.23 hierarchical level: A member of a linearly ordered set (i.e., hierarchy) of levels, e.g., a number in the range from 0 to 255.

1.3.1.24 information object: A well-defined piece of information, definition, or specification that requires a name to identify its use in an instance of communication. (ISO/IEC 8824: 1990)

1.3.1.25 Initialization Vector (IV): A binary vector used at the beginning of a cryptographic operation to allow cryptographic chaining. (ANSI X9.17-1985)

1.3.1.26 Integrity Check Value (ICV): A value that is derived by performing an algorithmic transformation on the data unit for which data integrity services are provided. The ICV is sent with the protected data unit and is recalculated and compared by the receiver to detect data modification. *See:* cryptographic checkvalue.

1.3.1.27 key: A sequence of symbols that controls the operations of encipherment and decipherment. (ISO/IEC 7498-2: 1989)

1.3.1.28 key management: The generation, storage, distribution, deletion, archiving, and application of keys in accordance with a security policy. (ISO/IEC 7498-2: 1989)

1.3.1.29 Key Management Stack: The protocols residing above SDE that request services via an SDE SAP that is supported by the use of a bootstrap SAID with either of the two values reserved for key management.

1.3.1.30 layer management: Functions related to the management of the (N)-layer partly performed in the (N)-layer itself according to the (N)-protocol of the layer, and partly performed as a subset of systems management. (ISO/IEC 7498-1: 1994)

1.3.1.31 Layer Manager (LM): A systems management service application for which a particular exchange of systems management information has taken a manager role of the (N)-layer. (ISO/IEC 10040: 1992)

1.3.1.32 managed object: The OSI structure of management information term used as an abstract representation of a resource. This managed object has a set of attributes. These attributes are equivalent to data objects. (ISO/IEC 10165-2: 1992)

1.3.1.33 Management Information Base (MIB): A conceptual database of information contained in the collection of all the managed object classes and their instances. (ISO/IEC 7498-4: 1989)

1.3.1.34 manipulation detection: A mechanism used to detect whether a data unit has been modified (either accidentally or intentionally). (ISO/IEC 7498-2: 1989)

1.3.1.35 masquerade: The pretense by an entity to be a different entity. (ISO/IEC 7498-2: 1989)

1.3.1.36 misordering data: A form of unauthorized data modification in which the reception sequence of data units is altered from the original transmission sequence in an unauthorized manner. This can be attempted by a combination of techniques involving deleting, delaying, and reinserting data; or modifying sequence control information; or both.

1.3.1.37 Named Tag Set: A field containing a Tag Set Name and its associated set of security tags. (FIPS PUB 188: 1994)

1.3.1.38 object: A data object that has an identifier (name) and a value.

1.3.1.39 Open Systems Interconnection (OSI) (N)-service: A capability of the (N)-layer, and the layers beneath it, that is provided to the (N)-entities at the boundary between the (N)-layer and the (N+1)-layer. (ISO/IEC 7498-1: 1994)

1.3.1.40 peer-entity authentication: The corroboration that a peer entity in an association is the one claimed. This service, when provided by the (N)-layer, provides corroboration to the (N+1)-entity that the peer entity is the claimed (N+1)-entity. (ISO/IEC 7498-2: 1989) *Note:* This is primarily intended for, although not limited to, connection-oriented service and may be either unilateral or mutual.

1.3.1.41 permissive [security] attribute: A security attribute that identifies an active entity or a resource as member of a group. An entity is granted access to all resources in the groups of which it is a member. Permissive attributes could be used alone or in combination with restrictive attributes. Commonly, when used in combination with restrictive attributes, they are secondary in the determination of access privilege.

1.3.1.42 policy: *See:* security policy.

1.3.1.43 protocol data unit: A unit of data specified in a protocol and consisting of protocol information and, possibly, user data. (ISO/IEC 7498-1: 1994)

1.3.1.44 protocol entity: An entity that follows a set of rules and formats (semantic and syntactic) that determines the communication behavior of other entities. (ISO/IEC 7498-1: 1994)

1.3.1.45 reflection: A form of data modification in which PDUs sent by an entity are returned in an unauthorized manner. This can be attempted by a combination of techniques involving deleting, delaying, and reinserting data; and/or modifying address or sequence control information.

1.3.1.46 register: A set of records (paper, electronic, or a combination) maintained by a Registration Authority containing assigned names and the associated information. (NISTIR 5308: 1993)

1.3.1.47 restrictive [security] attribute: A security attribute that indicates the minimum level of privilege required by an active entity (i.e., subject) in order to gain access to a resource (i.e., object). Commonly a set of restrictive security attributes are associated with each resource. An active entity may only gain access to a resource if its set of privileges is higher than, or a superset of (i.e., dominates), the attribute set for the resource.

1.3.1.48 secret key: The traditional cryptographic key known only to the communicating parties and used for both encipherment and decipherment.

1.3.1.49 secure data exchange (SDE) Layer Manager (LM): The SDE portion of the Layer 2 Manager.

1.3.1.50 security association: A cooperative relationship between entities formed by the sharing of cryptographic keying information and security management objects. This shared information need not be identical, but it must be compatible.

1.3.1.51 Security Association Identifier (SAID): A value placed in the clear header of the SDE PDU that is used to identify the security association.

1.3.1.52 security attribute: A security-related quality of an object. Security attributes may be represented as hierarchical levels, bits in a bit map, or numbers. Compartments, caveats, and release markings are examples of security attributes.

1.3.1.53 security label: A marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource. (ISO/IEC 7498-2: 1989)

1.3.1.54 security level: A hierarchical level whose purpose is to indicate degree of sensitivity to a designated security threat. It indicates a specific level of protection as specified by the security policy being enforced.

1.3.1.55 Security Management Information Base (SMIB): An MIB that stores security-relevant objects.

1.3.1.56 security policy: A set of criteria for the provision of security services. (ISO/IEC 7498-2: 1989)

1.3.1.57 security service: A service, provided by a layer of communicating open systems, that ensures adequate security of the systems or of data transfers. (ISO/IEC 7498-2: 1989) *Note:* These security services need not be directly requested at the (N)- and (N+1)-layer boundary as is required for an OSI (N)-service.

1.3.1.58 security tag: An information unit containing a representation of certain security-related information (e.g., a restrictive attribute bit map).

1.3.1.59 security threat: A potential violation of security. (ISO/IEC 7498-2: 1989)

1.3.1.60 System Management Stack: The protocols residing above SDE that request services via an SDE SAP that is supported by the use of a bootstrap SAID with either of the two values reserved for system management.

1.3.1.61 system management: Functions in the Application Layer related to the management of various OSI resources and their status across all layers of the OSI architecture. (ISO/IEC 7498-1: 1994)

1.3.1.62 Tag Set Name: A numeric identifier associated with a set of security tags. (FIPS PUB 188: 1994)

1.3.1.63 threat: A potential violation of security. (ISO/IEC 7498-2: 1989)

1.3.1.64 transparent: The state of a protocol when all of the following conditions are met:

a) Previously existing protocol implementations are able to recover when receiving packets formed by this new protocol.
b) The implementations of this protocol are able to process packets formed by previously existing protocols without problems.
c) The protocol does not affect the operations of the (N+1) and (N-1)-layer implementations.

1.3.1.65 trusted functionality: That which is perceived to be correct with respect to some criteria, e.g., as established by a security policy. (ISO/IEC 7498-2: 1989)

1.3.1.66 unauthorized data modification: Alteration of data not consistent with the defined security policy.

1.3.1.67 unauthorized disclosure: The process of making information available to unauthorized individuals, entities, or processes. (ISO/IEC 7498-2: 1989)

1.3.1.68 unauthorized resource use: Use of a resource not consistent with the defined security policy. (ISO/IEC 7498-2: 1989)

1.3.1.69 User Stack: The protocols residing above SDE that request services from any SDE SAP except those supported by the use of a bootstrap SAID.

1.3.2 Acronyms

This subclause contains acronyms that are applicable to the standard.

NOTE—Additional acronyms are listed in 1.2.1 of IEEE Std 802.10c-1998.

ANSI	American National Standards Institute
ASN.1	Abstract Syntax Notation One
CMIP	Common Management Information Protocol
CMIS	Common Management Information Service
CSOR	Computer Security Objects Register
DA	Destination Address
DEA	Data Encryption Algorithm

DIS	Draft International Standard
DMI	Definition of Management Information
DSAP	Destination Service Access Point
GDMO	Guidelines for the Definition of Managed Objects
ICV	Integrity Check Value
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
IV	Initialization Vector
LAN	Local Area Network
LLC	Logical Link Control
LM	Layer Manager
LME	Layer Management Entity
LMI	Layer Management Interface
LMMP	LAN/MAN Management Protocol
LMMPE	LAN/MAN Management Protocol Entity
LMMS	LAN/MAN Management Service
LMMU	LAN/MAN Management User
LSAP	Link Service Access Point
MAC	Medium Access Control
MAN	Metropolitan Area Network
MDF	Management-Defined Field
MER	Minimum Essential Requirements
MIB	Management Information Base
MIS-User	Management Information Services User
MO	Managed Object
MOCS	Management Object Conformance Statement
MSDU	MAC Service Data Unit
NM	Network Management
NMI	Network Management Interface
OID	Object Identifier
OSI	Open Systems Interconnection
OUI	Organizationally Unique Identifier
PDU	Protocol Data Unit
PE	Protocol Entity
PICS	Protocol Implementation Conformance Statement
RDN	Relative Distinguished Name
SA	Source Address
SAID	Security Association Identifier
SAP	Service Access Point
SDE	Secure Data Exchange
SDU	Service Data Unit
SILS	Standard for Interoperable LAN Security
SMAE	System Management Application Entity
SMI	Structure of Management Information
SMIB	Security Management Information Base
SNAP	Subnetwork Service Access Point
SSAP	Source Service Access Point
SSL	Standard Security Label
TCB	Trusted Computing Base

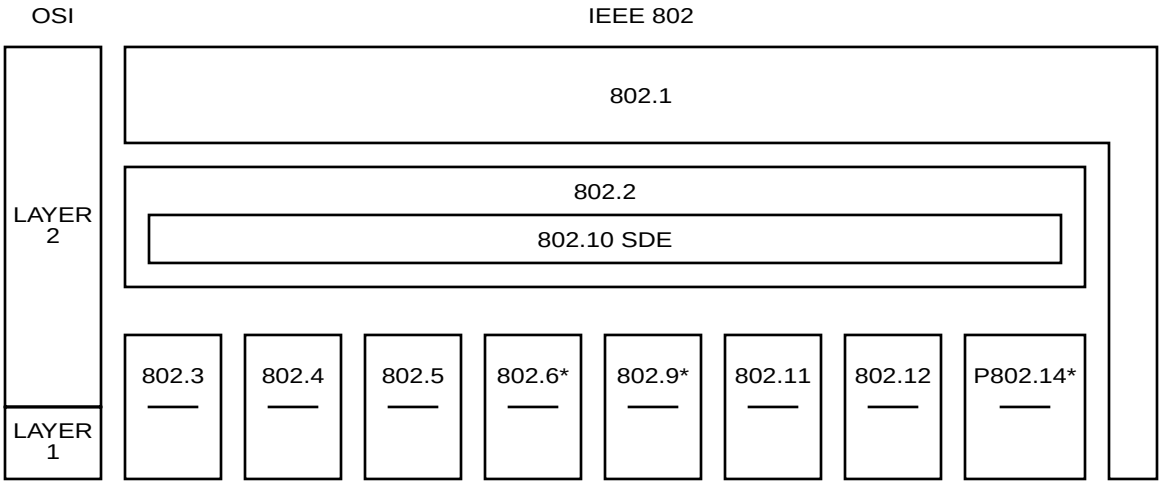
1.4 Architecture

The relationship of the each of the security protocols to the Open Systems Interconnection (OSI) Basic Reference Model will be described in this subclause, which is under consideration.[10]

2. Secure Data Exchange (SDE)

2.1 Overview

The SDE is an OSI Basic Reference Model (ISO/IEC 7498: 1984) Layer 2 entity. This entity provides services that permit the secure exchange of data at Layer 2. As part of the Logical Link Control (LLC) sublayer, the SDE entity provides a connectionless service immediately above the Medium Access Control (MAC) sublayer in IEEE 802 LANs and MANs. It provides security across the MAC sublayer using cryptographic mechanisms and security services provided transparently at the boundary to the LLC entity. The relationship of the SDE entity to the IEEE 802 reference model is shown in Figure 2-1.



*SDE is not applicable to MANs using isochronous and connection-oriented protocols.

Figure 2-1—Relationship to IEEE 802 reference model

The SDE interface services specification to the MAC sublayer, to the boundary of the LLC entity, and to the SDE layer management functions is defined in this subclause. The security services provided and the threats these services protect against are described in 2.3. In 2.4, the service specifications are defined and the interface to the MAC sublayer and to the LLC entity boundary is described in detail.

The SDE entity provides security services and an interface at the boundary to the LLC entity. However, it does not specify any of the higher protocols that reside in the User Stack, including those of the LLC sublayer. The SDE interface is equivalent to the unprotected MAC interface and thus requires no change to the existing upper-layer protocols in the User Stack.[11] SDE security services provided to a Key Management Stack or to a System Management Stack require the LLC protocol.

[10]This subclause will be included in IEEE P802.10a, which at the time of this publication was not an approved IEEE standard, but is available as a draft. For information about obtaining draft standards, contact the IEEE.

An SDE-specific Protocol Data Unit (SDE PDU) is introduced in 2.6. The SDE PDU has optional elements and fields to satisfy a broad range of potential security applications. A reserved Link Service Access Point (LSAP) in the clear header portion of the SDE PDU distinguishes the SDE PDU from LLC PDUs. In 2.5, the SDE PDU elements and element fields are defined and the transformation of an SDE SDU into an SDE PDU is described.

A security association is an important concept in this standard. A security association is a cooperative relationship between communicating entities, formed by sharing security management information. This shared information coordinates the transmission and reception processing of the SDE PDU. In practice, there are many defined security associations, but only one applies to the processing of a specific SDE PDU. A Security Association Identifier (SAID) associates a defined security association with a specific SDE PDU. In 2.6, the contents of the security management information are defined and the use of the SAID in finding the applicable security association is described.

The Layer 2 security services provided by the SDE rely on information from non-Layer 2 key management or system management entities. Management entities communicate the information to the SDE entity through a Security Management Information Base (SMIB). The implementation of the SMIB is a local issue; however, the standard specifies the structure of the information as defined in the Structure of Management Information (ISO/IEC 10165-2: 1992). The SMIB, the security management architecture, and the procedures for processing the SDE PDU based upon the security management information contained in the SMIB are described in 2.6.

2.2 Definitions

See 1.3.1.

2.3 SDE security services

This subclause contains a description of the security services provided by, or supported by, the SDE entity, and the threats these security services protect against.

The security services are as follows:

a) *Data confidentiality.* The SDE entity provides data confidentiality by enciphering the SDE SDU. The SDE entity provides for the use of multiple confidentiality algorithms and depends on an external key management service to establish a data enciphering key and data deciphering key and for choosing an appropriate cryptographic algorithm.
b) *Connectionless integrity.* The SDE entity provides connectionless integrity by calculating an Integrity Check Value (ICV) and placing it in the ICV field of the SDE PDU. The SDE entity depends on an external key management service to establish an integrity algorithm and integrity key.
c) *Data origin authentication.* Data origin authentication is achieved by the use of key management. It is supported by the SDE entity placing a Station ID in the Protected Header portion of the SDE PDU. The inclusion of the Station ID also prevents undetected reflection of the SDE PDU. Data origin authentication can only be provided in conjunction with the integrity service.
d) *Access control.* Access control is provided by one or a combination of the following: key management, system management, and the labeling of SDE PDUs. The SDE entity's use of security associations supports management's access control decisions. The SDE entity cannot transmit or deliver a PDU unless a security association exists. It is management's responsibility to set up the security associations and the SDE's responsibility to enforce the access control policy. Access control is

[11] To use the management functionality of ISO/IEC 15802-2, LAN/MAN management, service and protocol, and ISO/IEC 9596, Common Management Information Protocol (CMIP), the SDE is modeled as part of LLC. If these management protocols are not used, it is possible to model SDE as a Data Link sublayer directly above the MAC sublayer.

dependent on both integrity and authentication services. Access control can only be provided in conjunction with integrity and authentication.

The threats that these services protect against are as follows:

— Unauthorized disclosure
— Masquerading
— Unauthorized data modification
— Unauthorized resource use

The rationale for addressing these threats is contained in Annex 2A. The dependencies among the security services are shown in Table 2-1.

Table 2-1—Security service dependencies

Service	Dependency
Confidentiality	No dependencies
Integrity	No dependencies
Authentication	Depends on integrity
Access control	Depends on authentication and integrity

It is not necessary for all stations in the LAN or MAN to employ the SDE protocol. It is possible for entities that do not employ the protocol to communicate with those that do employ the protocol.

The SDE protocol is required to be transparent to existing implementations. Transparency, in the context of this standard, consists of meeting the following requirements:

a) Existing IEEE 802 entities shall be able to recover if they receive an SDE protected packet.
b) SDE entities shall be able to accept non-SDE protected packets without impairment.
c) The addition of security should not modify either the (N+1)-layer or (N-1)-layer implementations.

Note that the addition of the SDE protocol may cause certain network management values, such as the fragmentation size, to change, but can still be considered a transparent implementation.

2.4 SDE service specifications

The services provided by SDE are defined in this subclause. SDE is modeled as part of the LLC entity and relies on the services provided by the MAC sublayer. There are only two primitives that are used at the SDE boundary: UNITDATA.request and UNITDATA.indication. These primitives are described in detail in ISO/IEC 15802-1: 1995.

In subsequent subclauses of this document, the primitives on the upper boundary of the SDE are prefixed with “SDE”, and the primitives on the lower boundary are prefixed with “MA” (see Figure 2-2). The services provided at the upper SDE boundary include those provided by the MAC sublayer with the addition of those services provided transparently by the SDE.

The primitives used across the SDE service interface are a subset of the MAC primitives defined in ISO/IEC 15802-1: 1995. Additional primitives specified by other MAC interfaces shall be passed unaltered through

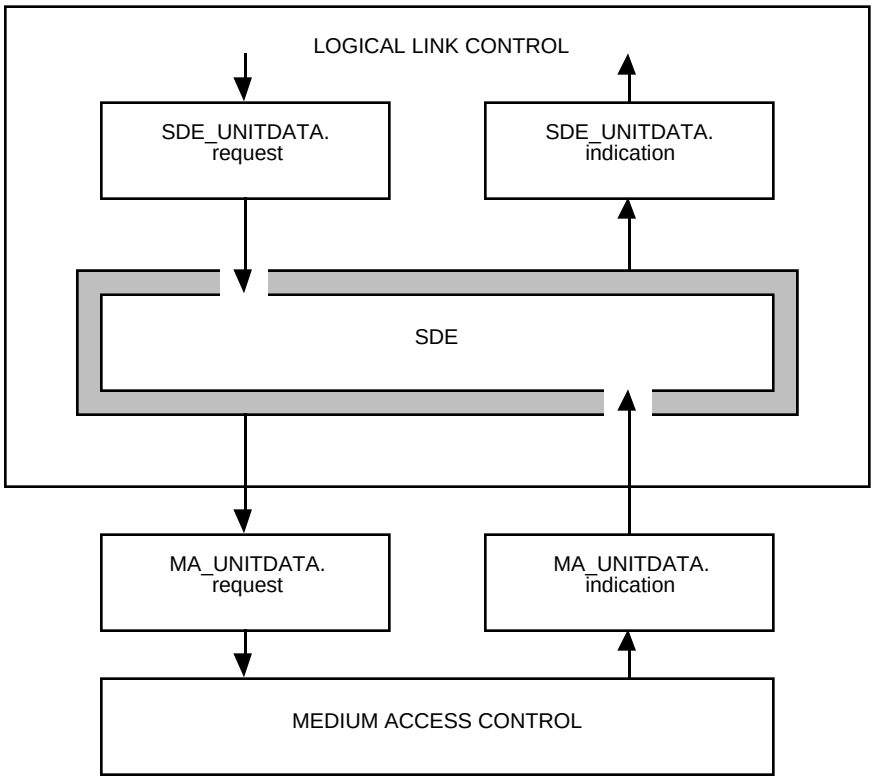


Figure 2-2—SDE primitives

SDE. Likewise, the minimum set of parameters of these primitives is specified. Other MAC interfaces, such as those in ISO/IEC 8802-5: 1998 [B2] are also allowed, and shall be passed through without modification. The MAC primitives that make up the SDE subset are as follows:

UNITDATA.request	Source Address (SA) Destination Address (DA) MAC Service Data Unit (MSDU)
UNITDATA.indication	SA DA MSDU

2.4.1 SDE_UNITDATA.request parameters

The parameters associated with the SDE_UNITDATA.request are defined in ISO/IEC 15802-1: 1995.

2.4.2 SDE_UNITDATA.indication parameters

The parameters associated with the SDE_UNITDATA.indication are defined in ISO/IEC 15802-1: 1995.

2.4.3 Services assumed

The service primitives assumed at the lower boundary of SDE are those defined in ISO/IEC 15802-1: 1995.

The SDE entity assumes the existence of an SMIB that is accurately maintained by a method outside the scope of the SDE entity.

2.5 SDE PDU structure

The structure of the SDE PDU is described in this subclause. The SDE PDU format is described in 2.5.1. In 2.5.2, the relative positions of the various elements of the SDE PDU are defined. This subclause includes descriptions of the fields in terms of size and content. These fields are also defined as either optional or mandatory. The transformation of an SDE SDU to an SDE PDU is described in 2.5.3.

2.5.1 SDE PDU format

SDE uses a single PDU type. The PDU contains an integral number of octets. The PDU format, which may contain up to five elements, is shown in Figure 2-3. These elements include the Clear Header, Protected Header, Data (SDE SDU), Pad, and the ICV. All of these elements are optional except Data. The contents of the Protected Header, Data, and Pad may be transformed prior to transmission by the integrity algorithm. The contents of the Protected Header, Data, Pad, and ICV shall always be transformed when the confidentiality algorithm is applied.

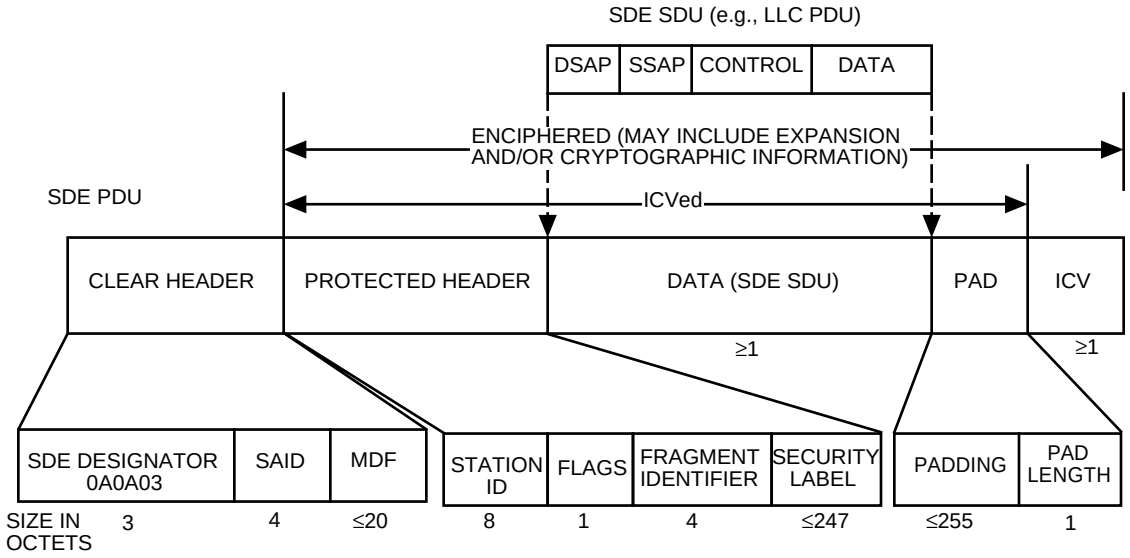


Figure 2-3—Structure of the SDE PDU

2.5.2 Elements of the SDE PDU

2.5.2.1 Clear Header

The Clear Header (see Figure 2-4) identifies the SDE PDUs and aids in the processing of information contained in these PDUs. The content of the Clear Header is determined during security association setup and is constant for the life of that security association. The use of the Clear Header is optional.[12] When the Clear Header is present, its length will be from seven to 27 octets, inclusive.

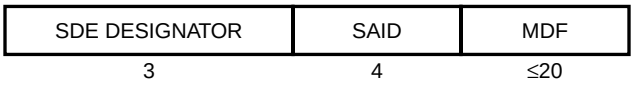


Figure 2-4—Clear Header

[12] Some bridges operate on data outside the MAC header. As a result, SDE frames that do not include the SDE Designator may not be processed correctly. Therefore, when bridges of this type are employed, the option for Clear Header should be selected.

2.5.2.1.1 SDE Designator

The first three octets of the Clear Header constitute the SDE Designator, which ensures that a non-SDE entity that contains an LLC-entity will not process the SDE PDU. The SDE Designator contains the value of a reserved LSAP in each of the first two octets and the Unnumbered Information control field, as defined in ISO/IEC 8802-2: 1998 (P-bit equal to zero), in the third octet. In this and subsequent subclauses, the octets in each field shown are ordered left to right and the leftmost bit is the first bit received from, or sent to, the MAC sublayer. The SDE Designator is mandatory when the Clear Header is present.

The reserved LSAP contains the binary value 0101 0000, where the leftmost digit is the least significant. This value has been assigned by ISO and IEC to ensure unique identification of SDE PDUs. The value of the Unnumbered Information Control field is 1100 0000, where the first "1" is the least significant bit. In hexadecimal, the SDE Designator is represented as a sequence of three octets with values 0A 0A 03, ordered left to right.

2.5.2.1.2 SAID

The SAID field identifies the security association. It contains the SAID associated with the destination SDE entity. If the destination is a group address, the SAID value is common for all the stations in the group and is negotiated by key management or system management. The SAID field is four octets in length and is mandatory when the Clear Header is present.

The format of the SAID is shown in Figure 2-5. The leftmost bit of the SAID is called the G-bit. This is the first bit received from the MAC sublayer. It is used to indicate whether the security association identified by the SAID is common to a group of SDE entities (value set to 1) or an individual SDE entity (value set to 0).

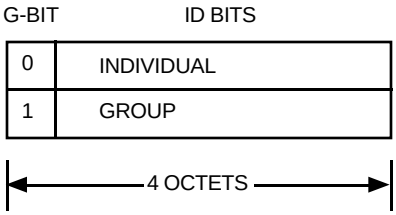


Figure 2-5—SAID format

Four SAID values are reserved for the purpose of establishing initial communication with key management or system management when an SAID has not already been negotiated. These SAID values are called "bootstrap" SAIDs, and identify preestablished security associations. If the bootstrap SAID is used for key management, the ID bits contain all zeroes. If the bootstrap SAID is used for system management, the ID bits contain all ones. The use of the bootstrap SAID mechanism is optional. Communication to the System Management and Key Management Stacks may be accomplished via the use of any security association whose SDE_SAP object indicates the appropriate stack. Also note that the function of key management or system management can reside on a User Stack; however, the bootstrap SAIDs cannot be used to support those implementations.

When the destination address is a group address, key management or system management will create group security associations by assigning a unique group SAID, distributing the keying material to the group members, and determining which services will be used for secure transmission of multicast data. To ensure that key management and system management can assign unique group SAIDs, the available numbers are divided as shown in Figure 2-6. Since some managers will be responsible for a large number of group security associations and other managers will be responsible for a few group security associations, three different group SAID classes are defined. The first class allows 1024 different managers, each responsible for 1 048 576 group security associations. A product vendor may obtain an identifier associated with this class

and assign a portion of the numbers to each instance of their product. The second class allows 131 072 different managers, each responsible for 4096 group security associations. The third class allows 2 097 152 different managers, each responsible for 256 group security associations.

One identifier from the third class of group SAIDs is reserved for testing and managers on isolated LANs. This reserved identifier is 800001 (hexadecimal). Therefore, group SAIDs between 80000100 and 800001FF (hexadecimal) are reserved for this purpose.

<table>
<tr><td>G-BIT</td><td>S1-BIT</td><td>S2-BIT</td><td colspan="3">ID BITS</td></tr>
<tr><td>0</td><td colspan="5">INDIVIDUAL SAID</td></tr>
<tr><td>1</td><td>1</td><td colspan="4">10 BITS 20 BITS</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td colspan="3">17 BITS 12 BITS</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td colspan="3">21 BITS 8 BITS</td></tr>
</table>

Figure 2-6—Group SAID Format

For the purpose of assigning group SAID identifiers, the Institute of Electrical and Electronics Engineers, Inc., USA, is the Registration Authority.[13]

2.5.2.1.3 Management-Defined field (MDF)

The MDF allows the transfer of information that may facilitate, but is not required for, the processing of the PDU. The MDF is variable in length and is an integral number of octets up to a maximum of 20. Its value is indicated by an entry in the SMIB. The MDF may contain any value and is *not* used to determine the appropriate security association. The MDF value is a unidirectional attribute of the security association and is constant for the duration of that security association. The MDF is optional.

An example of the application of the MDF is an SDE implementation that does not retain cryptographic state information. The transfer of cryptographic state information and keying information in the MDF could facilitate reception processing.

2.5.2.2 Protected Header

The Protected Header is in the portion of the SDE PDU to which the security services are applied. If present, the Protected Header contains up to four optional fields. The fields are Station ID, Flags, Fragment Identifier, and Security Label. These fields shall appear in the order listed if all options are selected. The Station ID uniquely identifies the originating station. It is eight octets and contains the canonical form of the MAC address as specified in IEEE Std 802-1990, Section 5.2. The first octet of the Station ID field shall contain the first octet of the MAC address: the contents of the field after the MAC address is undefined. Annex 2E covers the use of the Flags and Fragment Identifier fields to support fragmentation. Security labels are discussed in Annex 2I.

2.5.2.3 Data

The Data portion of the SDE PDU contains the SDE SDU, which is the MSDU parameter of the SDE service primitive.

[13]Communications regarding registration should be addressed to Registration Authority for IEEE Std 802.10, c/o Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA, tel. (732) 562-3813.

2.5.2.4 Pad

The Pad consists of the Padding and Pad Length fields. The Pad may be used to provide padding for confidentiality and integrity algorithms.[14] Pad is selected on a per-security-association basis. If it is selected, each PDU processed under the association shall contain the Pad Length field.

2.5.2.4.1 Padding field

The Padding field is optional but may be required by the specific confidentiality or integrity algorithm selected. The maximum size of the Padding field is 255 octets. The content of the Padding field is a local matter.

The Padding field specifies an integral number of octets; therefore, the Padding field cannot be used to correct octet alignment problems caused by either the integrity or confidentiality algorithms.

2.5.2.4.2 Pad Length field

The value of the Pad Length field contains the number of octets in the Padding field. This value does not include the one octet required by the Pad Length field itself. If no integrity is requested, the Pad Length field is the last octet of the SDE PDU. If integrity is requested, the Pad Length field is the octet before the ICV.

2.5.2.5 ICV field

The ICV field is a security mechanism for detecting data modification. The ICV value, if present, is contained in the last field in the SDE PDU. The length of the ICV is an attribute of the security association. The ICV is calculated over the Protected Header, the Data field, and Pad. It is an optional field.

2.5.3 Building the SDE PDU

The means by which the information passed to the MAC is used to construct the SDE PDU is described in this subclause. (The MSDU is the SDE SDU.) All of the parameters of the service request except the MSDU are copied unaltered from the SDE_UNITDATA.request to the MA_UNITDATA.request. Likewise, on incoming processing, all parameters except the MSDU are copied unaltered from the MA_UNITDATA.indication to the SDE_UNITDATA.indication. The MSDU is used to generate the SDE PDU as shown in Figure 2-7. On reception, the process is reversed to reconstruct the MSDU. The encipherment algorithm may require the addition of fields specific to the algorithm. These fields will be added and removed as part of the encipherment or decipherment processing. They will be transmitted as part of the SDE PDU provided in the MSDU of the MAC service primitives. An example of this type of field is the Initialization Vector (IV) required by certain algorithms.

2.6 SDE procedure

All elements of the SDE procedures, including transmission and reception processing and all other elements that direct those procedures, are defined in this subclause. These other elements include management architecture, addressing, the SMIB, and the definitions of the managed objects.

The SDE management architecture is described in 2.6.1. The architectural description includes the following:

a) The relationship between the management application entity and the SDE Layer Manager (LM);

[14]Many confidentiality algorithms take blocks (n bits) of cleartext and transform this cleartext to ciphertext as a unit. This block is known as a cryptographic block. The confidentiality algorithm may require that the input cleartext be a multiple of this block size. If the chosen confidentiality or integrity algorithm has this restriction, then the SDE protocol uses the Pad to make sure that the cleartext is a multiple of the block size. (The Pad follows the Data to allow stream processing for outgoing PDUs.)

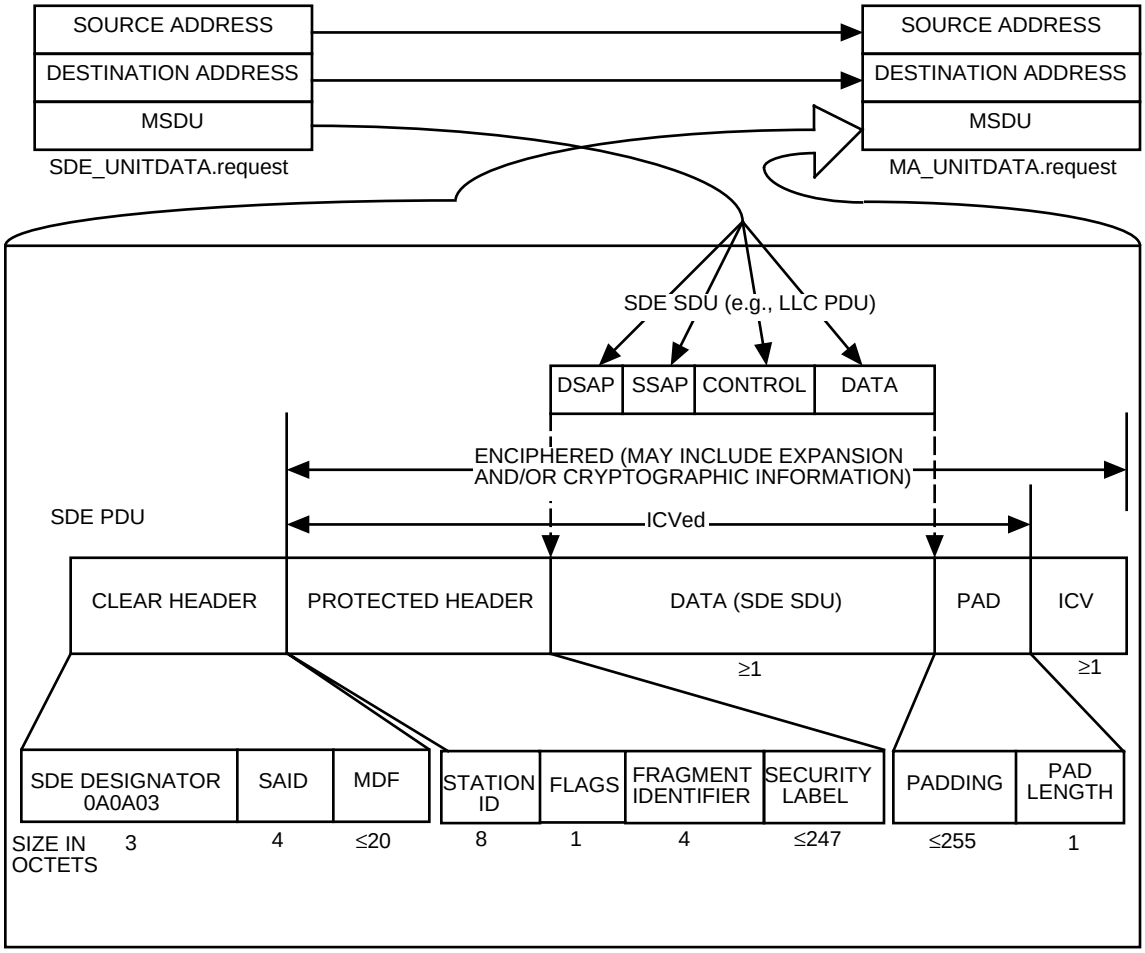


Figure 2-7—Construction of the SDE PDU

b) The role of the SMIB in their relationship;
c) How security associations are coordinated through the use and the exchange of SAIDs; and
d) The structure of the SDE managed objects.

The type of addressing used by the SDE entity is described in 2.6.2. The details of the SDE objects that are attributes of SDE managed objects are described in 2.6.3. Finally, the transmission and reception procedures are described in 2.6.4 and 2.6.5, respectively.

2.6.1 SDE management architecture

Each station that employs the SDE protocol has access to an SMIB. The SMIB contains a list of the current security associations. Key management and security management or both are responsible for maintaining this information base.

The SMIB provides the interface between the local System Management Application Entity (SMAE) and the LM of the protocol stack. This is illustrated in Figure 2-8.

There are three types of SDE managed objects: station, Service Access Point (SAP), and security association. Station objects, which set certain parameters for the SDE entity, apply to all processing by the SDE entity. The SAP objects apply to a specific SAP. The security association objects apply only to a specific instance of PDU transmission, reception, or both.

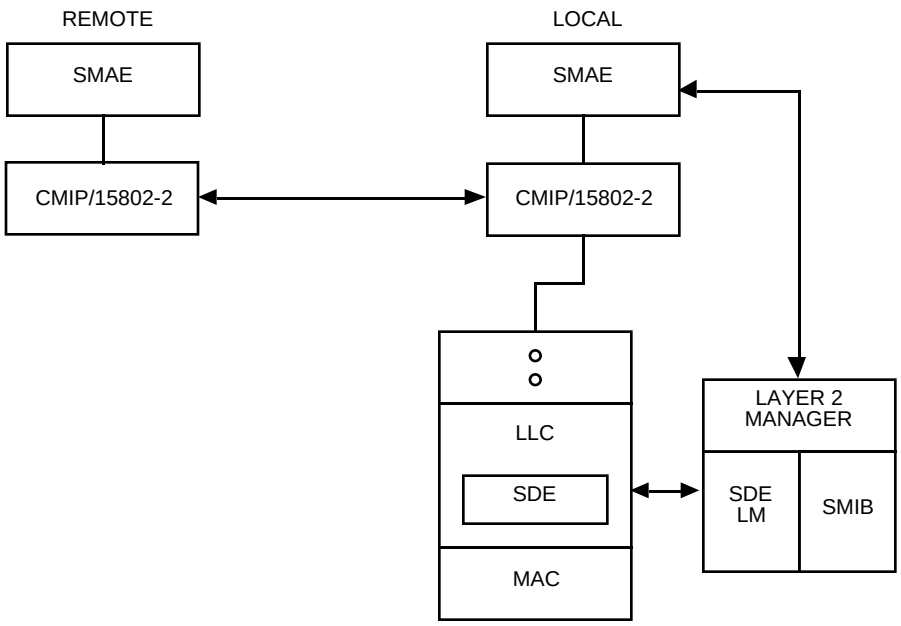


Figure 2-8—SDE management architecture

Since multiple security associations can exist at any time, the SDE entity shall identify which security association applies to that SDE PDU. For example, this identification may be passed via the optional SAID.

How the value of the SAID is coordinated between SDE entities is independent of the SDE protocol; however, it is useful to examine how a pairwise SAID could be established. During either a key or system management exchange, parties A and B exchange the values of the attributes of the security association managed object. These values specify the security parameters (e.g., the security services employed, keys, etc.) that will be needed for the security association. In this example, the SAID identifies this security association. This process is illustrated in Figure 2-9.

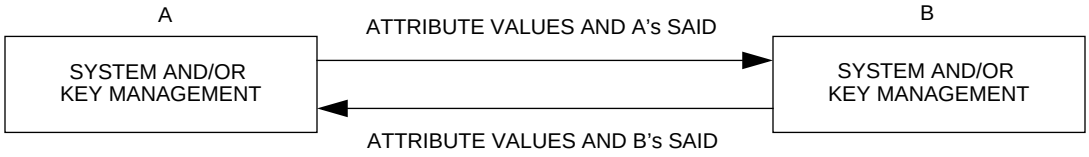


Figure 2-9—Initial exchange

System and/or key management enters the value for the security association object into the SMIB. An example of an SMIB that contains a table of security associations and the values of the associated attributes is illustrated in Figure 2-10.

The security association shall be selected for each PDU transferred through the SDE entity. Outgoing PDUs are PDUs that originate at one of the SDE stacks (i.e., System Management Stack, Key Management Stack, or one of the User Stacks) and are to be delivered to the MAC sublayer. Incoming PDUs are PDUs that arrive from the MAC sublayer and are to be delivered to one of these stacks. Incoming PDUs may contain the SDE Clear Header, which can be used to select the security association; whereas the Clear Header may be created for outgoing PDUs after the security association has been found. For this reason, the mechanism for select-

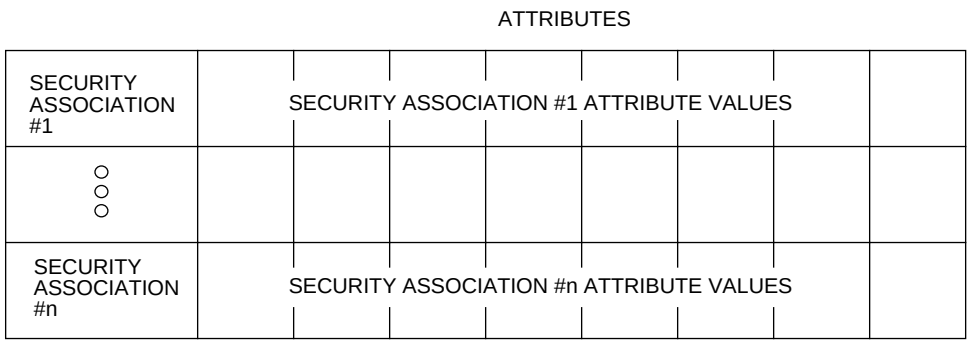


Figure 2-10—Example SMIB containing security associations

ing the security associations can be different. The different parameters and/or PDU fields that can be used for selecting the appropriate security association from the SMIB are shown in Figure 2-11.

Outgoing

— SDE SAP and/or outgoing MAC SA/DA pair

Incoming

— SAID and/or incoming MAC SA/DA pair

Figure 2-11—Parameters used for selecting security association

2.6.2 Addressing

All addresses referred to in this protocol are either LSAP addresses or MAC addresses. The LSAP address syntax and semantics are defined in ISO/IEC 8802-2: 1998, while the specifics of the MAC addresses are defined in IEEE Std 802-1990.

The Station ID contains the MAC address corresponding to the individual address of the station that originated the outgoing PDU. In group transmissions with a shared secret key, the Station ID prevents parties external to the multicast group from tricking the receiving party into believing that the PDU came from a party other than its originator. It does *not* prevent members within the same group from changing PDUs so that they appear to have originated from another valid member of the same group. The inclusion of a Station ID also provides protection against reflection where that protection is not provided implicitly by the SDE confidentiality or integrity algorithms or from the services of the key management protocols.

2.6.3 SDE objects

Security managed objects as outlined in 2.6.1 are described in this subclause. Security objects that apply to the entire SDE are described in 2.6.3.1; security objects that apply to transmission to and from an SDE SAP are described in 2.6.3.2; security objects that are specific to the security associations are described in 2.6.3.3; and SAIDs are described in 2.6.3.4.

2.6.3.1 Station objects

The station objects apply to the entire SDE regardless of security association. The formal definition of each of the objects in the SDE clause will be defined in 2.8, SDE sublayer management. The objects described in this and the following two subclauses are abstractions provided for the purpose of describing the protocol

processing. Some implementations may choose only manual management of these objects; in which case, the representation becomes a purely local matter. In this and the following two subclauses, the object names will be in boldface type.

a) **Station_Clear_Hdr:** Boolean. Station_Clear_Hdr=TRUE indicates that the Clear Header is always used when communicating with other SDE entities.
Station_Clear_Hdr=FALSE indicates that there is no Clear Header expected on any incoming PDUs, and there is none placed on outgoing PDUs.
For communication using this mode of the protocol, both stations shall agree to have Station_Clear_Hdr=FALSE. Delivery of SDE PDUs with no clear header (Station_Clear_Hdr=False) will have unpredictable results if the receiving entity is one of the following:

Its Layer 2 entity does not employ SDE.

Its station object Station_Clear_Hdr=TRUE.

b) **Station_MDF:** Boolean. Station_MDF shall be set to TRUE if the station sends or desires to receive the MDF in the Clear Header. The actual inclusion or exclusion of the MDF is determined by the value of the Assoc_MDF attribute.

2.6.3.2 SDE SAP objects

These objects may be defined by 2.8, SDE sublayer management, and in Annex 2E.

2.6.3.3 Security association objects

The SDE entity uses security associations available to it via the SMIB to provide the necessary services required for the secure transmission of data. The following are security objects that are attributes of a security association managed object:

a) **Local_SAID:** Octetstring. This contains the value of the SAID expected in incoming PDUs if Station_Clear_Hdr=TRUE.
b) **Remote_SAID:** Octetstring. This contains the value placed in the SAID field of outgoing PDUs if Station_Clear_Hdr=TRUE.
c) **Assoc_MDF:** Boolean. This indicates whether or not the MDF is used for the security association. If Station_MDF=FALSE, then this Boolean is always FALSE. The length and value of the MDF field in the PDU are unidirectional characteristics of the security association. Key management and/or system management can force this Boolean to FALSE. If the Boolean is TRUE, the value of the following attribute is placed in the MDF of outgoing PDUs:
    1) **Remote_MDF:** Octetstring. This attribute contains the value that will be placed in the MDF field in the Clear Header if the Assoc_MDF=TRUE.
d) **Protection set:** These attributes indicate the security services to be provided by SDE.
    1) **Confid:** Boolean. If TRUE, it indicates that data confidentiality is to be provided for the security association.
    2) **Integ:** Boolean. If TRUE, it indicates that connectionless integrity is to be provided for the security association.
e) **Security fields present:** Booleans indicate the presence (TRUE) or absence (FALSE) of security fields. These values shall remain constant over the life of the security association.
    1) Padding_pres: Boolean. Flag for the Pad Length field.
    2) ID_pres: Boolean. Flag for the Station ID.
f) **Confid_Alg_ID:** Octetstring. This is a label that specifies a complex object corresponding to a confidentiality algorithm if Confid=TRUE. The definition of the algorithm shall include everything that is necessary for the encipherment or decipherment to occur. This includes, but is not limited to, the length and placement of IVs, block size, and mode of operation.

g) **Integ_Alg_ID:** Octetstring. This is a label that specifies a complex object corresponding to an integrity algorithm if Integ=TRUE. The definition of the algorithm shall include everything that is necessary for the ICV to be calculated and verified upon receipt. This includes, but is not limited to, the length and placement of IVs, block size, and mode of operation.
h) **SDE_SAP:** Octetstring. This indicates the SDE SAP for the security association. This is used as part of the index into the SMIB for outgoing PDUs. On incoming PDUs, it indicates which protocol stack should receive the PDU.
i) **Remote_SDE:** Boolean. This boolean is TRUE if the remote entity implements SDE protocol and is FALSE otherwise.
j) **Outgoing_Source_MAC_Address:** Octetstring. This corresponds to the individual address of the station that originated the outgoing PDU. It is the value included in the Station ID field of the Protected Header.
k) **Outgoing_Destination_MAC_Address:** Octetstring. This address may be an individual or group address associated with the remote station(s).
l) **Incoming_Destination_MAC_Address:** Octetstring. This may be an individual or group address associated with the local station.
m) **Incoming_Source_MAC_Address:** Octetstring. If the Incoming_Destination_MAC_Address is an individual address, this object contains a single individual address. If the Incoming_Destination_MAC_Address is a group address, this object contains a list of individual addresses.

Within the SDE entity, the security association is represented by the security association object. Generic entries corresponding to multiple MAC addresses may be allowed in the SMIB depending on local policy. Changing the values of any of the security association attributes (or attributes of the complex objects labeled by the Confid_Alg_ID and the Integ_Alg_ID attributes) causes a new security association to be formed and the prior security association to be invalidated. The SAID is a convenient tag for the identification of these objects.

2.6.3.4 SAIDs

The SAID is primarily used to identify the security association, although it can be used for other purposes. In security associations between two entities, each entity chooses its own SAID and communicates it to the remote entity during a system and/or key management exchange. In security associations for multicast or broadcast groups, it is the responsibility of system management and/or key management to assign and coordinate the SAID used for that multicast or broadcast group address. Half of the possible values of the SAIDs are reserved as group SAIDs (see Figure 2-5).

There are bootstrap values (see 2.5.2.1.2) for the SAID that are sometimes used for communications with the System Management and/or Key Management Stacks. The communications under these bootstrap SAIDs have no security protection (confidentiality, integrity) and do not have a Station ID. In addition, no padding can be applied.

2.6.4 Transmission procedures

The transmission procedures are those involved in processing an SDE_UNITDATA.request. The functions are represented as a flow chart shown in Figure 2-12. (Object values are contained in the SMIB.) Also, Annex 2B contains an example of the transmission and reception procedures using specific algorithms.

In response to an SDE_UNITDATA.request from the LLC sublayer, the supplied address parameters or the SDE SAP or both are used to search for a security association in the SMIB.

a) If the search is successful, a security header comprised of a Clear Header and a Protected Header may be created and prepended to the Data field that contains the MSDU of the request. The options of integrity and/or confidentiality may be provided. A Pad may be created and an ICV may be com-

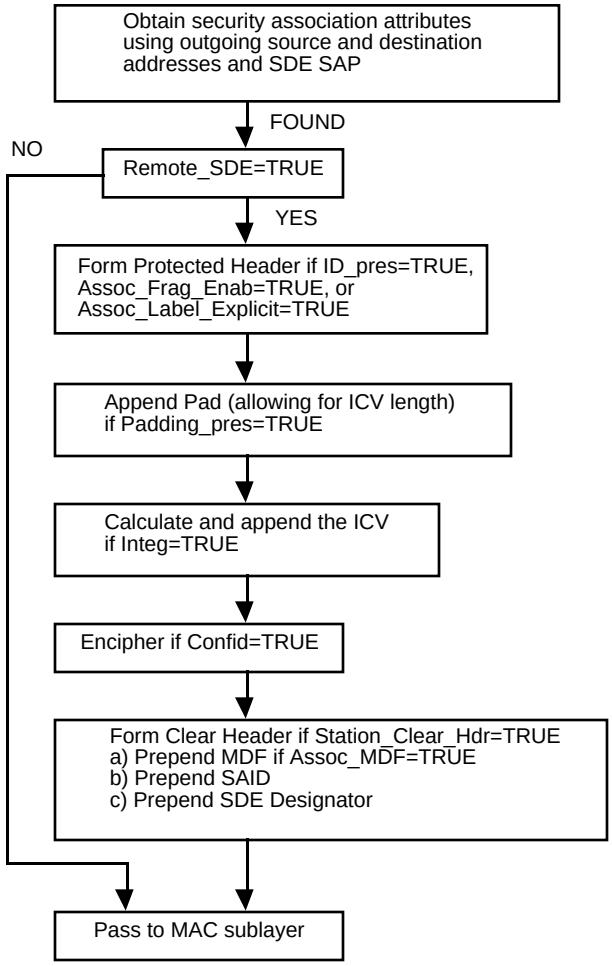


Figure 2-12—Transmission of an MA_UNITDATA.request

puted; both are appended to the Data field. The Protected Header, Data, Pad, and the ICV may be enciphered. Finally, an MA_UNITDATA.request is constructed and passed on to the MAC sublayer.

b) If no security association is found, the SDE LM is notified.

If the expansion causes the PDU to exceed the maximum size the MAC will accept, fragmentation may be required. Fragmentation is not part of this standard; however, if it is implemented, the method of fragmentation specified in Annex 2E is the recommended approach.

2.6.4.1 Obtaining the security association

The security association shall be retrieved from the SMIB if a security association exists. The outgoing MAC addresses and the SDE SAP are used to search for the security association in the SMIB. If a security association is found, the values for each object of the security association are returned. If the request is originated by system management and/or key management, the SMIB may contain a bootstrap SAID security association that will allow communication. If no security association is found corresponding to the SDE SAP and addresses specified in the SDE_UNITDATA.request, the SDE entity indicates the error to the SDE LM and discards the PDU.

From this step in the process until the end of transmission processing, it will be assumed that an appropriate security association is already established.

2.6.4.2 Transmission to non-SDE entities

If Remote_SDE = FALSE, bypass further SDE processing and pass the SDE_UNITDATA.request to the MAC sublayer.

2.6.4.3 Forming the Protected SDE Header

After the security association is retrieved from the SMIB, a Protected Header is formed and prepended to the Data field in the SDE_UNITDATA.request, if either one or a combination of the following are TRUE: ID_pres, Assoc_Frag_Enab, and Assoc_Label_Explicit.

If ID_pres=TRUE, the Outgoing_Source_MAC_Address is placed in the Station_ID field. The Station_ID is an optional field. Refer to Annex 2E for the appropriate processing if Assoc_Frag_Enab=TRUE. Refer to Annex 2I for the appropriate processing if Assoc_Label_Explicit=TRUE.

2.6.4.4 Pad

If padding is required by the security association (Padding_pres=TRUE), the maximum size of Pad is 256 octets (255 Padding octets plus a one octet Pad Length field). Pad may be used to expand the size of the outgoing PDU for the integrity algorithm, for the confidentiality algorithm, or in a local manner.

2.6.4.5 Calculation of the ICV

If integrity should be applied (Integ=TRUE), the ICV is computed over the Protected Header, Data, and Pad using the algorithm specified in the SMIB. The ICV is appended to the Data field.

2.6.4.6 Encipherment of the PDU

If confidentiality is an attribute of the security association (Confid=TRUE), then the Protected Header, Data, Pad, and ICV will be enciphered using the algorithm specified in the SMIB.

2.6.4.7 Clear Header

The Clear Header is used both to signal the remote SDE entity that the PDU had been processed by the local SDE entity and to supply the necessary information to determine the appropriate security association. If Station_Clear_Hdr=TRUE, the Clear Header is placed in the outgoing PDU.

MDF: If Assoc_MDF=TRUE, then the Remote_MDF is placed in the outgoing PDU. It is an optional field.

SAID: The Remote_SAID shall be placed in the SAID field of the PDU. It is a mandatory field when the Clear Header is present.

SDE Designator: The SDE Designator is placed as the first three octets in the outgoing PDU. It is a mandatory field when the Clear Header is present.

2.6.4.8 MAC request

The SDE PDU is passed to the MAC sublayer as the MSDU parameter in the MA_UNITDATA.request. All other parameters are passed through, unaltered, by the SDE entity.

2.6.5 Reception procedures

When an MA_UNITDATA.indication is received from the MAC sublayer, processing can vary, depending on the local management functions. The security association shall be identified, and the appropriate security

mechanisms are applied to the PDU. If appropriate, the PDU is deciphered and the ICV is checked. Finally, the SDE_UNITDATA.indication is forwarded to the designated stack.

If any security-relevant exceptions are encountered during processing by the SDE entity, the PDU in question is discarded and the SDE LM is notified.

The reception functions are those involved in processing an MA_UNITDATA.indication (see Figure 2-13).

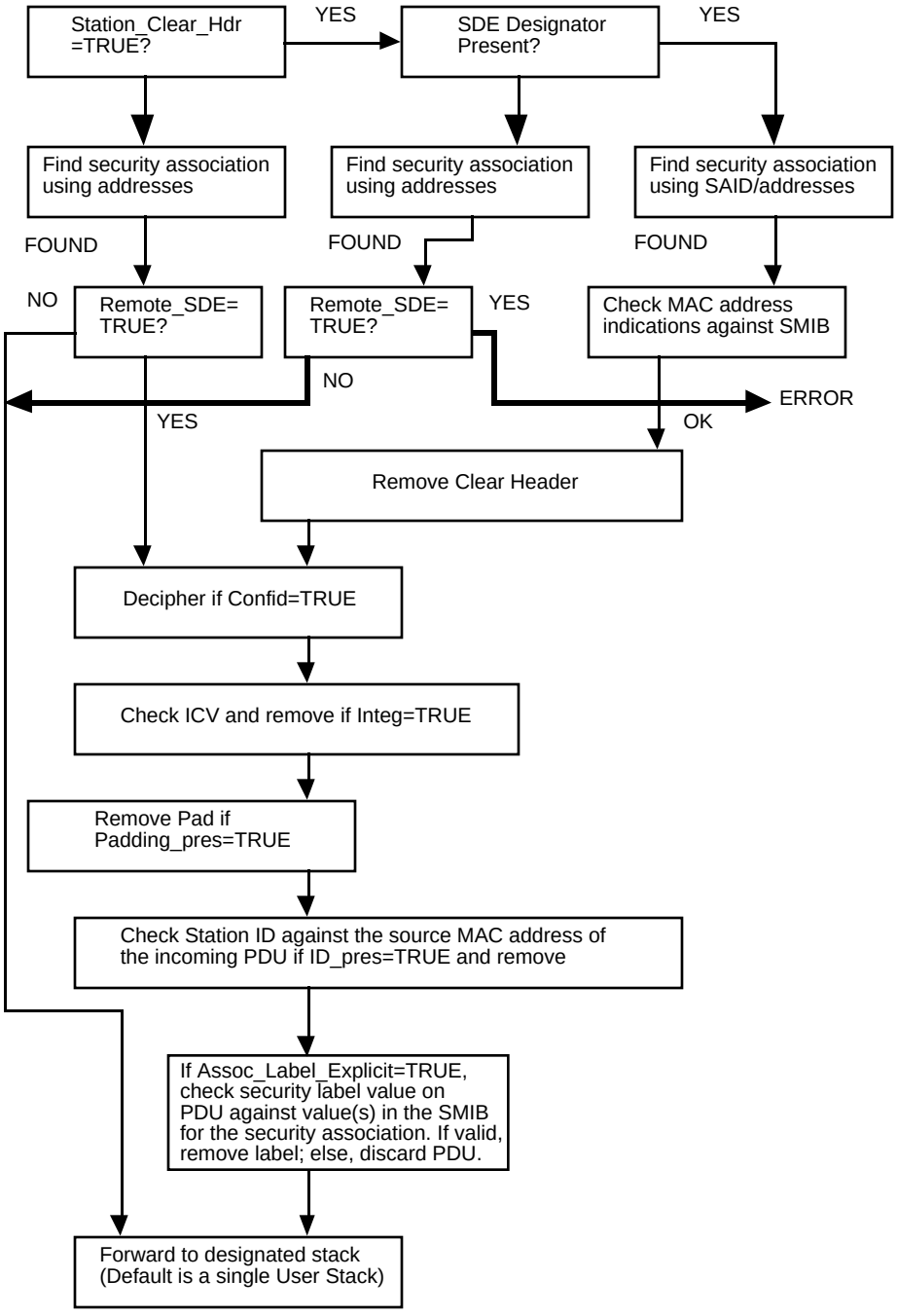


Figure 2-13—Reception of an MA_UNITDATA.indication

2.6.5.1 Requirements for reception

Before a station can process an incoming SDE PDU, a security association shall exist for communication to be allowed. Note that it is possible to configure the SMIB such that loss of information in the SMIB (e.g., power failure) could prevent automated recovery.

The bootstrap values of the SAID shall have a security association in the SMIB. There are four bootstrap values: Individual Key Management, Group Key Management, Individual System Management, and Group System Management.

2.6.5.1.1 Station configured for Clear Header

If Station_Clear_Hdr = TRUE, then the presence of an SDE Designator indicates that an SAID may be used for finding the security association. A security association shall exist for all communications, even with non-SDE entities. An association for communication with a non-SDE entity is found by searching the SMIB using the source and destination addresses. If the security association has Remote_SDE = FALSE, the PDU will bypass the rest of the security processing and be forwarded to the stack designated by the SDE SAP.

The SDE entity checks the source and destination address parameters in the MA_UNITDATA.indication against those denoted by the security association (Incoming_Source_MAC_Address and Incoming_Destintion_MAC_Address). The security association in the SMIB may indicate the presence of an MDF. The MDF is used in a locally determined manner. The Clear Header is removed before the PDU is deciphered.

2.6.5.1.2 Station configured with no Clear Header

If Station_Clear_Hdr=FALSE, then the security association and the correct protocol stack shall be determined based on source and destination addresses in the MA_UNITDATA.indication. If Remote_SDE = FALSE for the security association, the PDU will bypass the rest of the security processing and be forwarded to the stack designated by the SDE SAP.

2.6.5.1.3 Decipherment of the PDU

If Confid=TRUE, the confidentiality algorithm is selected from the SMIB, and the PDU is deciphered.

2.6.5.2 ICV checking

If Integ=TRUE, the PDU is assumed to have an ICV that shall be checked using the chosen algorithm retrieved from the SMIB and subsequently removed from the PDU. If the ICV fails, the PDU is discarded and the SDE LM is notified.

2.6.5.3 Pad

The SDE entity strips any Pad that may be present in the PDU.

2.6.5.4 Station ID

If ID_pres=TRUE, the SDE entity checks that the contents of the Station ID field are the same as the source address in the MA_UNITDATA.indication. The Station ID field is removed.

2.6.5.5 Security label

If Assoc_Label_Explicit=TRUE, the SDE entity checks the security label value to ensure that it is within the set of allowed values for the association in the SMIB. If valid, the security label is removed. If not valid, discard the PDU and notify local management.

Refer to Annex 2I for detailed security label processing.

2.6.5.6 SDE_UNITDATA.indication

The parameters received in the MA_UNITDATA.indication are passed up to the appropriate protocol stack in the SDE_UNITDATA.indication with the SDE SDU (e.g., the LLC PDU) replacing the received MSDU.

2.7 Minimum Essential Requirements (MERs)

The MERs are stated in terms of the values of certain management objects in 2.7.1 and 2.7.2. Additional MERs are contained in 2.7.3 and 2.7.4. These objects are abstractions used to represent the options for the SDE entity. These MERs do not mean that the objects shall be managed remotely. The effect of setting the object to a particular value shall affect the protocol state as described previously in Clause 2. When constrained to the values specified in the following two subclauses and combined with transmission and reception processing, these objects delineate the minimally compliant protocol state machine.

2.7.1 Station objects

a) Station_Clear_Hdr: Boolean
   SDE entities shall allow the Station_Clear_Hdr to be TRUE. Entities with Station_Clear_Hdr set to TRUE are not interoperable with stations that have Station_Clear_Hdr set to FALSE.
b) Station_MDF: Boolean
   All entities shall allow the Station_MDF to be FALSE. Entities may support TRUE, but then values for each individual security association are determined by system and/or key management. If Station_Clear_Hdr is set to FALSE, then Station_MDF shall also be set to FALSE.

2.7.2 Security association objects

a) Assoc_MDF: Boolean
   This attribute is TRUE if the MDF will be used on the security association. Any party in the negotiation can force the MDF not to be used. The protocol processing shall not depend on the presence of the MDF in any implementation. Each entity shall have the capability of communicating with this attribute set to FALSE. If TRUE is supported, the entity shall have the capability of supporting an integral length (in octets) from zero to 20.
b) Protection Set: Includes Confid and Integ Booleans
   An entity shall be capable of operating with at least one security association having a TRUE value in at least one of these two Booleans. Entities implementing only Integrity, or only Confidentiality, shall be considered conformant.
c) Padding_pres: Boolean
   The Pad field is mandatory only if either the integrity or confidentiality algorithm requires padding. Thus, some entities may support this object only being TRUE, and others may support it only set to FALSE. Still others may support both. If the TRUE value is supported, the entity shall be able to accept a maximum length of Pad (256 including Pad Length field).[15] Negotiation of the

[15]The maximum Pad length must be specified due to its effect on stream processing and buffer sizes. It cannot be restricted to the block-size of the Integrity or Confidentiality algorithm since this would defeat the objective of algorithm independence and require conformance testing to be tied to a particular cryptographic algorithm.

Padding_pres by key management and/or system management may allow the value to be set to FALSE where neither cryptographic algorithm requires padding.

d) ID_pres: Boolean
A device shall be capable of supporting ID_pres=FALSE.

2.7.3 SAID requirements

All entities shall support the reception of bootstrap, group, and individual SAIDs. In systems with key management appearing on the User Stack, the SDE entity associates the bootstrap SAID with the appropriate stack identified by the SDE SAP in the SMIB. If no security association is found for the SAID, the PDU is discarded and the SDE LM is notified.

2.7.4 Security services

Compliant entities shall support at least the Data Confidentiality Service or the Connectionless Integrity Service.

a) To claim that the entity provides the service of Data Confidentiality, the entity shall allow Confid to be TRUE. The strength of this service is dependent upon the confidentiality algorithm used.
b) To claim that the entity provides the service of Connectionless Integrity, the entity shall allow Integ to be TRUE. The strength of this service is dependent upon the integrity algorithm used. The entity shall be able to send and receive PDUs with ICVs.

2.8 SDE sublayer management

2.8.1 Overview

This clause specifies the managed objects that permit the operation of the SDE to be remotely operated, managed, and maintained. Both fully specified managed objects and partially specified managed objects together with their contained management information are documented.

2.8.2 Scope

This standard provides specifications for ISO/IEC 10165-style managed object classes that represent SDE sublayer resources that are subject to management. These specifications define managed object classes, attribute types, specific attributes, actions, and notification types in accordance with ISO/IEC 10165-4: 1992 (GDMO) and the generic definitions contained in ISO/IEC 10165-2: 1992 (DMI). This standard also follows the procedures and uses the managed objects defined in IEEE Std 802.1F-1993.

This standard provides the following two types of management information to the developers of SDE sublayer management services:

a) Definition of the managed object classes required for management of the SDE protocol; including their relationships with other managed objects within the Data Link layer and the bindings that are required in the naming of an instantiation of the management service, and;
b) Definition of common management information such as ATTRIBUTES, PACKAGES, BEHAVIOURS, NOTIFICATIONS, and encodings of information that are used in managing the exchange of data in accordance with SDE specifications.

These managed object classes allow a manager (or management application processes) to monitor and control resources, represented as managed objects, through local agents in other (managed) systems. Event reports (notifications) may be returned from resources in other systems through their local agents to a designated manager or managers.

2.8.3 Definitions

2.8.3.1 LAN/MAN Management definitions

The SDE sublayer management subclause makes use of the following terms defined in ISO/IEC 15802-2: 1995 and ISO/IEC 7498-1: 1994:

a) LAN/MAN Management

2.8.3.2 Common Definitions and Procedures for IEEE 802 Management Information definitions

The SDE sublayer management subclause makes use of the following terms defined in the text of IEEE Std 802.1F-1993:

a) MACAddress
b) ResourceTypeID

2.8.3.3 Management Framework definitions

The SDE sublayer management subclause makes use of the following terms defined in ISO/IEC 7498-4: 1989:

a) managed object
b) management information base
c) systems management

2.8.3.4 Abstract Syntax Notation One (ASN.1) definitions

The SDE sublayer management subclause makes use of the following terms defined in ISO/IEC 8824: 1990:

a) object identifier
b) type

2.8.3.5 Common Management Information Service (CMIS) definitions

The SDE sublayer management subclause makes use of the following terms defined in ISO/IEC 9595: 1991:

a) attribute (of managed object)
b) Common Management Information Services

2.8.3.6 Guidelines for the Definition of Managed Objects (GDMO) definitions

The SDE sublayer management subclause makes use of the following terms defined in ISO/IEC 10165-4: 1992:

a) managed object class definition
b) template

2.8.3.7 Systems Management Overview definitions

The SDE sublayer management subclause makes use of the following terms defined in ISO/IEC 10040: 1992:

a) agent
b) agent role
c) generic definitions
d) layer management protocol
e) managed object class
f) managed (open) system
g) management information
h) manager
i) manager role
j) notification
k) notification type
l) (systems management) operation
m) systems management application protocol

2.8.3.8 Structure of Management Information (SMI) Information Model definitions

The SDE sublayer management subclause makes use of the following terms defined in ISO/IEC 10165-1: 1993:

a) abstract datatype
b) attribute type
c) attribute value assertion
d) behaviour
e) containment
f) containment hierarchy
g) distinguished name
h) inheritance
i) inheritance hierarchy
j) name binding
k) naming tree
l) mandatory package
m) (conditional) package
n) relative distinguished name
o) subclass
p) subordinate object
q) superclass
r) superior object
s) unrestricted abstract datatype

2.8.4 Management model

This standard describes the management of the SDE sublayer in terms of a general model of resource management within an open systems environment. Two major components of this management model are a *managing system* and a *managed system*.

Within the managing system (manager role), management processes invoke management operations that are translated into management messages, which are sent to the managed system. A managed system (agent role) provides an interface to those internal resources that are to be managed and processes the applicable management (operations) messages.

The internal resources of the managed system are represented at the interface by abstractions known as *managed objects*. Each managed object responds to a defined set of management operations with internal actions and notifications that the managed system translates into messages. These messages are sent to the manage-

ment process that requested the management operation. Also, internal changes in the managed resources may trigger sending other notification messages to a management process.

2.8.4.1 SDE management requirements

In a dynamic system, where SDE must be set up, managed, enabled or disabled, both commands (operations) and information requests must be sent to the SDE protocol entity to perform these functions. These operations involve the creation and deletion of various SDE sublayer resources that manage that SDE sublayer and the setting of information values. The information requests involve the transfer of management information values. Other management information transfers include the reporting of events (notifications) within the SDE protocol entity.

Other methods for establishing and manipulating management knowledge may be used in addition to or in place of the ones described in this standard. However, the availability and the use of mechanisms identified in this standard will promote vendor interoperability.

2.8.4.2 SDE management services

Management services provide for the exchange of management operations and information.

Within the OSI environment, a set of management standards, that includes ISO/IEC 9595: 1991 (CMIS) and ISO/IEC 9596-1: 1991 (CMIP), defines these services and messages. Within the IEEE 802 environment, the LAN/MAN Management standard (ISO/IEC 15802-2: 1995) specifies these management services. ISO/IEC 15802-2: 1995 defines an OSI management-compatible architecture, services, and protocol elements for use in a LAN/MAN environment for the purpose of performing remote management of LAN/MAN-based devices. Local management on each LAN/MAN-based host is done by local agents that perform the management operations on locally managed objects and return the management responses and notifications. This local management is performed internally using methods that are not subject to standardization. The responses and notifications are sent to the remote manager via the standardized interface, services, and protocol.

Figure 2-14 illustrates ISO/IEC 15802-2: 1995 LAN/MAN Management and OSI Management of the SDE sublayer.

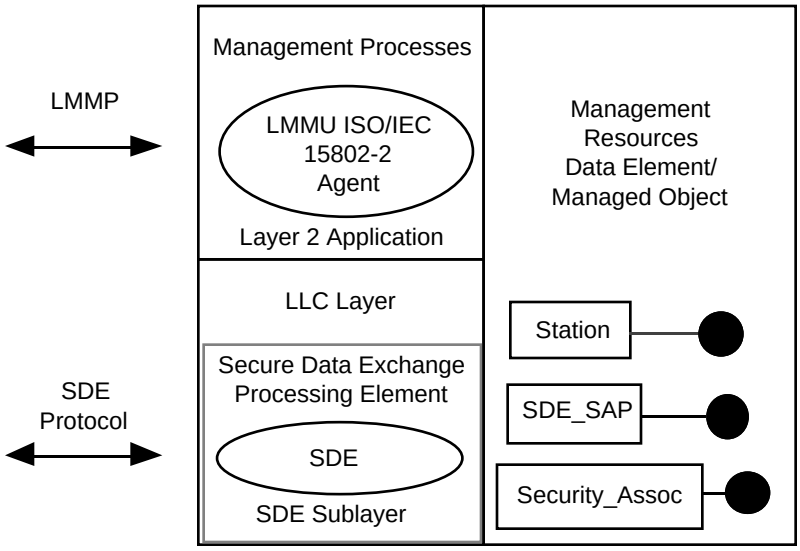
The SDE management information is represented as data objects contained within managed objects that represent those resources that are being managed. These management variables are available to the SDE protocol entity as data input or as output values, such as counters or writable variables.

These management variables can be viewed externally as attributes to managed objects through the management interface. Thus the SDE sublayer resources can be remotely managed either by LAN/MAN Management user processes (LMMUs) using the LAN/MAN Management Services (LMMS) and Protocol (LMMP) or by Management Information Service user processes (MIS-User at ISO layer 7) using Common Management Information Services (CMIS) and Protocol (CMIP).

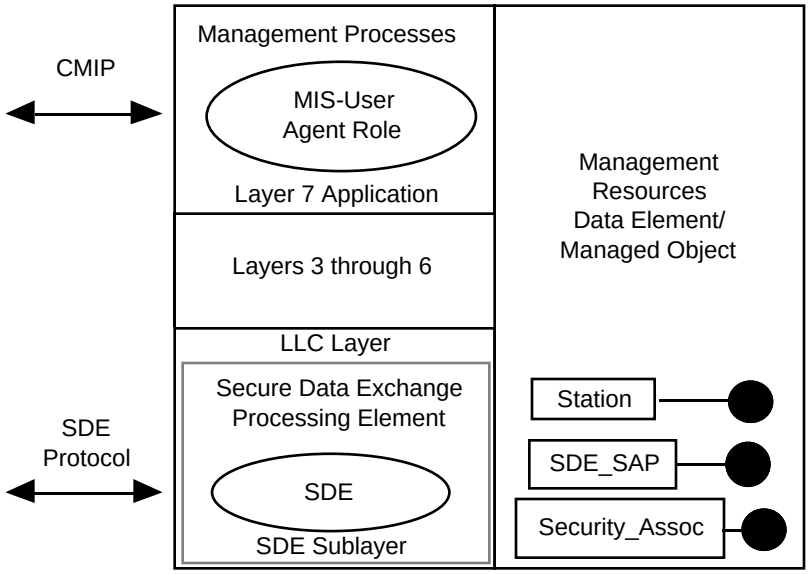
The interactions between an IEEE LAN/MAN (ISO/IEC 15802-2: 1995) compliant manager and the managed objects associated with an ISO/IEC 15802-2 LMMU Agent are illustrated in Figure 2-15. A similar set of interactions occurs between an ISO Management Process and those managed objects either via a MIS-User Agent, via a hybrid agent that handles both CMIP and LMMP messages, or a via a LMMU Agent that transforms the CMIP messages to LMMP messages and vice versa.

2.8.4.3 SDE management operations

The manipulation of managed objects is carried out by a set of management operations, which are part of the management service at the management interface. The management operations that can be sent to a managed



a) ISO/IEC 15802-2 LAN/MAN Management



b) ISO Management

Figure 2-14—SDE management relationships

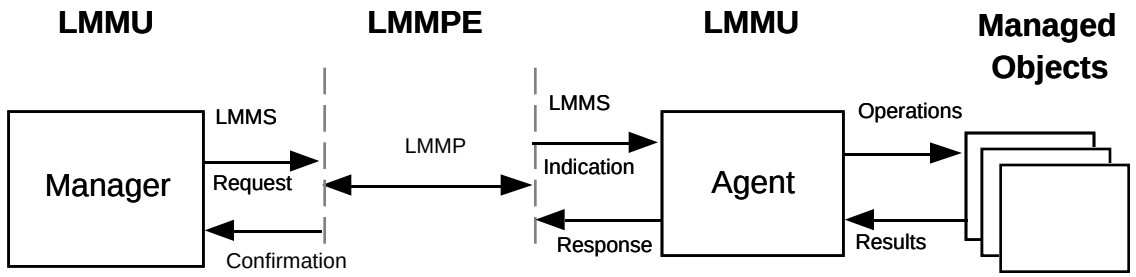
object to be applied to its attributes, or to the object as a whole, are specified in ISO/IEC 10165-1: 1993. These operations are as follows:

Get attribute value	(GET);	--attribute related
Replace attribute value	(REPLACE);	
Replace-with-default	(REPLACE-WITH-DEFAULT);	
Add member	(ADD);	
Remove member	(REMOVE);	
Create	(CREATE);	--object related
Delete	(DELETE);	
Action	(ACTION), and;	

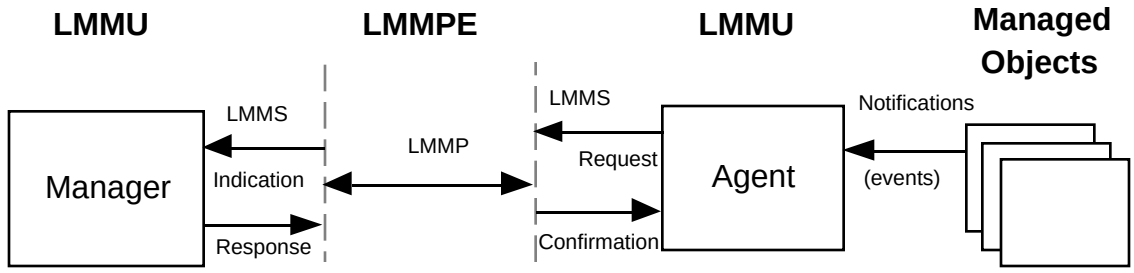
Notification.

The ISO/IEC 15802-2 LAN/MAN management services and the ISO CMIS/CMIP services that support these operations are as follows:

M_CREATE	Confirmed	
M_DELETE	Confirmed	
M_GET	Confirmed	
M_SET	Confirmed/nonconfirmed	
M_CANCEL_GET	Confirmed	
M_ACTION	Confirmed/nonconfirmed	
M_EVENT_REPORT	Confirmed/nonconfirmed	--notification event.



a) Manager requests



b) Managed system notifications

Figure 2-15—ISO/IEC 15802-2: 1995 management interactions (remote)

A managed system will complete one management operation before beginning the next operation. In addition to returning any data at the completion of the management operation, the agent will respond with an indication of the success or failure of that operation.

2.8.4.4 Management information

The management information required for the management of the SDE sublayer fall into the following categories:

a) Station information
b) Agent and managed object information
c) Event notification routing information
d) Station access control information
e) Private information

2.8.4.5 Managed object model

An information model is used to represent management's view of those elements of the SDE sublayer that are subject to management operations. The information model defines the abstraction of those elements as managed objects. Managed objects are defined by their ATTRIBUTES, their emitted NOTIFICATIONS, the management OPERATIONS that may be applied to them or their attributes, and their exhibited behavior in response to those management operations.

2.8.4.6 Managed object knowledge

For the information contained in the managed objects of a system to meet the requirements of extensibility, it is necessary to provide basic management functionality. Each managed object contains all the information needed or the purposes of management access. The following attributes are defined for all managed objects to meet this capability:

a) Name
b) Object Class
c) Name Bindings
d) Packages

2.8.4.7 Managed object templates

Templates are used to document the specification of a managed object class. These templates are specified in ISO/IEC 10165-4: 1992. The definition of a managed object class, as specified by a template, consists of the following:

a) Class naming
b) ATTRIBUTES
c) System management operations
d) BEHAVIOUR
e) Mandatory and conditional PACKAGES
f) ACTIONS and NOTIFICATIONS
g) Class position in the inheritance hierarchy

2.8.4.8 Managed object class naming

Managed objects that share the same definitions form a managed object class. Managed object classes are related together by inheritance. The managed object class "TOP" forms the top (the superclass) of an inheritance tree from which the other managed object classes (subclasses) are derived. The hierarchical arrangement of classes is based on refinement that includes inheritance of ATTRIBUTES, ACTIONS, and NOTIFICATIONS from the class from which they have been derived, together with extensions that can add new ATTRIBUTES, ACTIONS, NOTIFICATIONS, or BEHAVIOUR.

Each managed object class is registered and has a unique Object Identifier (OID) that is a leaf of a registration hierarchy of Managed Object Class Identifiers.

2.8.4.9 Managed object name binding

Instances of managed objects exist in a hierarchical structure representing object containment. The containment structure of managed object instances is coupled with specific instance naming. This distinguishing information is reflected by a managed object's distinguishing (naming) attribute and is referred to as that object's Relative Distinguished Name (RDN). All instances of managed object instances have a unique distinguished name formed by the hierarchical chain of Relative Distinguished Names.

A name binding template defines the naming of a managed object instance. This template identifies the managed object class being named and defines the RDN that shall be used to name instances of the class in the context of a particular superior class. This template also provides for the specification of relationships that exist between two managed objects as the consequence of a particular name binding.

2.8.4.10 Managed object packages

The GDMO (ISO/IEC 10165-4: 1992) makes provision for the grouping of ATTRIBUTES, ACTIONS, and NOTIFICATIONS in implementation "groups" or PACKAGES within a managed object class. This standard groups sets of ATTRIBUTES and NOTIFICATIONS together with their BEHAVIOUR definition into PACKAGES. Within each managed object, a PACKAGE has been set up to contain all the ATTRIBUTES that SDE requires to be mandatory. Other PACKAGES have been set up for groupings that can be used for functions, such as buffer management that may not be necessary in all implementations. These extra PACKAGES are identified as optional.

This standard defines for the SDE processing entity those managed object classes containing a mandatory PACKAGES and optional PACKAGES that are referenced by the SDE functions.

The management of the SDE sublayer of the Data link layer is achieved by actions and settings of a set of managed objects. The following subclauses define these managed objects along with their BEHAVIOUR, ATTRIBUTES, and NOTIFICATIONS, in template form.

2.8.5 SDE sublayer management entity definitions

This standard uses the concepts of management information and managed objects as expressed in ISO/IEC 10165-1: 1993 and ISO/IEC 10165-4: 1992.

The following set of managed object classes are defined for the purpose of managing the SDE portion of the IEEE 802 LLC sublayer:

a) **SDE_Station** managed object *(mandatory for SDE operations)*
b) **SDE_ResourceTypeId** managed object *(mandatory for SDE operations)*
c) **SDE_SAP** managed object *(at least one mandatory for SDE operations)*
d) **Security_Association** managed object *(at least one mandatory for SDE operations)*

These managed objects represent management's view of those elements that provide the security services at the LLC sublayer that are subject to management operations. The containment hierarchy is illustrated in Figure 2-16. Managed objects that can have multiple instances are represented by double boxes.

2.8.5.1 SDE_Station managed object class description

This managed object class is used as the primary SDE sublayer management entity. Within this managed object, the main SDE environment control variables are represented as manageable attributes. The mandatory PACKAGE holds those variables that are mandatory, while an optional (conditional) PACKAGE can be used for those systems that need buffer management control. Another conditional PACKAGE allows control and management of message fragmentation and reassembly.

2.8.5.2 SDE_ResourceTypeId managed object class description

A mandatory instance of this managed object class is linked by name binding to the SDE_Station managed object. This ResourceTypeID managed object class provides identification and the manufacturer's name, OUI, and product version number. The manufacturer's information is ReadOnly.

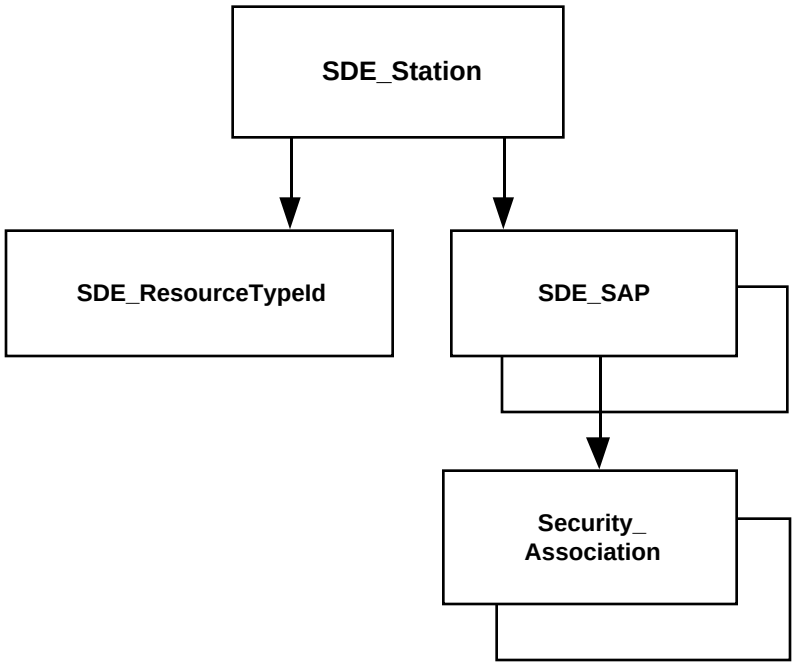


Figure 2-16—Entity diagram

2.8.5.3 SDE_SAP managed object class description

This managed object class can be used to manage the control information for handling the PDUs at the SAP interface. Each SAP used by the SDE protocol can be represented by an individual managed object with each instance having a unique name binding to the SDE_Station managed object instance.

A conditional PACKAGE is available for those systems that require management and control of the handling of oversize PDUs.

2.8.5.4 Security_Association managed object class description

Instances of this managed object class provide a means for management processes to set up and control such security associations. The operation of the SDE protocol between two open systems is controlled by the security_association's parameters. Each security association represents a single instance of those attributes (variables) needed for the operation of SDE. For each SDE_SAP managed object instance representing a "security" SAP, a set of Security_Association managed object instances, each with a unique name binding, can be used to manage and control each SDE security association using that SAP.

A conditional PACKAGE permits control and management of fragmentation by indicating whether the remote station of that security association pair supports reassembly of the incoming (fragmented) SDE PDUs using that remote security SAP. The two SDE_Station managed objects involved in the security association must contain the other conditional PACKAGE (remote_frag_package) with the attribute aStation_Fragmentation_Enabled set to TRUE before message fragmentation and reassembly is permitted.

2.8.6 SDE sublayer management managed object definitions

This subclause defines the managed object classes associated with the SDE sublayer management and their ATTRIBUTES and BEHAVIOUR. Unless specified otherwise, support of these managed objects is mandatory for stations supporting SDE.

Templates are used to define managed object classes. The template structure is defined in ISO/IEC 10165-4: 1992. A set of ASN.1 encodings required by these managed object class definitions appears in Annex 2F.

All information types contained within this standard are registered under the following object identifier root:

sils = iso(1) member-body(2) us(840) ieee-802dot10(10022)

This object identifier root is a sequence of registration arcs that forms the head part of all the object identifiers of the registered MANAGED OBJECT CLASSES, ATTRIBUTES, BEHAVIOURS, NOTIFICATIONS, PACKAGES, PARAMETERS, PACKAGES, and NAME BINDINGS defined by this standard.

2.8.6.1 Labeling name conventions

The labeling conventions used in this standard for naming templates are as follows:

a,	lowercase a is used to prefix the names supplied to the ATTRIBUTES Template;
b,	lowercase b is used to prefix the name applied to the BEHAVIOUR Template;
n,	lowercase n is used to prefix the names supplied to the NOTIFICATION Template;
nb,	lowercase nb is used to prefix the name applied to the NAME BINDING Template;
p,	lowercase p is used to prefix the name applied to the PACKAGE Template;
pa,	lowercase pa is used to prefix the name applied to the PARAMETER Template; and,
o,	lowercase o is used to prefix the name applied to the MANAGED OBJECT CLASS Template.

Also, the registration tree has been partitioned in an organized fashion. The common object identifier root sils has an organized set of arc extensions. These arc extensions follow the order that is used in other registration trees for managed objects within IEEE Project 802:

standardSpecificExtension(0);
asn1Module(2);
managedObjectClass(3);
package(4);
parameter(5);
nameBinding(6);
attribute(7);
attributeGroup(8);
action(9); and,
notification(10).

2.8.6.2 Packages

The ATTRIBUTES, NOTIFICATIONS, and BEHAVIOUR aspects of the managed object classes have been partitioned into PACKAGES. These PACKAGES are useful groupings that enable statements to be made about each collection as a whole.

One use of packaging is to distinguish between groups that are mandatory and those that are optional. For each managed object class, there can be mandatory PACKAGES or conditional PACKAGES. For example, for the SDE_SAP managed object, there is a conditional PACKAGE for oversize PDUs. Fragmentation requires two additional PACKAGES and the existence of the SDE_SAP PACKAGES. If fragmentation is to be supported, it is strongly recommended that the approach described in Annex 2E be used to promote interoperability. Fragmentation support is not mandatory.

SDE sublayer management can be configured one of two ways. The first is a static configuration where all the managed objects are created at start-up (or initialization) time. These “statically created” managed

objects cannot be deleted nor can other managed objects be added. The attributes of these managed objects may have their values read, reset, or changed by remote management operations acting through a local management agent.

For a static configuration of the SDE sublayer, the following name bindings are used: **nbstation-system**, **nbSDE_ResourceTypeId**, **nbSDE_SAP-SDE_Station**, and **nbSecurity_Association-SDE_SAP**. Additionally, the PACKAGES **pdynamic_SDE_package**, **pdynamic_SAP_package** and **pdynamic_SA_package** are not permitted.

The second is a dynamic configuration where the managed objects can be created or deleted by management action. For this way, there are a set of dynamic PACKAGES and a different set of name bindings. The dynamic PACKAGES (optional) are coupled with an associated name binding. The pairings are as follows:

pdynamic_SDE_package with **nbstation-system2** name binding;
pdynamic_SAP_package with **nbSDE_SAP-SDE_Station2** name binding, and;
pdynamic_SA_package with **nbSecurity_Association-SDE_SAP2** name binding.

Dynamic configuration can be set up either with all dynamic PACKAGES and dynamic name bindings, or with either of two subsets that include static bindings.

One subset uses a mixed configuration that has static name bindings **nbstation-system** and **nbSDE_ResourceTypeID** and dynamic **nbSDE_SAP-SDE_Station2** and **nbSecurity_Association-SDE_SAP2** name bindings with their associated PACKAGES **pdynamic_SAP_package** and **pdynamic_SA_package**.

The other dynamic subset uses a mixed configuration that has static name bindings **nbstation-system**, **nbSDE_ResourceTypeId** and **nbSDE_SAP-SDE_Station** and the dynamic name binding **nbSecurity_Association-SDE_SAP2** with its associated PACKAGE **pdynamic_SA_package**.

2.8.6.3 SDE_Station managed object class

```
oSDE_Station                    MANAGED OBJECT CLASS

    DERIVED FROM                "CCITT Rec. X.721 | ISO/IEC 10165-2: 1992":top ;
    CHARACTERIZED BY
      pstation_package          PACKAGE

        BEHAVIOUR
          bstation_package      BEHAVIOUR
            DEFINED AS          !This is the set of attributes that control the SDE Station's
                                environment for processing of security messages, "clear header"
                                security messages, and management-defined options.! ;
          ;
        ATTRIBUTES      aSDE_Station_Name          GET,          -- Naming Attribute
                        aStation_Clear_Hdr         GET-REPLACE,
                        aStation_MDF               GET-REPLACE,
                        aBadPDUsCount              GET-REPLACE,
                        aBadPDUsThreshold          GET-REPLACE
        ;

        NOTIFICATIONS           nBadPDUsDiscarded;

      REGISTERED AS             { sils package(4) station_package(1) } ;
    ;
    CONDITIONAL PACKAGES        pbuffer_package
        PRESENT IF              !an instance supports it.! ,
                                psde_frag_package
        PRESENT IF              !an instance supports it and remote_frag_package.! ,
                                pdynamic_SDE_package
```

```
            PRESENT IF          !SDE_Station managed objects are created and
                                deleted by management action dynamically.!
                                psde_security_label_package
            PRESENT IF          !an instance supports it and security_label_package! ,

    ;
REGISTERED AS                   { sils managedObjectClass(3) SDE_Station(1) } ;

pbuffer_package                 PACKAGE

    BEHAVIOUR
        bbuffer_package         BEHAVIOUR
            DEFINED AS          !This sets up and controls the buffer management of security
                                message processing.! ;
    ;
    ATTRIBUTES                  aBufferSize                 GET-REPLACE,
                                amaxBufferUseSize           GET-REPLACE,
                                aAvgBufferUseSize           GET-REPLACE,
                                aBufferProblemsCount        GET-REPLACE,
                                aBufferProblemsThreshold    GET-REPLACE
    ;
    NOTIFICATIONS               nBufferProblemsEvent;
    ;
REGISTERED AS                   { sils package(4) buffer_package(2) } ;
psde_frag_package               PACKAGE

    BEHAVIOUR
        bsde_frag_package       BEHAVIOUR
            DEFINED AS          !These are the additions needed for stations that permit
                                fragmentation of security messages.! ;
    ;
    ATTRIBUTES          aStation_Fragmentation_Enabled          GET-REPLACE,
                        aStation_Max_MAC_SDU_Size               GET-REPLACE,
                        aStation_Reassembly_Timer               GET-REPLACE,
                        aStation_Reassembly_Expiration_Count    GET-REPLACE,
                        aStation_Receive_Fragment               GET-REPLACE
    ;
REGISTERED AS               { sils package(4) sde_frag_package(3) } ;

pdynamic_SDE_package        PACKAGE
    BEHAVIOUR
        bdynamic_SDE_package    BEHAVIOUR
            DEFINED AS          !This package adds the notifications that are emitted when an
                                SDE_Station managed object is created or destroyed by management
                                action.! ;
    ;
    NOTIFICATIONS
                                "Rec X.721|ISO/IEC 10165-2":objectCreation,
                                "Rec X.721|ISO/IEC 10165-2":objectDeletion
    ;
REGISTERED AS                   { sils package(4) dynamic_SDE_package(8) };

psde_security_label_package     PACKAGE
    BEHAVIOUR
        bsde_security_label_package BEHAVIOUR
            DEFINED AS          !The station supports/requires the use of security labels! ;
    ;
    ATTRIBUTES
        aStation_Security_Label_Enabled             GET-REPLACE,
        aStation_Security_Label_Sets_Allowed        GET-REPLACE,
        aStation_Security_Label_Values              GET-REPLACE
    ;
REGISTERED AS                   { sils package(4) sde_security_label_package(11) } ;
```

2.8.6.3.1 Specification of SDE_Station name binding

nbstation-system NAME BINDING

SUBORDINATE OBJECT CLASS **oSDE_Station** AND SUBCLASSES ;
NAMED BY SUPERIOR OBJECT CLASS **"ISO/IEC 10165-2":system** AND SUBCLASSES ;
WITH ATTRIBUTE **aSDE_Station_Name** ;
BEHAVIOUR
bstation-system BEHAVIOUR
DEFINED AS **!A single instance of the SDE_Station managed object class exists within the superior object class.!** ;
;
REGISTERED AS { **sils nameBinding(6) station-system(1)** };

nbstation-system2 NAME BINDING

SUBORDINATE OBJECT CLASS **oSDE_Station** AND SUBCLASSES ;
NAMED BY SUPERIOR OBJECT CLASS **"ISO/IEC 10165-2":system** AND SUBCLASSES ;
WITH ATTRIBUTE **aSDE_Station_Name** ;
BEHAVIOUR
bstation-system2 BEHAVIOUR
DEFINED AS **!A single instance of the SDE_Station managed object class exists within the superior object class. It can be created and deleted dynamically by management action.!** ;
;
CREATE **paCreateError1**;
DELETE DELETES-CONTAINED-OBJECTS;
REGISTERED AS { **sils nameBinding(6) station-system2(5)** };

2.8.6.3.2 Specification of SDE_Station attributes

aSDE_Station_Name ATTRIBUTE
WITH ATTRIBUTE SYNTAX **IEEE802.10_SDE_ASN1Module.Station_name** ;
MATCHES FOR EQUALITY ;
BEHAVIOUR
bSDE_Station_NameBEHAVIOUR
DEFINED AS **!This attribute is used to name a single instance of the SDE_Station managed object class that exists within a superior object class. The value of this name is fixed and is equal to string "SDE_Station".!** ;
;
REGISTERED AS { **sils attribute(7) sde_station_name(1)** } ;

aStation_Clear_Hdr ATTRIBUTE
WITH ATTRIBUTE SYNTAX **IEEE802.10_SDE_ASN1Module.SDE_Boolean**;
MATCHES FOR EQUALITY ;
BEHAVIOUR
bStation_Clear_Hdr BEHAVIOUR
DEFINED AS **!--Subclause 2.6.3.1-a of IEEE Std 802.10-1998, Clause 2--!** ;
;
REGISTERED AS { **sils attribute(7) station_clear_header(2)** } ;

aStation_MDF ATTRIBUTE
WITH ATTRIBUTE SYNTAX **IEEE802.10_SDE_ASN1Module.SDE_Boolean**;
MATCHES FOR EQUALITY ;
BEHAVIOUR
bStation_MDF BEHAVIOUR
DEFINED AS **!--Subclause 2.6.3.1-b of IEEE Std 802.10-1998, Clause 2--!** ;
;
REGISTERED AS { **sils attribute(7) station_mdf(3)** } ;

aStation_Fragmentation_Enabled ATTRIBUTE
WITH ATTRIBUTE SYNTAX **IEEE802.10_SDE_ASN1Module.SDE_Boolean**;

```
        MATCHES FOR EQUALITY ;
        BEHAVIOUR
            bStation_Fragmentation_Enabled          BEHAVIOUR
                DEFINED AS          !--Subclause 2E.3 of Annex 2E: Fragmentation of IEEE Std 802.10-
                                    1998, Clause 2--! ;
        ;
REGISTERED AS                       { sils attribute(7) fragmentation_enable(4) } ;

aStation_Max_MAC_SDU_Size           ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       IEEE802.10_SDE_ASN1Module.SDE_Integer;
        MATCHES FOR EQUALITY ;
        BEHAVIOUR
            bStation_Max_MAC_SDU_Size           BEHAVIOUR
                DEFINED AS          !The maximum MAC SDU size in octets that is permitted by the
                                    MAC sublayer for this station.! ;
        ;
REGISTERED AS                       { sils attribute(7) max_mac_sdu_size(5) } ;

aStation_Reassembly_Timer           ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       IEEE802.10_SDE_ASN1Module.SDE_Integer;
        MATCHES FOR EQUALITY ;
        BEHAVIOUR
            bStation_Reassembly_Timer   BEHAVIOUR
                DEFINED AS          !The number of milliseconds a station will store a received SDE PDU
                                    that contains a fragment of an SDE PDU.! ;
        ;
REGISTERED AS                       { sils attribute(7) station_reassembly_timer(6) } ;

aStation_Reassembly_Expiration_Count ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       IEEE802.10_SDE_ASN1Module.SDE_Integer;
        MATCHES FOR EQUALITY ;
        BEHAVIOUR
            bStation_Reassembly_Expiration_Count    BEHAVIOUR
                DEFINED AS          !The number of SDE PDUs that have been discarded by the SDE
                                    station when the SDE reassembly timer has expired.! ;
        ;
REGISTERED AS                       { sils attribute(7) station_reassembly_expiration_count(7) } ;

aStation_Receive_Fragment           ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       IEEE802.10_SDE_ASN1Module.SDE_Integer;
        MATCHES FOR EQUALITY ;
        BEHAVIOUR
            bStation_Receive_Fragment   BEHAVIOUR
                DEFINED AS          !The number of SDE PDUs that contain SDE SDU fragments
                                    received by this SDE Station.! ;
        ;
REGISTERED AS                       { sils attribute(7) station_receive_fragment(8) } ;

aBufferSize                         ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       IEEE802.10_SDE_ASN1Module.SDE_Integer;
        MATCHES FOR EQUALITY ;
        BEHAVIOUR
            bBufferSize             BEHAVIOUR
                DEFINED AS          !The average amount of total buffer space in octets for use in this
                                    station. This value can be read and set by system management
                                    action.! ;
        ;
REGISTERED AS                       { sils attribute(7) buffersize(9) } ;

amaxBufferUseSize                   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       IEEE802.10_SDE_ASN1Module.SDE_Integer;
        MATCHES FOR EQUALITY ;
        BEHAVIOUR
                bmaxBufferUseSize   BEHAVIOUR
```



```
                DEFINED AS          !The maximum amount of buffer space in octets in use at the same
                                    time. This value can be read and set by system management action.! ;
        ;
REGISTERED AS                       { sils attribute(7) maxbufferusesize(10) }

aAvgBufferUseSize                   ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       IEEE802.10_SDE_ASN1Module.SDE_Integer;
        MATCHES FOR EQUALITY ;
        BEHAVIOUR
            bAvgBufferUseSize       BEHAVIOUR
                DEFINED AS          !The average amount of buffer space in octets in use at the same time.
                                    This value can be read and set by system management action.! ;
        ;
REGISTERED AS                       { sils attribute(7) avgbufferusesize(11) } ;

aBadPDUsCount                       ATTRIBUTE
        DERIVED FROM`               "ISO/IEC 10165-2":counter;
        BEHAVIOUR
            bBadPDUsCount           BEHAVIOUR
                DEFINED AS          !This attribute is a counter that increments each time an SDE PDU
                                    that has no valid security association is received. It is associated with
                                    Bad PDUs Threshold and Bad PDUs Discarded. This value can be
                                    read and set by system management action.!;
        ;
REGISTERED AS                       { sils attribute(7) badpduscount(41) } ;

aBadPDUsThreshold                   ATTRIBUTE
        DERIVED FROM                "ISO/IEC 10165-2":counter-Threshold;
        BEHAVIOUR
            bBadPDUsThreshold       BEHAVIOUR
                DEFINED AS          !This attribute is associated with Bad PDUs Count and Bad PDUs
                                    Discarded. The triggering of this threshold by the associated Bad
                                    PDUs count causes the generation of a Bad PDUs Discarded event.
                                    This value can be read and set by system management action.! ;
        ;
REGISTERED AS{ sils attribute(7) badpdusthreshold(42) } ;

aBufferProblemsCount                ATTRIBUTE
        DERIVED FROM                "ISO/IEC 10165-2":counter;
        BEHAVIOUR
            bBufferProblemsCount    BEHAVIOUR
                DEFINED AS          !This attribute is a counter that increments each time an SDE PDU is
                                    discarded due to buffer limitations. It is associated with Buffer
                                    Problems Threshold and Buffer Problems Event. This value can be
                                    read and set by system management action.!;
        ;
REGISTERED AS                       { sils attribute(7) bufferproblemscount(43) } ;

aBufferProblemsThreshold            ATTRIBUTE
        DERIVED FROM                "ISO/IEC 10165-2":counter-Threshold;
        BEHAVIOUR
            bBufferProblemsThreshold BEHAVIOUR
                DEFINED AS          !This attribute is associated with Buffer Problems Count and Buffer
                                    Problems Event. The triggering of this threshold by the associated
                                    Buffer Problems Count causes the generation of a Buffer Problems
                                    Event. This value can be read and set by system management
                                    action.! ;
        ;
REGISTERED AS                       { sils attribute(7) bufferproblemsthreshold(44) } ;
aStation_Security_Label_Enabled     ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       IEEE802.10_SDE_ASN1Module.SDE_ Boolean;
        MATCHES FOR EQUALITY;
        BEHAVIOUR
            bStation_Security_Label_Enabled  BEHAVIOUR
```

```
                DEFINED AS          !This attribute indicates whether the use of security labels is
                                    authorized on the station.! ;
        ;
REGISTERED AS                       { sils attribute(7) station_security_label_enabled(45) } ;

aStation_Security_Label_Sets_Allowed    ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       IEEE802.10_SDE_ASN1Module.SLabel_ Sets;
        MATCHES FOR EQUALITY;
        BEHAVIOUR
            bStation_Security_Label_Sets_Allowed    BEHAVIOUR
                DEFINED AS          !This attribute conveys an ordered list of object identifiers that
                                    indicate, in decreasing order of preference, the label sets supported
                                     by the station.! ;
        ;
REGISTERED AS                       {sils attribute(7) station_security_label_sets_allowed(46)} ;

aStation_Security_Label_Values      ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       IEEE802.10_SDE_ASN1Module.Label_Values;
        MATCHES FOR EQUALITY;
        BEHAVIOUR
            bStation_Security_Label_Values      BEHAVIOUR
                DEFINED AS          !This attribute carries the set of possible security information
                                     values that can be accepted by the SDE station! ;
        ;
 REGISTERED AS                      { sils attribute(7) station_security_label_values(47) } ;
```

2.8.6.3.3 Specification of SDE_Station notifications

```
nBufferProblemsEvent                NOTIFICATION
        BEHAVIOUR
            bBufferProblemsEvent    BEHAVIOUR
                    DEFINED AS      !This notification is sent when the Buffer Problems Count reaches
                                    the Buffer Problems Threshold value.  The following information
                                    may optionally be included in the notification message; the Buffer
                                    Problems Count. ! ;
        ;
        MODE                        CONFIRMED AND NONCONFIRMED;
        WITH INFORMATION SYNTAX     Notification-ASN1Module.AttributeValueChangeInfo
                                    AND ATTRIBUTEIDS ;
REGISTERED AS                       { sils notification(10) bufferproblemsevent(1) } ;

nBadPDUsDiscarded                   NOTIFICATION
        BEHAVIOUR
            bBadPDUsDiscarded       BEHAVIOUR
        DEFINED AS                  !This notification is sent when the Bad PDUs Count reaches the Bad
                                    PDUs Threshold value.  The following information may optionally be
                                    included in the notification message; the Bad PDUs Count and/or the
                                    last incoming PDU with an invalid SAID field.! ;
        ;
        MODE                        CONFIRMED AND NONCONFIRMED;
        WITH INFORMATION SYNTAX     Notification-ASN1Module.AttributeValueChangeInfo
                                    AND ATTRIBUTEIDS ;
REGISTERED AS                       { sils notification(10) badpdusdiscarded(5) } ;
```

2.8.6.3.4 Specification of SDE_Station parameters

```
paCreateError1                      PARAMETER
        CONTEXT                     SPECIFIC-ERROR ;
        WITH SYNTAX                 IEEE802.10_SDE_ASN1Module.CreateError1 ;
        BEHAVIOUR
                bCreateError1       BEHAVIOUR
```

DEFINED AS! **This CreateError1 information is returned when a managed object cannot be created. The error information is placed in the SpecificErrorInfo and is of the form**
SpecificErrorInfo ::= SEQUENCE {
errorid OBJECT IDENTIFIER
errorinfo CreateError1 }
The OBJECT IDENTIFIER will be that which this parameter is registered under and errorinfo will have the syntax of CreateError1 containing the error reason. ! ;

;

REGISTERED AS { **sils parameter(5) createerror1(1)** } ;

2.8.6.4 SDE_ResourceTypeId managed object class

A single instance of the SDE_ResourceTypeId managed object is attached to an instance of the SDE_Station managed object. This managed object is of the generic Resource Type ID managed object class for use by the SDE sublayer to associate manufacturer information with an instance of the SDE_Station.

The managed object class definition of a Resource Type ID is contained in IEEE Std 802.1F-1993; therefore, only the name binding appears in this standard. However, implementation of the managed object in accordance with the definition contained in IEEE Std 802.1F-1993 is an IEEE 802.1 conformance requirement. The SDE_ResourceTypeId managed object class contains manufacturer and product information related to the implementation of the management functionality.

Support for this managed object is mandatory.

2.8.6.4.1 Specification of SDE_ResourceTypeId name binding

nbSDE_ResourceTypeId NAME BINDING

SUBORDINATE OBJECT CLASS **"IEEE Std. 802.1F":oResourceTypeID** AND SUBCLASSES ;
NAMED BY SUPERIOR OBJECT CLASS **oSDE_Station** AND SUBCLASSES ;
WITH ATTRIBUTE **“IEEE Std 802.1F-1993”:aResourceTypeIDName** ;
BEHAVIOUR
bSDE_ResourceTypeIdnamebinding BEHAVIOUR;
DEFINED AS **!A single instance of Resource Type ID managed object class exists within any instance of SDE_Station. It cannot be created nor deleted dynamically by management action.!** ;

;

REGISTERED AS { **sils nameBinding(6) sde_resourcetypeId(2)** } ;

2.8.6.5 SDE_SAP managed object class

oSDE_SAP MANAGED OBJECT CLASS

DERIVED FROM **“CCITT Rec. X.721 | ISO/IEC 10165-2: 1992”:top** ;
CHARACTERIZED BY
pSDE_SAP_package PACKAGE

BEHAVIOUR
bSDE_SAP_package BEHAVIOUR
DEFINED AS! **This package holds the basic environment constants for handling PDU’s at the SAP interface.!** ;
;
ATTRIBUTES **aSAP_ID** GET-REPLACE, --naming
aSAP_Worst_Case_Expansion GET-REPLACE,
aSAP_Max_SDE_SDU GET-REPLACE
;
REGISTERED AS { **sils package(4) sde_sap_package(4)** } ;

```
        ;
        CONDITIONAL PACKAGES        poversize_pdu_package
                PRESENT IF          !an instance supports it.! ,
                                    pdynamic_SAP_package
                PRESENT IF          !SDE_SAP managed objects are created and deleted by management
                                    action dynamically.!
        ;
REGISTERED AS                       { sils managedObjectClass(3) SDE_SAP(2) } ;

poversize_pdu_package               PACKAGE

        BEHAVIOUR
                boversize_pdu_package   BEHAVIOUR
                        DEFINED AS      !This is an optional package for management of oversize PDU
                                        handling.! ;
        ;
        ATTRIBUTES          aSAP_Exceed_Size_Cnt        GET-REPLACE,
                            aSAP_Exceed_Size_Thres      GET-REPLACE,
                            aSAP_Max_Oversize_Short     GET-REPLACE,
                            aSAP_Max_Oversize_Long      GET-REPLACE
        ;
        NOTIFICATIONS               nSAP_Exceed_Size_Event;

REGISTERED AS                       { sils package(4) oversize_pdu_package(5) };

pdynamic_SAP_package                PACKAGE
        BEHAVIOUR
                bdynamic_SAP_package    BEHAVIOUR
                        DEFINED AS      !This package adds the notifications that are emitted when an
                                        SDE_SAP managed object is created or destroyed by management
                                        action.! ;
        ;
        NOTIFICATIONS
                                        "Rec X.721|ISO/IEC 10165-2":objectCreation,
                                        "Rec X.721|ISO/IEC 10165-2":objectDeletion;
        ;
REGISTERED AS                       { sils package(4) dynamic_SAP_package(9) };
```

2.8.6.5.1 Specification of SDE_SAP name binding

```
nbSDE_SAP-SDE_Station                       NAME BINDING

        SUBORDINATE OBJECT CLASS            oSDE_SAP ;
        NAMED BY SUPERIOR OBJECT CLASS      oSDE_Station ;
        WITH ATTRIBUTE                      aSAP_ID ;
        BEHAVIOUR
                bSDE_SAP-SDE_Station    BEHAVIOUR
                        DEFINED AS      !A single instance of SDE_SAP managed object class exists within
                                        any instance of the SDE_Station for each SDE sublayer SAP.! ;
        ;
REGISTERED AS                           { sils nameBinding(6) sde_sap-sde_station(3) };

nbSDE_SAP-SDE_Station2                      NAME BINDING

        SUBORDINATE OBJECT CLASS            oSDE_SAP ;
        NAMED BY SUPERIOR OBJECT CLASS      oSDE_Station ;
        WITH ATTRIBUTE                      aSAP_ID ;
        BEHAVIOUR
                bSDE_SAP-SDE_Station2   BEHAVIOUR
                        DEFINED AS      !A single instance of SDE_SAP managed objectclass exists within any
                                        instance of the SDE_Station for each SDE sublayer SAP. The
                                        SDE_SAP managed object instances are created or deleted
                                        dynamically by system management action.! ;
        ;
```

```
        CREATE          paCreateError2;
        DELETE          DELETES-CONTAINED-OBJECTS;
REGISTERED AS                       { sils nameBinding(6) sde_sap-sde_station2(6) };
```

2.8.6.5.2 Specification of SDE_SAP attributes

```
aSAP_ID                             ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       IEEE802.10_SDE_ASN1Module.SAP_ID ;
        MATCHES FOR EQUALITY;
        BEHAVIOUR
                bSAP_ID             BEHAVIOUR
                        DEFINED AS  !Multiple instances of the SDE_SAP managed object class exist
                                    within a superior object class.! ;
        ;
REGISTERED AS                       { sils attribute(7) sap_ID(12) } ;

aSAP_Worst_Case_Expansion           ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       IEEE802.10_SDE_ASN1.SDE_Integer ;
        MATCHES FOR EQUALITY;
        BEHAVIOUR
                bSAP_Worst_Case_Expansion       BEHAVIOUR
                        DEFINED AS  !The maximum size in octets that an SDE SDU can be permitted to
                                    expand but still be accepted by the MAC sublayer for this SAP! ;
        ;
REGISTERED AS                       { sils attribute(7) sap_worst_case_expansion(13) } ;
        ;
aSAP_Max_SDE_SDU                    ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       IEEE802.10_SDE_ASN1Module.SDE_Integer ;
        MATCHES FOR EQUALITY;
        BEHAVIOUR
                bSAP_Max_SDE_SDU    BEHAVIOUR
                        DEFINED AS  !The maximum MAC SDU size in octets that is permitted by the
                                    MAC sublayer for this SDE SAP.! ;
        ;
REGISTERED AS                       { sils attribute(7) sap_max_SDE_sdu(14) } ;

aSAP_Exceed_Size_Cnt                ATTRIBUTE
        DERIVED FROM                "ISO/IEC 10165-2":counter;
        BEHAVIOUR
                bSAP_Exceed_Size_Cnt BEHAVIOUR
                        DEFINED AS  !This attribute is a counter that increments each time a MAC SDU
                                    exceeds the size permitted by the MAC sublayer for this SAP. It is
                                    associated with SAP_Exceed_Size_Thres and
                                    SAP_Exceed_Size_Event.  This value can be read and set by system
                                    management action.! ;
        ;
REGISTERED AS                       { sils attribute(7) sap_exceed_size_cnt(15) } ;

aSAP_Exceed_Size_Thres              ATTRIBUTE
        DERIVED FROM                "ISO/IEC 10165-2":counter-Threshold;
        BEHAVIOUR
                bSAP_Exceed_Size_Thres BEHAVIOUR
                        DEFINED AS  !This attribute is associated with the attributes,
                                    SAP_Exceed_Size_Cnt and SAP_Exceed_Size_Event. The triggering
                                    of this threshold by the SAP_Exceed_Size_Cnt causes the generation
                                    of a SAP_Exceed_Size_Event. This value can be read and set by
                                    system management action.! ;
        ;
REGISTERED AS                       { sils attribute(7) sap_exceed_size_thres(16) } ;

aSAP_Max_Oversize_Short             ATTRIBUTE
        WITH ATTRIBUTE SYNTAX       IEEE802.10_SDE_ASN1Module.SDE_Integer ;
        MATCHES FOR EQUALITY;
        BEHAVIOUR
```

bSAP_Max_Oversize_Short BEHAVIOUR
DEFINED AS **!The value in octets of the smallest SDE PDU that exceeds the maximum MAC SDU size permitted by the MAC sublayer for this station.! ;**
;
REGISTERED AS { **sils attribute(7) sap_max_oversize_short(17)** } ;

aSAP_Max_Oversize_Long ATTRIBUTE
WITH ATTRIBUTE SYNTAX **IEEE802.10_SDE_ASN1Module.SDE_Integer**;
MATCHES FOR EQUALITY;
BEHAVIOUR
bSAP_Max_Oversize_Long BEHAVIOUR
DEFINED AS **!The value in octets of the largest SDE PDU that exceeds the maximum MAC SDU size permitted by the MAC sublayer for this station.! ;**
;
REGISTERED AS { **sils attribute(7) sap_max_oversize_long(18)** } ;

2.8.6.5.3 Specification of SDE_SAP notifications

nSAP_Exceed_Size_Event NOTIFICATION
BEHAVIOUR
bSAP_Exceed_Size_Event BEHAVIOUR
DEFINED AS **!This notification is sent when the SAP_Exceed_Size_Cnt reaches the SAP_Exceed_Size_Thres. The following information may optionally be included in the notification message; the SAP_Exceed_Size_Cnt and/or the last MAC SDU that was too long to be handled at the SAP interface.!** ;
;
MODE CONFIRMED AND NONCONFIRMED ;
WITH INFORMATION SYNTAX;
REGISTERED AS { **sils notification(10) sap_exceed_size_event(3)** };

2.8.6.5.4 Specification of SDE_SAP parameters

paCreateError2 PARAMETER
CONTEXT SPECIFIC-ERROR ;
WITH SYNTAX **IEEE802.10_SDE_ASN1Module.CreateError2 ;**
BEHAVIOUR
bCreateError2 BEHAVIOUR
DEFINED AS **!This CreateError information is returned when a managed object cannot be created. The error information is placed in the SpecificErrorInfo and is of the form**
SpecificErrorInfo ::= SEQUENCE {
errorid OBJECT IDENTIFIER
errorinfo CreateError2 }
The OBJECT IDENTIFIER will be that which this parameter is registered under and errorinfo will have the syntax of CreateError2 containing the error reason.! ;
;
REGISTERED AS { **sils parameter(5) createerror2(2)** } ;

2.8.6.6 Security_Association managed object class

oSecurity_Association MANAGED OBJECT CLASS

DERIVED FROM **"CCITT Rec. X.721 | ISO/IEC 10165-2: 1992":top** ;
CHARACTERIZED BY
pSAID_Package PACKAGE
BEHAVIOUR
bSAID_Package BEHAVIOUR
DEFINED AS **!The security association controls the processing of the incoming and outgoing PDUs that pass through SDE.!** ;


```
            ;
            ATTRIBUTES
                  aSA_ID                                    GET,              --naming-
                  aLocal_SAID                               GET-REPLACE,
                  aRemote_SAID                              GET-REPLACE,
                  aAssoc_MDF                                GET-REPLACE,
                  aRemote_MDF                               GET-REPLACE,
                  aConfid                                   GET-REPLACE,
                  aInteg                                    GET-REPLACE,
                  aPadding_pres                             GET-REPLACE,
                  aID_pres                                  GET-REPLACE,
                  aConfid_Alg_ID                            GET-REPLACE,
                  aInteg_Alg_ID                             GET-REPLACE,
                  aSDE_SAP                                  GET-REPLACE,
                  aRemote_SDE                               GET-REPLACE,
                  aOutgoing_Source_MAC_Address              GET-REPLACE,
                  aOutgoing_Destination_MAC_Address         GET-REPLACE,
                  aIncoming_Destination_MAC_Address         GET-REPLACE,
                  aIncoming_Source_MAC_Address              GET-REPLACE ADD-REMOVE,
                  aBadPDUsICVCount                          GET-REPLACE,
                  aBadPDUsICVThreshold                      GET-REPLACE,
                  aBadPDUsSAIDCount                         GET-REPLACE,
                  aBadPDUsSAIDThreshold                     GET-REPLACE
            ;
            NOTIFICATIONS
                  nBadPDUsICVDiscarded,
                  nBadPDUsSAIDDiscarded
            ;
      REGISTERED AS        { sils package(4) said_package(6) } ;
;

CONDITIONAL PACKAGES       premote_frag_package
      PRESENT IF           !an instance supports it.!,
                           pdynamic_SA_package
      PRESENT IF           !security associations are created and deleted by management action
                           dynamically.!
                           psecurity_label_package
      PRESENT IF           !an instance supports it! ,
;
REGISTERED AS              { sils managedObjectClass(3) Security_Association(3) };

premote_frag_package       PACKAGE
      BEHAVIOUR
            bremote_frag_package    BEHAVIOUR
                  DEFINED AS        !The security association controls the fragmentation handling of the
                                    incoming PDUs that pass through SDE.! ;
      ;
      ATTRIBUTES
            aAssoc_Frag_Enab                            GET-REPLACE
      ;
REGISTERED AS              { sils package(4) remote_frag_package(7) };

pdynamic_SA_package        PACKAGE
      BEHAVIOUR
            bdynamic_SA_package     BEHAVIOUR
                  DEFINED AS        !This package adds the notifications that are emitted when a Security
                                    Association managed object is created or destroyed by management
                                    action.! ;
      ;
      NOTIFICATIONS
            "Rec X.721|ISO/IEC 10165-2":objectCreation,
            "Rec X.721|ISO/IEC 10165-2":objectDeletion
      ;
REGISTERED AS              { sils package(4) dynamic_SA_package(10) };
```

psecurity_label_package PACKAGE
BEHAVIOUR
bsecurity_label_package BEHAVIOUR
DEFINED AS **!The security association indicates whether a security label has been negotiated, its value or set of values to be accepted over the association, and, if a single value is accepted, whether the label value shall be carried on every SDE PDU.! ;**
;
ATTRIBUTES
aAssoc_Label_Set GET-REPLACE,
aAssoc_Label_Value GET-REPLACE,
aAssoc_Label_Explicit GET-REPLACE
;
REGISTERED AS { **sils package(4) security_label_package(12)** } ;

2.8.6.6.1 Specification of Security_Association name binding

nbSecurity_Association-SDE_SAP NAME BINDING
SUBORDINATE OBJECT CLASS **oSecurity_Association** ;
NAMED BY SUPERIOR OBJECT CLASS **oSDE_SAP** ;
WITH ATTRIBUTE **aSA_ID** ; --naming--
BEHAVIOUR
bSecurity_Association-SDE_SAP BEHAVIOUR
DEFINED AS **!A unique instance of the Security Association managed object class exists within any instance of the SDE_SAP for each security association that has been established.! ;**
;
REGISTERED AS { **sils nameBinding(6) security_association-sde_sap(4)** } ;

nbSecurity_Association-SDE_SAP2 NAME BINDING
SUBORDINATE OBJECT CLASS **oSecurity_Association** ;
NAMED BY SUPERIOR OBJECT CLASS **oSDE_SAP** ;
WITH ATTRIBUTE **aSA_ID** ; --naming--
BEHAVIOUR
bSecurity_Association-SDE_SAP2 BEHAVIOUR
DEFINED AS **!A unique instance of the Security Association managed object class exists within any instance of the SDE_SAP for each security association that has been established. The security association managed object instances are created or deleted dynamically by system management or key management action.! ;**
;
CREATE **paCreateError3**;
DELETE DELETES-CONTAINED-OBJECTS;
REGISTERED AS { **sils nameBinding(6) security_association-sde_sap2(7)** } ;

2.8.6.6.2 Specification of Security_Association attributes

aSA_ID ATTRIBUTE
WITH ATTRIBUTE SYNTAX **IEEE802.10_SDE_ASN1Module.SA_ID** ;
MATCHES FOR EQUALITY ;
BEHAVIOUR
bSA_ID BEHAVIOUR
DEFINED AS **!Multiple instances of the Security_Association managed object class exist within a superior object class. Each instance will be uniquely identified. !** ;
;
REGISTERED AS { **sils attribute(7) sa_ID(19)** } ;

aLocal_SAID ATTRIBUTE
WITH ATTRIBUTE SYNTAX **IEEE802.10_SDE_ASN1Module.SDE_Ostring** ;
MATCHES FOR EQUALITY;


```
        BEHAVIOUR
            bLocal_SAID                 BEHAVIOUR
                    DEFINED AS          !--Subclause 2.6.3.3-a of IEEE Std 802.10-1998, Clause 2--! ;
        ;
REGISTERED AS                           { sils attribute(7) local_SAID(20) } ;

aRemote_SAID                            ATTRIBUTE
        WITH ATTRIBUTE SYNTAX           IEEE802.10_SDE_ASN1Module.SDE_Ostring ;
        MATCHES FOR EQUALITY;
        BEHAVIOUR
                bRemote_SAID            BEHAVIOUR
                    DEFINED AS          !--Subclause 2.6.3.3-b of IEEE Std 802.10-1998, Clause 2--! ;
        ;
REGISTERED AS                           { sils attribute(7) remote_SAID(21) } ;

aAssoc_MDF                              ATTRIBUTE
        WITH ATTRIBUTE SYNTAX           IEEE802.10_SDE_ASN1Module.SDE_Boolean ;
        MATCHES FOR EQUALITY;
        BEHAVIOUR
                bAssoc_MDF              BEHAVIOUR
                    DEFINED AS          !--Subclause 2.6.3.3-c of IEEE Std 802.10-1998, Clause 2--! ;
        ;
REGISTERED AS                           { sils attribute(7) assoc_MDF(22) } ;

aRemote_MDF                             ATTRIBUTE
        WITH ATTRIBUTE SYNTAX           IEEE802.10_SDE_ASN1Module.SDE_Ostring ;
        MATCHES FOR EQUALITY;
        BEHAVIOUR
                bRemote_MDF             BEHAVIOUR
                    DEFINED AS          !--Subclause 2.6.3.3-c1 of IEEE Std 802.10-1998, Clause 2--! ;
        ;
REGISTERED AS                           { sils attribute(7) remote_MDF(23) } ;

aConfid                                 ATTRIBUTE
        WITH ATTRIBUTE SYNTAX           IEEE802.10_SDE_ASN1Module.SDE_Boolean ;
        MATCHES FOR EQUALITY;
        BEHAVIOUR
                bConfid                 BEHAVIOUR
                    DEFINED AS          !--Subclause 2.6.3.3-d1 of IEEE Std 802.10-1998, Clause 2--! ;
        ;
REGISTERED AS                           { sils attribute(7) confid(24) } ;

aInteg                                  ATTRIBUTE
        WITH ATTRIBUTE SYNTAX           IEEE802.10_SDE_ASN1Module.SDE_Boolean ;
        MATCHES FOR EQUALITY;
        BEHAVIOUR
                bInteg                  BEHAVIOUR
                    DEFINED AS          !--Subclause 2.6.3.3-d2 of IEEE Std 802.10-1998, Clause 2--! ;
        ;
REGISTERED AS                           { sils attribute(7) integ(25) } ;

aPadding_pres                           ATTRIBUTE
        WITH ATTRIBUTE SYNTAX           IEEE802.10_SDE_ASN1Module.SDE_Boolean ;
        MATCHES FOR EQUALITY;
        BEHAVIOUR
                bPadding_pres           BEHAVIOUR
                    DEFINED AS          !--Subclause 2.6.3.3-e1 of IEEE Std 802.10-1998, Clause 2--! ;
        ;
REGISTERED AS                           { sils attribute(7) padding_pres(26) } ;

aID_pres                                ATTRIBUTE
        WITH ATTRIBUTE SYNTAX           IEEE802.10_SDE_ASN1Module.SDE_Boolean ;
        MATCHES FOR EQUALITY;
        BEHAVIOUR
```

```
            bID_pres                    BEHAVIOUR
                DEFINED AS              !--Subclause 2.6.3.3-e2 of IEEE Std 802.10-1998, Clause 2--! ;
    ;
REGISTERED AS                           { sils attribute(7) ID_pres(27) } ;

aConfid_Alg_ID                          ATTRIBUTE
    WITH ATTRIBUTE SYNTAX               IEEE802.10_SDE_ASN1Module.SDE_Ostring ;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
            bConfid_Alg_ID              BEHAVIOUR
                DEFINED AS              !--Subclause 2.6.3.3-f of IEEE Std 802.10-1998, Clause 2--! ;
    ;
REGISTERED AS                           { sils attribute(7) confid_alg_ID(28) } ;

aInteg_Alg_ID                           ATTRIBUTE
    WITH ATTRIBUTE SYNTAX               IEEE802.10_SDE_ASN1Module.SDE_Ostring ;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
            bInteg_Alg_ID               BEHAVIOUR
                DEFINED AS              !--Subclause 2.6.3.3-g of IEEE Std 802.10-1998, Clause 2--! ;
    ;
REGISTERED AS                           { sils attribute(7) integ_alg_ID(29) } ;

aSDE_SAP                                ATTRIBUTE
    WITH ATTRIBUTE SYNTAX               IEEE802.10_SDE_ASN1Module.SDE_Ostring ;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
            bSDE_SAP                    BEHAVIOUR
                DEFINED AS              !--Subclause 2.6.3.3-h of IEEE Std 802.10-1998, Clause 2--! ;
    ;
REGISTERED AS                           { sils attribute(7) sde_sap(30) } ;

aRemote_SDE                             ATTRIBUTE
    SINGLE-VALUED
    WITH ATTRIBUTE SYNTAX               IEEE802.10_SDE_ASN1Module.SDE_Boolean ;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
            bRemote_SDE                 BEHAVIOUR
                DEFINED AS              !--Subclause 2.6.3.3-i of IEEE Std 802.10-1998, Clause 2--! ;
    ;
REGISTERED AS                           { sils attribute(7) remote_sde(31) } ;

aOutgoing_Source_MAC_Address            ATTRIBUTE
    WITH ATTRIBUTE SYNTAX               IEEE802CommonDefinitions.MACAddress ;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
            bOutgoing_Source_MAC_Address    BEHAVIOUR
                DEFINED AS              !--Subclause 2.6.3.3-j of IEEE Std 802.10-1998, Clause 2--! ;
    ;
REGISTERED AS                           { sils attribute(7) outgoing_source_mac_address(32) } ;

aOutgoing_Destination_MAC_Address       ATTRIBUTE
    WITH ATTRIBUTE SYNTAX               IEEE802CommonDefinitions.MACAddress ;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
            bOutgoing_Destination_MAC_Address                   BEHAVIOUR
                DEFINED AS              !--Subclause 2.6.3.3-k of IEEE Std 802.10-1998, Clause 2--! ;
;
REGISTERED AS                           { sils attribute(7) outgoing_destination_mac_address(33) } ;

aIncoming_Destination_MAC_Address       ATTRIBUTE
    WITH ATTRIBUTE SYNTAX               IEEE802CommonDefinitions.MACAddress ;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION ;
    BEHAVIOUR
```



```
            bIncoming_Destination_MAC_Address                BEHAVIOUR
                DEFINED AS      !---Subclause 2.6.3.3-l of IEEE Std 802.10-1998, Clause 2--! ;
        ;
REGISTERED AS                   { sils attribute(7) incoming_destination_mac_address(35) } ;

aIncoming_Source_MAC_Address    ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   IEEE802.10_SDE_ASN1Module.MACAddresses ;
        MATCHES FOR EQUALITY;
        BEHAVIOUR
            bIncoming_Source_MAC_Address    BEHAVIOUR
                DEFINED AS      !--Subclause 2.6.3.3-m of IEEE Std 802.10-1998, Clause 2--! ;
        ;
REGISTERED AS   { sils attribute(7) incoming_source_mac_address(34) } ;

aAssoc_Frag_Enab                ATTRIBUTE
        WITH ATTRIBUTE SYNTAX   IEEE802.10_SDE_ASN1Module.SDE_Boolean ;
        MATCHES FOR EQUALITY;
        BEHAVIOUR
            bAssoc_Frag_Enab    BEHAVIOUR
                DEFINED AS      !--Subclause 2E.5 of Annex 2E: Fragmentation of IEEE Std 802.10-
                                1998, Clause 2--! ;
        ;
REGISTERED AS                   { sils attribute(7) assoc_frag_enab(36) };

aBadPDUsICVCount                ATTRIBUTE
        DERIVED FROM            "ISO/IEC 10165-2":counter;
        BEHAVIOUR
            bBadPDUsICVCount    BEHAVIOUR
                DEFINED AS      !This attribute is a counter that increments each time a bad SDE
                                PDU that has a valid security association is received but the ICV is
                                incorrect. It is associated with Bad PDUs ICV Threshold and Bad
                                PDUs ICV Discarded. This value can be read and set by system
                                management action! ;
        ;
REGISTERED AS                   { sils attribute(7) badpdusICVcount(37) } ;

aBadPDUsICVThreshold            ATTRIBUTE
        DERIVED FROM            "ISO/IEC 10165-2":counter-Threshold;
        BEHAVIOUR
            bBadPDUsICVThreshold BEHAVIOUR
                DEFINED AS      !This attribute is associated with Bad PDUs ICV Count and Bad
                                PDUs ICV Discarded. The triggering of this threshold by the
                                associated Bad PDUs ICV Count causes the generation of a Bad
                                PDUs ICV Discarded notification. This value can be read and set by
                                system management action.! ;
        ;
REGISTERED AS                   { sils attribute(7) badpdusICVthreshold(38) } ;

aBadPDUsSAIDCount               ATTRIBUTE
        DERIVED FROM            "ISO/IEC 10165-2":counter;
        BEHAVIOUR
            bBadPDUsSAIDCount   BEHAVIOUR
                DEFINED AS      !This attribute is a counter that increments each time a bad SDE
                                PDU that has a valid security association and correct ICV is received
                                but the PDU cannot be processed. It is associated with Bad PDUs
                                SAID Threshold and Bad PDUs SAID Discarded. This value can be
                                read and set by system management action.! ;
        ;
REGISTERED AS                   { sils attribute(7) badpdusSAIDcount(39) } ;

aBadPDUsSAIDThreshold           ATTRIBUTE
        DERIVED FROM            "ISO/IEC 10165-2":counter-Threshold;
        BEHAVIOUR
            bBadPDUsSAIDThreshold BEHAVIOUR
```

DEFINED AS **!This attribute is associated with Bad PDUs SAID Count and Bad PDUs SAID Discarded. The triggering of this threshold by the associated Bad PDUs SAID Count causes the generation of a Bad PDUs SAID Discarded notification. This value can be read and set by system management action.!** ;

;

REGISTERED AS { **sils attribute(7) badpdusSAIDthreshold(40)** } ;

aAssoc_Label_Set ATTRIBUTE
WITH ATTRIBUTE SYNTAX **IEEE802.10_SDE_ASN1Module.ALabel_Set ;**
MATCHES FOR EQUALITY;
BEHAVIOUR
bAssoc_Label_Set BEHAVIOUR
DEFINED AS **!This attribute contains the registered name for the label set supported by the association.! ;**

;

REGISTERED AS { **sils attribute(7) assoc_label_set(48)** } ;

aAssoc_Label_Value ATTRIBUTE
WITH ATTRIBUTE SYNTAX **IEEE802.10_SDE_ASN1Module.Label_Values ;**
MATCHES FOR EQUALITY;
BEHAVIOUR
bAssoc_Label_Value BEHAVIOUR
DEFINED AS **!This attribute contains the label information value or values that can be used over the association.! ;**

;

REGISTERED AS { **sils attribute(7) assoc_label_value(49)** } ;

aAssoc_Label_Explicit ATTRIBUTE
WITH ATTRIBUTE SYNTAX **IEEE802.10_SDE_ASN1Module.SDE_Boolean ;**
MATCHES FOR EQUALITY;
BEHAVIOUR
bAssoc_Label_Explicit BEHAVIOUR
DEFINED AS **!This attribute indicates whether the negotiated security label is required on every SDE PDU or if the label implied from the SAID. This value can only be false if the association accepts a single label value.! ;**

;

REGISTERED AS { **sils attribute(7) assoc_label_explicit(50)** } ;

2.8.6.6.3 Specification of Security_Association notifications

nBadPDUsICVDiscarded NOTIFICATION
BEHAVIOUR
bBadPDUsICVDiscarded BEHAVIOUR
DEFINED AS **!This notification is sent when the Bad PDUs ICV Count reaches the Bad PDUs ICV Threshold value. The following information may optionally be included in the notification message; the Bad PDUs ICV Count and/or the last incoming PDU with a bad ICV.!** ;

;

MODE CONFIRMED AND NONCONFIRMED;
WITH INFORMATION SYNTAX **Notification-ASN1Module.AttributeValueChangeInfo**
AND ATTRIBUTEIDS ;

REGISTERED AS { **sils notification(10) badpdusICVdiscarded(4)** } ;

nBadPDUsSAIDDiscarded NOTIFICATION
BEHAVIOUR
bBadPDUsSAIDDiscarded BEHAVIOUR
DEFINED AS **!This notification is sent when the Bad PDUs SAID Count reaches the Bad PDUs SAID Threshold value. The following information may optionally be included in the notification message; the Bad PDUs ICV Count and/or the last incoming PDU that cannot be processed.!** ;

```
        ;
        MODE                        CONFIRMED AND NONCONFIRMED;
        WITH INFORMATION SYNTAX     Notification-ASN1Module.AttributeValueChangeInfo
                                    AND ATTRIBUTEIDS ;
REGISTERED AS                       { sils notification(10) badpdusSAIDdiscarded(2) } ;
```

2.8.6.6.4 Specification of Security_Association parameters

```
paCreateError3                      PARAMETER
        CONTEXT                     SPECIFIC-ERROR ;
        WITH SYNTAX                 IEEE802.10_SDE_ASN1Module.CreateError3 ;
        BEHAVIOUR
                bCreateError3       BEHAVIOUR
                        DEFINED AS  !This CreateError information is returned when a managed object
                                    cannot be created. The error information is placed in the
                                    SpecificErrorInfo and is of the form
                                    SpecificErrorInfo ::=       SEQUENCE {
                                        errorid                 OBJECT IDENTIFIER,
                                        objectName              aSA_ID          OPTIONAL,
                                        errorinfo               CreateError3 }
                                    The OBJECT IDENTIFIER will be that which this parameter is
                                    registered under, objectName will optionally contain the naming
                                    attribute, and errorinfo will have the syntax of CreateError3
                                    containing the error reason.! ;
        ;
REGISTERED AS                       { sils parameter(5) createerror3(3) } ;
```

2.8.7 Conformance

In order to allow conformance to be claimed to ISO/IEC 15802-2: 1995, a conformance clause is included in that standard. A product claiming conformance shall implement the following:

a) A defined subset of the ISO/IEC 15802-2: 1995 LAN/MAN Management protocol;
b) The set of managed objects that are defined as mandatory in all IEEE 802 standards to which conformance is claimed within a station (i.e., IEEE 802.1 plus the appropriate MAC and LLC standards including those for IEEE 802.10).

Claims of conformance shall indicate which name bindings are supported by the implementation.

2.8.7.1 Conformance statements

The conformance statement associated with the sublayer management section shall

a) List the mandatory managed objects and operations for the defined managed objects, and any mandatory name bindings.
b) List the attributes, actions, notifications and parameters that are supported.
c) List the name bindings, attribute groups, and packages that are supported.

NOTE—The complete package must be supported to claim conformance.

The PICS for ISO/IEC 15802-2: 1995 is basically the same as those for CMIP and is specified in ISO/IEC 9596-2: 1993.

2.8.7.2 Recommendations

It is recommended that any conformance statements that are provided by a supplier of an implementation of the managed objects specified in this standard provide that information in a format similar to the one specified for MOCS proforma as outlined in ISO/IEC 10165-6: 1997.

MOCS proforma is a statement of the implementor of an (management) agent that contains the managed object or just the managed object instance. That proforma statement claims conformance to that managed object's class definition or set of definitions. For each managed object class claim, there are a set of additional statements about the status and support of that managed object class's associated items that include name bindings, attributes, attribute groups, actions, notifications, and parameters.

The MOCS proforma statements use a set of notations for compliance status and for support responses in statements about the associated items. These notations are as follows:

Status	c	conditional
	m	mandatory
	o	optional
	x	prohibited
	—	not applicable
Support	Ig	the item is ignored
	N	not implemented
	Y	implemented
	—	not applicable

3. Key Management

The purpose of Key Management is to establish the keying material and association attributes needed by security protocols. This information is represented as management objects; the set of all security-related objects is called an SMIB. While Key Management to support the SILS SDE protocol is the primary goal, every attempt will be made to allow this key management function to support security protocols residing at other layers (e.g., ISO/IEC 10736 [B1], Transport Layer Security Protocol). This clause is currently published separately as IEEE Std 802.10c-1998.

4. Bibliography

The following documents are referred to in the text of this standard. They are listed here for information.

[B1] ISO/IEC 10736: 1995, Information technology—Telecommunications and information exchange between systems—Transport layer security protocol.

[B2] ISO/IEC 8802-5: 1998 [ANSI/IEEE Std 802.5, 1998 Edition], Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 5: Token ring access method and physical layer specifications.

[B3] NCSG-TG-005 Version-1, National Computer Security Center, "Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria," July 31, 1987.

Annexes

The annexes in IEEE Std 802.10-1998 are numbered to correspond to each clause of the document. Thus the annexes applicable to Clause 2, Secure Data Exchange (SDE), are numbered 2A, 2B, etc. Annexes for Clause 3, Key Management, can be found in IEEE Std 802.10c-1998.

Annex 2A

(informative)

Service rationale

This annex contains the rationale for the selection of SDE security services.

2A.1 Layer 2 security services for LANs

2A.1.1 Abstract

The ISO Security Architecture, ISO 7498-2: 1989 [2A-2],[16] was developed using Packet Switched Networks (PSNs) and Wide Area Networks (WANs) as architectural models. Since that time, there have been significant changes in networking practices. LANs have introduced a new range of vulnerabilities that are not present in the Data Link Layer of PSNs and WANs. The point-to-point nature of the Data Link Layer (Layer 2) of PSNs and WANs led to the dismissal of the need for extensive security services at Layer 2. Subnetworks and routing were the focus of the need for inclusion of particular security services at the Network and Transport Layers. LANs, however, have introduced subnetworks and routing into the Data Link Layer of many networks. Efforts aimed at providing security services for LANs have found the current Link Layer security service profile in ISO 7498-2: 1989 to be deficient. It is necessary to expand this service profile to protect LANs, even in the presence of security services at higher layers in the protocol stack.

2A.1.2 Introduction

In the spring of 1988, preliminary meetings were held to determine interest in security standards for LANs. This led to the formation of the IEEE 802.10 LAN Security Working Group, which is sponsored by the IEEE LAN/MAN Standards Committee. The working group's charter is the development of standards for Interoperable LAN Security (SILS).

The IEEE 802.10 LAN Security Working Group concentrated its efforts on developing an SDE protocol to be inserted between the MAC and the LLC sublayers of the Link Layer in the ISO OSI Basic Reference Model. The working group also completed development of a Key Management protocol.

In the course of the development of the SDE protocol, the IEEE 802.10 LAN Security Working Group drew up a list of necessary security services. In large part, this list was based on the attributes of emerging LAN security devices. An analysis of the attributes of LANs that make these security services necessary is presented in this annex. The pertinent attributes are identified and the associated security threats are detailed. Then, the security services necessary to counter those threats are indicated, examples of the benefits of application of those security services are given, and mechanisms for providing the services are discussed.

[16]The numbers in brackets correspond to those of the bibliography in 2A.3.

2A.1.3 Security services under the ISO security architecture

Five basic security services are identified in ISO 7498-2: 1989 [2A-2]: access control, authentication, data confidentiality, data integrity, and non-repudiation. These services provide assurance against the security threats of unauthorized resource use, masquerade, unauthorized data disclosure, unauthorized data modification, and repudiation, respectively. The ISO standard also defines the layers within the ISO OSI Basic Reference Model where it is appropriate to apply these services. A brief justification for the indicated services placement is given in annex B of ISO 7498-2: 1989.

In ISO 7498-2: 1989, data confidentiality is the only security service indicated for the Data Link Layer of the ISO OSI Basic Reference Model. Other security services were “not considered useful” at this layer. Arguments for the inclusion of the services of authentication, access control, and data integrity at the Data Link Layer as well are provided in this annex. It is important to note that the arguments presented in this annex are based on changes in networking practices since ISO 7498-2: 1989 was completed, not on deficiencies intrinsic to ISO 7498-2: 1989 as it was originally conceived. LAN standards have only recently begun to appear in the ISO standards arena (e.g., ISO/IEC 8802-2: 1998 [2A-4]). Because of changes in LAN technology, the risks to LANs have become more critical than first anticipated. High-speed, long distance LANs (e.g., the Fibre Distributed Data Interface, or, FDDI), filtering LAN bridges, and LAN server facilities have increased the range of resources that are vulnerable to abuse. Ring topology networks not only make every PDU (e.g., packet, frame) available to every station on the LAN, but also require every station on the LAN to receive and then forward every PDU, in order for the LAN to operate properly. These issues raised the concerns that led to this set of arguments. The differences between the security service profile defined in ISO 7498-2: 1989 and the profile proposed for LANs is illustrated in Figure 2A.1.

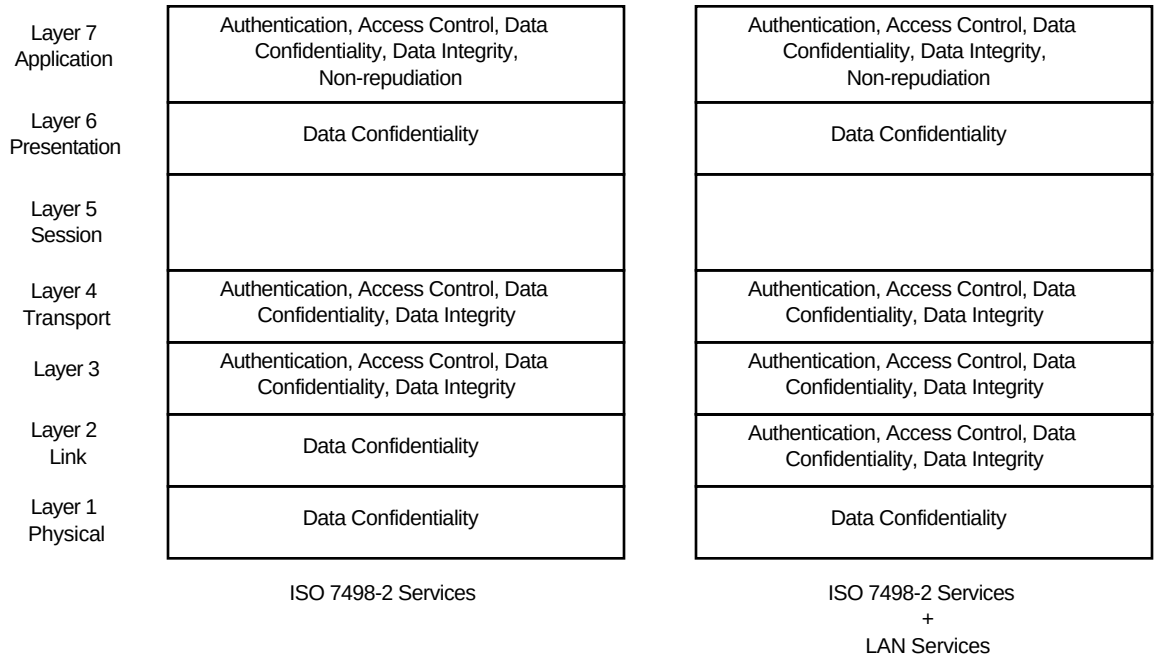


Figure 2A.1—Comparisons of ISO 7298-2 security service profile and proposed LAN profile

In a specific implementation, a security service can be implemented in any layer at which it is indicated. A service may appear in one layer, more than one layer, or not at all. ISO 7498-2: 1989 indicates only where the service can appear, not where the service is required to appear. The security requirements for a particular implementation will determine where the service will be provided. In practice, it is desirable to protect infor-

mation at both the highest possible point in the protocol stack (i.e., the application layer) and any layers at which subnetworks and routing are implemented.

The ISO Security Architecture was developed using PSNs and WANs as an architectural model. It was assumed that these networks would have a tightly controlled Data Link Layer configuration. In this model, the HDLC Frame was used to represent the Data Link Layer PDU.[17] It was also assumed that the Data Link Layer of LANs had the same attributes as the Data Link Layer of the model. In fact, while LANs are similar to PSNs and WANs at the Data Link Layer, they also exhibit some of the attributes of the Network Layer of PSNs and WANs. For example, the Data Link Layer of LANs exhibits subnetwork and routing functions similar to those of the Network Layer. These functions are cited as justification for the Network Layer security service profile, which is the same as the security service profile proposed in this annex for the Link Layer. These similarities and differences are indicated in the following subclauses as the security-pertinent attributes of LANs are explored.

2A.1.4 LAN characteristics that necessitate security services at the Data Link Layer

There are certain characteristics of LANs that necessitate security services at the Data Link Layer—the manner in which data is transmitted, the manner in which data is received, the nature of LAN address space, and geographic dispersion of LANs. The security threats associated with these characteristics will be identified. Then the security services required to address these threats will be indicated and how they are applied to LAN data will be shown. Finally, mechanisms for providing these services will be discussed.

2A.1.4.1 Data transmission on a LAN

The manner in which data is transmitted on LANs is one of the attributes that necessitates additional security services at Layer 2. In a LAN's Data Link Layer, data is transmitted on media that is shared by every attached system. Effectively, every PDU is transmitted to every other station on the LAN, and the source of a given transmission is difficult to authenticate.

The nature of data transmission at the Data Link Layer on a LAN presents two security threats. First, any station attached to a LAN can transmit to any other station attached to a LAN. There are no implicit controls at Layer 2 on access to a resource attached to a LAN. Second, since it is difficult to identify the source of a given data transmission, one station can claim to be another station. Any station, or set of stations, can be imitated from a single tap into the LAN. The source of a given PDU is difficult to authenticate. These threats to the security of a LAN are known formally as *unauthorized resource use* and *masquerade*.

2A.1.4.2 Data reception on a LAN

The manner in which data reception is received on LANs is another attribute that necessitates additional security services at Layer 2. Since data transmission at a LAN's Data Link Layer is over commonly accessible media, every PDU is available to all attached stations. A PDU could traverse any station on its way to its destination. This means that while it may be addressed to a specific entity, every PDU is effectively received by every other station attached to the LAN.

The nature of data reception on a LAN presents two security threats, since any PDU could be intercepted by any attached station. First, a station could receive data for which it is not authorized. Second, and worse yet, a station could change the data in a PDU before it is received at its intended destination. On LANs, data for any station, or set of stations, can be received from a single station on the LAN. This is especially significant in LANs employing a ring topology, where every attached system must receive and retransmit every PDU in

[17]While this simplified model may not represent all possible implementation of PSNs and WANs, it does represent the mapping of many PSNs and WANs onto the ISO OSI Basic Reference Model. X.25 Packet Level Interface functions are attributed to the Network Layer. The assumption of tightly controlled configurations, in particular, may seem restrictive, but reflects standard practices in the implementation of secure networks.

order for the LAN to function properly. These threats to the security of a LAN are known formally as *unauthorized disclosure* and *data modification*.

2A.1.4.3 LAN address space

Assignments within the address space of a LAN are also pertinent to security. Each station interface is permanently assigned a specific address. Since any station interface can be attached to any other station interface through a common medium at Layer 2, LAN addresses must be unique at Layer 2. This means that a station cannot determine, by observation, whether the source address of a PDU is valid or not. There is no hierarchical address assignment in LANs, so any possible link address could be valid on any LAN.

As with data transmission, the nature of address assignment at the Data Link Layer on a LAN presents two security threats. First, any station attached to a LAN can transmit to any station attached to the LAN. There are no implicit controls at Layer 2 on access to a station attached to a LAN. Second, since it is difficult to identify the source of a given data transmission, one station can claim to be another station. Any station, or set of stations, can be imitated from a single tap into the LAN. The source of a given PDU is difficult to authenticate. These threats of a LAN are known formally as *unauthorized resource use* and *masquerade*.

2A.1.4.4 Geographic dispersion of LANs

LANs span vast geographic areas, rendering them vulnerable to eavesdropping or wiretap. This also renders them vulnerable to the threats of unauthorized disclosure and data modification. As indicated previously, there is a significant scope of information and access available on a LAN at Layer 2; any station, or set of stations, can be imitated from a single tap into the LAN.

Wiretapping on a LAN presents two security threats. First, a station can receive data for which it is not authorized. Second, and worse yet, a station can change the data in a PDU before it is received at its intended destination. Again, on LANs, data for any station, or set of stations, can be received from a single tap into the LAN. This is especially significant in LANs employing a ring topology, where every attached system must receive every PDU for the LAN to function properly. These threats to the security of a LAN are known formally as *unauthorized disclosure* and *data modification*.

2A.1.5 Security services

In this subclause, the type of architecture that requires the indicated security services will be described; the security services will be described in detail; and the formal definition of each service from the ISO Security Architecture will be reviewed. Also, the application of each service to PDUs at the Data Link Layer on a LAN will be examined, making note of the portions of a PDU that are protected by the service.

In Figure 2A.2, a LAN has been subdivided into several local segments, or subnetworks, that are interconnected through a backbone network. The subnetworks are effected through the use of bridges, which pass a PDU between a subnetwork and the backbone network only when that PDU is directed from a station on one side of the bridge to a station on the other side of the bridge. Some of the subnetworks have been designated as protected subnetworks, i.e., subnetworks that are safe from attachment of unauthorized stations, as opposed to unprotected networks.

Rogue stations are those that participate in unauthorized activities, whether or not the station is authorized to be attached to the LAN. These rogue stations exploit the risks that have been identified, necessitating the indicated security services. Precautions are necessary to provide protection from these stations wiretapping into the backbone LAN. LAN security services are also necessary to prevent abuse by systems that are authorized to be connected to the LAN, but are being used in an unauthorized fashion. Without the proper security services, even protected subnetworks are susceptible to abuse.

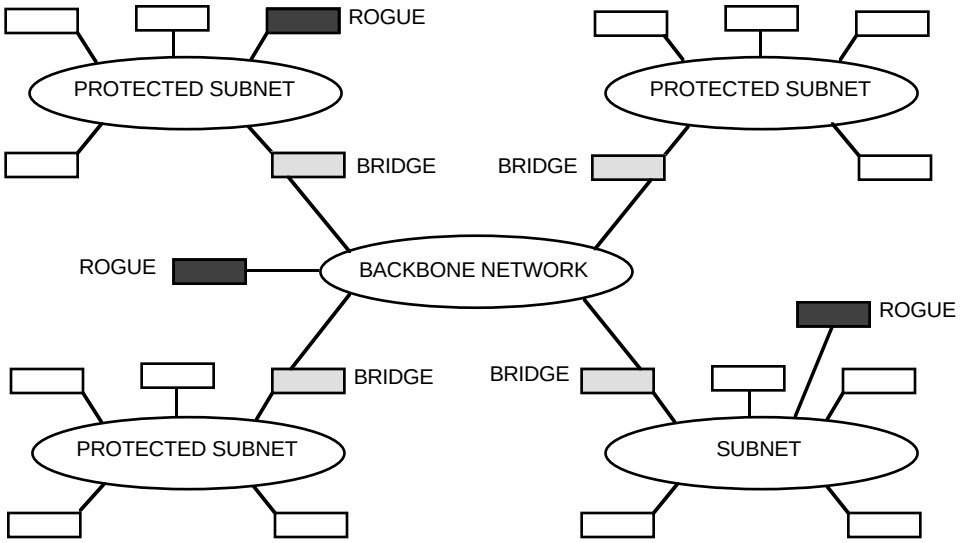


Figure 2A.2—Subnetworks

Ultimately, protection of application data can be provided at the application layer. In practice, however, it is desirable to protect information at both the highest possible point in the protocol stack (i.e., the application layer) and any layers at which subnetworks and routing are implemented. This is true for several reasons.

First, security services provided at any layer of a protocol stack protect only the SDU, i.e., the data portion, of that layer's PDU. If data integrity is provided at an upper layer, the header information from that layer and all lower layers is left unprotected. One example of data in a Layer 2 information PDU that is unprotected, even in the presence of higher layer security services, is the security option specified for ISO CLNP, which is included in the US Government Open Systems Interconnection Profile (US GOSIP) [2A-9]. Since this data is contained within the Network Layer header, it cannot be protected by security services provided above the Data Link Layer.

Second, PDUs that originate and terminate within Layer 2 are also unprotected in the presence of security services at upper layers. Examples of this type of PDU are the TEST and XID PDUs in ISO/IEC 8802-2: 1998 [2A-4] LLC, which is also part of the US GOSIP. Network management uses these PDUs, creating a need for protection for this type of PDU as well as information PDUs. ISO 7498-2: 1989 [2A-2] considers only information PDUs. It does not address administrative functions and artifacts of protocols. Connectionless data integrity at the Link Layer will provide protection for this type of PDU, as well as information outside the boundary of protection of higher layer security services.

Third, security services provided at the Link Layer provide uniform, common protection for all applications from risks that are intrinsic to LANs and the increased connectivity they provide. Security services provided at another layer can neither take advantage of the attributes of a LAN nor be affected by the deficiencies of a LAN.

Finally, implementations of security at upper layers are developing too slowly to address some users' needs. Emerging LAN security devices can address these needs until upper layer security is available.

2A.1.5.1 Connectionless data integrity

Connectionless data integrity is defined in ISO 7498-2: 1989 [2A-2] as "the property that the data in a single connectionless PDU has not been altered or destroyed in an unauthorized manner." As the definition indi-

cates, this service inhibits undetected modification of the protected data. This assures the receiving station that the SDU portion of a PDU has not been tampered with since it was transmitted. Given the nature of data transmission and reception at the Link Layer of LANs and the susceptibility of LANs to wiretap, this service is badly needed to protect data on LANs. This service is important not only in its own right, but as a necessary supportive service for authentication services.

The application of this service to information PDUs is illustrated in Figure 2A.3. As previously indicated, security services provided at any layer of a protocol stack protect only the SDU portion of that layer's PDU. In implementations where integrity is provided at a higher layer, connectionless data integrity at Layer 2 protects the headers of the layer above the MAC sublayer up to and including the higher layer at which integrity is provided. The security option specified in the US GOSIP for ISO CLNP [2A-9] is one example of critical data protected in this case. Since this data is contained within the Network Layer Header, it cannot be protected by security services provided above the Link Layer. Modification of the data contained in the security option, combined with the modification of the CLNP header checksum, could result in delivery of a PDU to a station not authorized to process that data. Implementations where connectionless data integrity is provided at the MAC sublayer are protected from undetected modification. When implemented at the Data Link Layer, this service also provides protection for logical subnetwork addressing for communities of interest on a common secure backbone LAN.

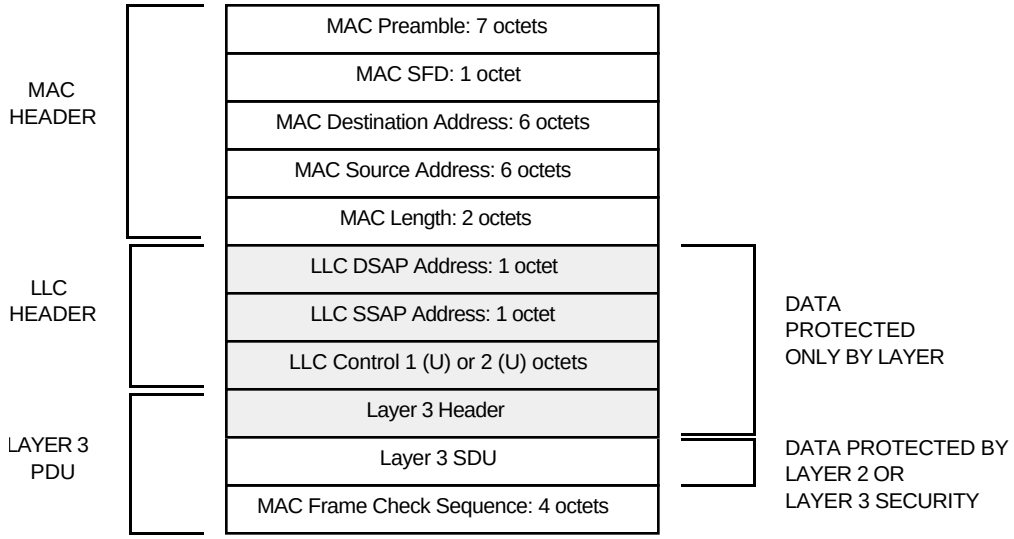


Figure 2A.3—Connectionless data integrity service applied to information PDUs

Connectionless data integrity is also necessary at the Data Link Layer to inhibit data modification of the data field of the TEST PDU. The application of connectionless data integrity to this type of PDU is illustrated in Figure 2A.4. If the data in a TEST PDU is altered by a third party, either during the request or reply phase, it might result in a bad quality path being marked as good. Distortion of TEST data could also cause a good quality path to be marked as bad, but this is indistinguishable from a failure in the media itself and is, in fact, an indication that there is something wrong with the communication path anyway. This service also protects the integrity of the LLC header fields, preventing misdelivery of the TEST PDU or modification of the Control field, which identifies the PDU as a TEST PDU. Finally, integrity is also necessary as a supportive service for authentication of this type of PDU, since assurance of authenticity of the source address without assurance of the integrity of the source address is of little value.

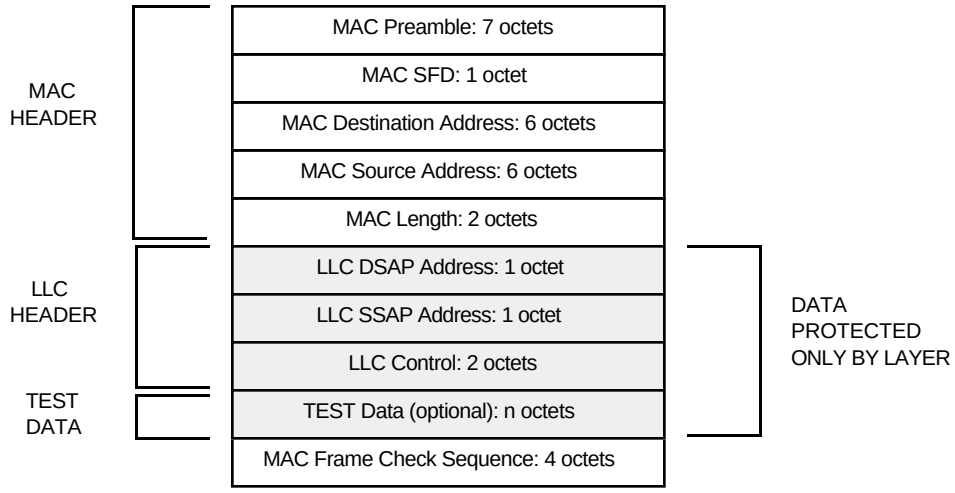


Figure 2A.4—Connectionless data integrity service at the Data Link Layer

2A.1.5.2 Data origin authentication

Data origin authentication inhibits one station from masquerading as another to abuse resources attached to a LAN (i.e., unauthorized resource use). This service assures a receiving station that the SDU portion of a PDU came from the station indicated by the Data Link Layer source address in the PDU header. Data integrity is necessary as a supportive service for data origin authentication, since assurance of authenticity of the source address without assurance of the integrity of the source address is of little value. This service protects resources (e.g., file servers) attached to LANs from one station masquerading as another, whether or not the station is authorized to be connected to the LAN. At Layer 2, this service provides protection for logical subnet addressing for communities of interest on a common secure backbone. Given the nature of data transmission and reception at the Link Layer of LANs and the susceptibility of LANs to wiretap, this service is necessary to protect resources on LANs.

The application of this service to information PDUs at the Data Link Layer is illustrated in Figure 2A.3. When authentication is provided at an upper layer, the header data from that upper layer and all lower layers is left unprotected. Again, an example of data in a Layer 2 information PDU that is unprotected even in the presence of higher layer security services, is the security option specified in the US GOSIP for ISO CLNP [2A-9]. Since this data is contained within the Network Layer Header, it cannot be protected by security services provided above the Link Layer. If an unauthorized station masqueraded as an authorized station and replayed the data contained in the security option from a valid PDU, it could result in delivery of data to a station not authorized to process that data. In implementations where data origin authentication is provided at the Link Layer rather than at a higher layer, application data and all of the headers of the protocol layers above the MAC sublayer are protected. When implemented at the Link Layer, this service also provides protection for logical subnet addressing for communities of interest on a common secure backbone LAN.

Data origin authentication is also necessary at Layer 2 to inhibit modification of the source address field of a TEST PDU. The application of data origin authentication to this type of PDU is illustrated in Figure 2A.4. If the source address in a TEST PDU is altered, either during the request or reply phases, it might result in a bad quality path being marked as good. Misrepresentation of the source address in a TEST PDU could also cause a good quality path to be marked as bad, but this is indistinguishable from a failure in the media itself and, in fact, is an indication that there is something wrong with the communications path anyway. Together with the supportive service of integrity, data origin authentication provided necessary protection for this type of PDU, since assurance of authenticity of the source address without assurance of the integrity of the source address is of little value.

2A.1.5.3 Access control

Access control inhibits unauthorized use of resources. This service is sometimes thought of as a way to inhibit unauthorized disclosure. But, in fact, data confidentiality is used to protect data from unauthorized disclosure. Access control provides assurance that access to a resource is granted only to authorized stations for authorized purposes. Access control can be applied at either the course of a data transmission or at the destination. However, when access control is applied at a PDU's destination, the data has effectively been transmitted to all stations on a LAN before this service is applied. If nothing else, this leaves stations open to unauthorized depletion of network bandwidth and receiver processing resources. Also, due to the manner in which every PDU is effectively transmitted to every station on a LAN and the susceptibility of LANs to wiretap, access control applied at the destination cannot prevent transmissions to stations not authorized to be connected to a LAN. At the Data Link Layer of a LAN, access control, when applied at the source of data transmission, can inhibit communications between stations not authorized to communicate with one another, including a station authorized to be connected to the LAN and a station not authorized to be connected to the LAN.

The application of this service to information PDUs is illustrated in Figure 2A.3. In an implementation where authentication is provided at a higher layer, access control at Layer 2 provides protection from abuse of resources that operate upon data contained in the headers of the higher layer at which the service is provided and all other layers above Layer 2. For example, in a network where access control is provided as a Layer 3 end-to-end service over ISO CLNP, PDUs generated on one LAN could be sent to a remote LAN with particular Quality of Service (QOC) option parameters requested and the Record Route option invoked. This would provide information on the topology of a set of interconnected subnetworks that could develop more complete information from Error Report PDUs, without the participation of a second rogue unit. This information could be used to exploit weaknesses in the network, such as identifying operational characteristics of particular routes (e.g., relative levels of congestion, transit delay, or residual error probability). While access control at Layer 2 cannot limit this type of abuse between stations authorized to communicate with one another, it can inhibit this type of communication between stations not authorized to communicate with one another. In implementations where access control is provided at the Link Layer rather that at a higher layer, this service provides protection from abuse of application data and data in the headers of the protocol layers above Layer 2. For example, this service can limit access to a particular file server to only those stations that required that access. It can also prohibit access to a gateway from unauthorized stations.

At the Link Layer of a LAN, this service can prevent use of the TEST PDU from the LLC sublayer to create an unauthorized communications association. The application of access control to this type of PDU is illustrated in Figure 2A.4. Since the data to be used for a TEST PDU is not defined, the entire data field of this PDU could be filled with any data. By transmitting unnecessary TEST PDUs, cooperating stations could transfer any data. While access control will not limit this type of abuse between stations authorized to communicate with one another (e.g., a station authorized to be connected to the LAN and a station not authorized to be connected to the LAN).

2A.1.5.4 Data confidentiality

Data confidentiality inhibits unauthorized disclosure of the protected data. This assures the sending station that the protected portion of a PDU will be available only to the intended recipient. Given the nature of the Link Layer of LANs and the susceptibility of LANs to wiretap, this service is necessary to protect data on LANs. This service is already indicated as appropriate for Layer 2 in ISO 7498-2: 1989 [2A-2].

2A.1.6 Mechanisms for provision of security systems

Concerns that are raised when one suggests expanding the Layer 2 security profile include how the additional security services can be provided, and what impact this will have on the complexity and performance of the LAN interface to a station. Data confidentiality is most commonly provided via encryption, also referred to as encipherment. In fact, data confidentiality through encryption is what most people associate

with network security. While there are other mechanisms for providing data confidentiality, encryption is one of the simplest and most reliable. Fortunately, the mechanism most commonly used to provide data confidentiality, i.e., encryption, can be used to provide all of the indicated security services. In fact, the additional services can be provided with almost no impact to the performance or the complexity of the LAN interface.

Connectionless data integrity is almost an automatic side effect of data confidentiality via encryption. Most cryptographic algorithms produce a checksum or some other mathematical residue that can only be reproduced with the correct combination of cryptographic algorithm, key material, and data. For systems handling classified data, a cryptographic checksum calculated over the data, using an algorithm and key different from those used for the data confidentiality service, might be required.

Data origin authentication can easily be provided by including a copy of the source address within the encrypted data field, either as a prefix or a suffix to the Layer 2 SDU.[18] As with connectionless data integrity, in systems handling classified data, a cryptographic checksum calculated over the data using an algorithm and key different from those used for the data confidentiality service might be required. Again, however, this is unnecessary for unclassified data.

Access control can be effected implicitly through the management and application of cryptographic association, i.e., keying relationships. If all PDUs are encrypted, only those stations with cryptographic mechanisms and knowledge of the correct keying relationships can exchange information. A station without these facilities will be unable to access any of the protected resources.

With the exception of data origin authentication, all of the additional services can be provided as by-products of encryption when used to provide data confidentiality—and data origin authentication can be included so easily that it is hardly worth noting as an exception. Using the single mechanism of encryption, all of the indicated services can be provided with a minimum of impact to the complexity and performance to the LAN interface of an attached station.

2A.2 Summary

The pertinent attributes of LANs that have been identified, the vulnerabilities that those attributes present, the security threat associated with those vulnerabilities, and the security services required to inhibit exploitation of those risks are summarized in Table 2A.1. In each case, the Link Layer of LANs has been shown to have qualities more like the Network Layer of WANs than those of the Link Layer of WANs. Given these arguments, it makes sense to provide the same range of security services for the Link Layer of LANs as for the Network Layer of WANs.

[18]Data origin authentication is assured only to the granularity of the cryptographic key. A key that is unique to the source and destination address pair provides assurance of the individual source host identity; a key shared by a group only provides assurance that the source of the PDU is a member of the group.

Table 2A.2—Summary of security services required for LANs

LAN attribute	Vulnerability	Security threat	Services indicated
Data transmission	Any station can transmit to any other station, using any address	Masquerade, unauthorized resource use	Data origin authentication
Data reception	Any station can access any transmission	Data modification, unauthorized disclosure	Connectionless data integrity, data confidentiality
Address space	No implicit controls through address management	Masquerade, unauthorized resource use	Data origin authentication, access control
Geographic dispersion	Eavesdropping, wiretapping	Data modification, unauthorized disclosure	Connectionless data integrity, data confidentiality

2A.3 Bibliography

[2A-1] IEEE Std 802.3, 1998 Edition, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.

[2A-2] ISO 7498-2: 1989, Information processing systems—Open Systems Interconnection—Basic Reference Model—Part 2: Security Architecture.

[2A-3] ISO 8473: 1988, Information processing systems—Data Communications—Protocol for providing the connectionless-mode network service.

[2A-4] ISO/IEC 8802-2: 1998 [ANSI/IEEE Std 802.2, 1998 Edition], Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 2: Logical link control.

[2A-5] ISO/IEC 8802-4: 1990 [ANSI/IEEE Std 802.4-1990], Information processing systems—Local area networks—Token-passing bus access method and physical layer specifications.

[2A-6] ISO/IEC 8802-5: 1998 [ANSI/IEEE Std 802.5, 1998 Edition], Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 5: Token ring access method and physical layer specifications.

[2A-7] Berson, Thomas A. and Beth, Thomas, eds., *Lecture Notes in Computer Science: Local Area Network Security.* Springer-Verlag, April 1989.

[2A-8] Tanenbaum, Andrew S., *Computer Networks.* New York: Prentice-Hall, 1981.

[2A-9] United States Government Open Systems Interconnection Profile (US GOSIP), Version 2; Federal Register; National Institute of Standards and Technology; July 1989.

Annex 2B

(informative)

Example

An example of the use of the SDE protocol is presented in this annex. This example will include two parties (A and B) and will examine the contents of SMIB and the PDU construction. Since it is an example, it contains some implications for local processing that are not part of the standard. It uses the following objects that are not defined in SDE; however, these objects may subsequently be defined in 2.8, SDE sublayer management, of the standard.

Station_Max_SDU_Size: The maximum size SDU that the MAC sublayer can support. In this example, it is set to 1518 octets for IEEE Std 802.3.

SAP_Worst_Case Expansion: This is the maximum number of octets that can be added by SDE for SDUs originating at the indicated SDE SAP. The calculation for this object is described later in this document.

MAX_SDE_SDU_Size: This is calculated by subtracting SAP_Worst_Case_Expansion from the Station_MAX_SDU_Size. It is the maximum size SDU that SDE will accept.

2B.1 Algorithm registry

The SDE protocol expects the attributes of any confidentiality algorithm to be registered. This subclause contains excerpts from the registry.

Algorithm ID: 1
Name: DES CBC mode (ANSI X3.106-1983 [2B-1][19])
IV Length: 64 bits
Key Length: 56 bits + 8 bits parity
Class: Symmetric
Service: Confidentiality
Additional Fields and Placement: none

Algorithm ID: 2
Name: ANSI X9.9-1986 [2B-2] (Revised) Binary Option, modified
ICV Length: 32 bits
Key Length: 56 bits + 8 bits parity
Class: Symmetric
Service: Integrity
Modifications:
Date of Message Origin: Not used.
Message Identifier: Not used.

2B.2 Key management

In this example, certain parameters are negotiated between the two key management applications to set up parameters for the communication. The effect on the SMIB will be shown in Clause 3 of this document.

[19]The numbers in brackets correspond to those of the bibliography in 2B.7.

Note that there are some parameters that are set by system management (e.g., Addresses, Remote_SDE). The SAID and the MDF are unique among the negotiated attributes in that each is a unidirectional attribute of the SAID and are simply accepted as opposed to negotiated.

2B.2.1 Party A's proposed options

This subclause contains the proposed options that Party A sends Party B. Party B will select a subset of the provided options. In some cases, Party A specifies an alternate option. Binary fields are represented in hexadecimal.

A's SAID=00000034

Assoc_MDF=TRUE,

MDF= 558977883344

Confid=TRUE, ALT=none

Integ=FALSE, ALT=TRUE

Padding_pres=TRUE, ALT=none

ID_pres=TRUE, ALT=FALSE

Confid_Alg_ID=1, ALT=none

Integ_Alg_ID=None, ALT=2

Station_ID=8ABCDE3456780000

Assoc_Label_Set={ joint-iso-ccitt(2) country(16) us(840) gov(101) csor(3) sec-labels(1) lyr-2-exa(1) }

Assoc_Label_Value={ security-level = confidential(A0), bit-map = 00 }

Assoc_Label_Explicit=TRUE

2B.2.2 Party B's selected options

Party B chooses the following set from the options provided by Party A:

B's SAID=000000A5

Assoc_MDF=FALSE

Confid=TRUE

Integ=TRUE

Padding_pres=TRUE

ID_pres=FALSE

Confid_Alg_ID=1

Integ_Alg_ID=2

Assoc_Label_Set={ joint-iso-ccitt(2) country(16) us(840) gov(101) csor(3) sec-labels(1) lyr-2-exa(1) }

Assoc_Label_Value={ security-level = confidential(A0), bit-map = 00 }

Assoc_Label_Explicit=FALSE

Note that the cryptographic algorithm required padding, so there was no option. The Assoc_MDF is forced to FALSE although that option is stated by Party A. Party A must be able to support FALSE since it is an MER. Also, since ID_pres is selected to be FALSE, no Station ID is supplied.

2B.3 Party A's SMIB

This subclause describes the relevant entries in Party A's SMIB after the key management negotiation.

2B.3.1 Station parameters

Station_Clear_Hdr=TRUE
Station_MDF=TRUE
Station_Max_MAC_SDU_Size=1518

2B.3.2 SAP parameters

SAP_Worst_Case_Expansion= 41 = 3 (SDE Designator) + 4 (SAID) + 6 (MDF) + 8 (IV) + 8 (Station ID) + 8 (Pad) + 4 (ICV)

Calculated Max_SDE_SDU_Size=1477

2B.3.3 Security association parameters

This subclause contains the relevant parameters in Party A's SMIB after the key management negotiation.

Local_SAID=34
Remote_SAID=A5
Assoc_MDF=FALSE
Confid=TRUE
Integ=TRUE
Padding_pres=TRUE
ID_pres=FALSE
Confid_Alg_ID=1: with key of "763b9d52290886e9"
Integ_Alg_ID=2: with key of "6846c72fab7501a4"
SDE_SAP= reference to User Stack (set by key management)
Remote_SDE=TRUE (set by key management)
Outgoing/Incoming Addresses (set by system management)
Assoc_Label_Set={ joint-iso-ccitt(2) country(16) us(840) gov(101) csor(3) sec-labels(1) lyr-2-exa(1) }
Assoc_Label_Value={ security-level = confidential(A0), bit-map = 00 }
Assoc_Label_Explicit=FALSE

2B.4 Party B's SMIB

This subclause describes the relevant entries in Party B's SMIB after the key management negotiation.

2B.4.1 Station parameters

Station_Clear_Hdr=TRUE
Station_User_Def=TRUE
Station_Max_MAC_SDU_Size=1518

2B.4.2 SAP parameters

SAP_Worst_Case_Expansion= 27 = 3 (SDE Designator) + 4 (SAID) + 8 (IV) + 8 (Pad) + 4 (ICV)

Calculated Max_SDE_SDU_Size=1491

2B.4.3 Security association parameters

This subclause contains the relevant parameters in Party B's SMIB after the key management negotiation.

Local_SAID=A5
Remote_SAID=34
Assoc_MDF=FALSE
Confid=TRUE
Integ=TRUE
Padding_pres=TRUE
ID_pres=FALSE
Confid_Alg_ID=1: with key of "763b9d52290886e9"
Integ_Alg_ID=2: with key of "6846c72fab7501a4"
SDE_SAP= reference to User Stack (set by key management)
Remote_SDE=TRUE (set by key management)
Outgoing/Incoming Addresses (set by system management)
Assoc_Label_Set={ joint-iso-ccitt(2) country(16) us(840) gov(101) csor(3) sec-labels(1) lyr-2-exa(1) }
Assoc_Label_Value={ security-level = confidential(A0), bit-map = 00 }
Assoc_Label_Explicit=FALSE

2B.5 Transmission processing (from Party A)

Assume an SDE_UNITDATA.request with data of length 1005 octets.

2B.5.1 Obtaining the attributes

This subclause and the following subclauses in this annex correspond to 2.6.4.2–2.6.4.8 in the standard. The security association is identified using the SAP and the source and destination outgoing addresses.

2B.5.2 Transmission to non-SDE

Remote_SDE=TRUE, so this does not apply.

2B.5.3 Oversize SDU

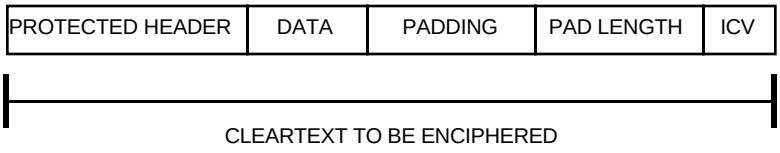
This step is not in the SDE protocol. It is an additional check by the implementation using the objects mentioned in the introduction to this annex. No fragmentation is needed because 1005 is less than SAP_Max_SDE_SDU_size (1477).

2B.5.4 Forming the Protected Header

ID_pres=FALSE and Assoc_Label_Explicit=FALSE, therefore no information is required in the Protected Header. No Protected Header will be formed for this example.

2B.5.5 Pad

The fields that are enciphered in the SDE protocol are illustrated below.



The following is the calculation for the value of the Pad Length:

Pad Length	=	8 –	CBC block size
	(	(0	Protected Header
		+ 1005	size of SDE SDU
		+ 1	Pad Length
		+ 4	ICV
		) MOD 8)	CBC block size

= 8 – (1010 MOD 8) = 8 – 2 = 6

The value in the Pad Length field should be 6.

2B.5.6 Calculation of the ICV

A four-octet ICV is added as specified in ANSI X9.9-1986 [2B-2].

2B.5.7 Encipher the PDU

The PDU is enciphered using CBC, which adds an 8-octet IV.

2B.5.8 Clear Header

The Clear Header is prepended with the Remote_SAID placed in the SAID field.

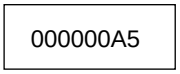
2B.5.9 MAC request

The following appears in the Data field of the MAC request (binary values represented in hexadecimal with leftmost bit most significant):

LSAP	LSAP	UI	SAID	IV	ENCIPHERED DATA
1	1	1	4	8	1016

The UI field contains:

The SAID contains:



The IV is 8 octets of random data.

Before encryption (and, it is hoped, after decryption), the enciphered data contains the following:

LLC PDU	PADDING	PAD LENGTH	ICV
1005	6	1	4

The LLC PDU and the Padding can contain any values. The Pad Length field contains “06”, and the ICV is calculated based on the contents of the preceding three fields.

2B.6 Reception processing (at Party B)

The following steps correspond to the procedure described in 2.6.5.1.1–2.6.5.6 of the standard.

2B.6.1 Requirements for reception

The contents of Party B’s SMIB are contained in 2B.4 of this annex. It is assumed that the values for the bootstrap SAIDs also exist.

Since Station_Clear_Hdr=TRUE, 2.6.5.1.1 is applicable. The first three octets of the received PDU correspond to the SDE_Designator, so the next four octets are used as the SAID. The SAID octets indicate “A5”. In Party B’s SMIB, this indexes into the security association due to the presence of “A5” in the Local_SAID object.

The addresses in the MAC indication are checked against those set by system management in the SMIB. Since they check out as okay in this example, the Clear Header is removed.

2B.6.2 Decipherment of the PDU

Since Confid=TRUE, the PDU is decrypted using the algorithm specified by the Confid_Alg_ID, which is CBC. CBC uses the supplied eight-octet IV, which is also removed prior to further processing. (The decryption key is part of the complex object pointed to by the Confid_Alg_ID.)

2B.6.3 ICV checking

Since Integ=TRUE, the ICV is confirmed using the algorithm specified in Integ_Alg_ID (ANSI X9.9-1986 [2B-2]). (The key is part of the complex object pointed to by the Integ_Alg_ID.) The ICV field is then removed.

2B.6.4 Pad

The last octet in the PDU after the ICV is checked corresponds to the Pad Length. The number of octets in this field plus one (seven total for our example) is removed from the end of the PDU. This leaves the cleartext, integrity-checked Data field (LLC PDU).

2B.6.5 Station ID

ID_pres=FALSE, therefore this subclause is not applicable.

2B.6.6 Security label

Assoc_Label_Explicit=FALSE, therefore this subclause is not applicable.

2B.6.7 SDE_UNITDATA

The LLC PDU is placed in the data parameter of the indication. All other parameters are transferred unaltered to the LLC.

2B.7 Bibliography

[2B-1] ANSI X3.106-1983 (R1996), Modes of Operation for the Data Encryption Algorithm.

[2B-2] ANSI X9.9-1986, Financial Institution Message Authentication (Wholesale).

Annex 2C

(informative)

Objectives of SDE

Before the SDE Protocol was defined, the IEEE 802.10 LAN Security Working Group drew up a list of objectives that they wanted the protocol specification to meet. These objectives were discussed and refined over the course of several meetings. The objectives have been used to evaluate and develop the SDE proposals that were submitted to the working group. These objectives are present in the standard as the requirements for transparency.

a) Make the data exchange protocol independent of the encryption and integrity check algorithms.
b) Allow SILS protected broadcast and multicast.
c) Choose security mechanisms that allow exportability.
d) Allow coexistence of protected and unprotected traffic.
e) Do not rely on layers above the IEEE 802 architecture to provide SILS security services.
f) Support security service and mechanism (as defined in ISO 7498-2: 1989) management by specifying appropriate objects, etc.
g) Maintain the MAC/LLC interface.
h) Allow encipherment in transparent and nontransparent implementations.
i) Allow the support of multiple MAC addresses behind a MAC bridge entity that implements the SILS SDE.

Annex 2D

(informative)

Rationale for placement

2D.1 Introduction

IEEE 802 describes a class of LANs and MANs represented by Figure 2D.1. The placement of security within this architecture can logically occur between the MAC and the LLC Layer, above the LLC Layer, or integrated into either the LLC or MAC sublayer.

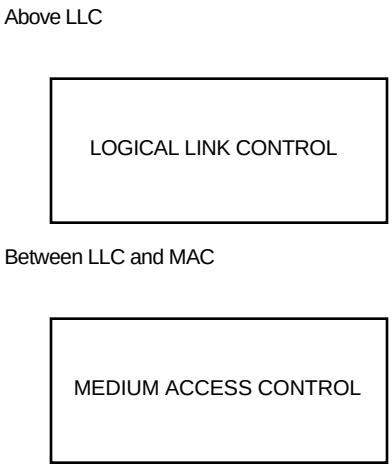


Figure 2D.1—Choices for placement

The attributes of each of these placements are discussed in this annex. It is recommended that the placement directly above the MAC sublayer as a sublayer or as an LLC entity is the most suitable candidate.

2D.2 Integrated into MAC

The MAC sublayer has been developed by several different standards working groups: Carrier Sense Multiple Access/Collision Detection (CSMA/CD) (IEEE Std 802.3); Token Bus (IEEE Std 802.4); Token Ring (IEEE Std 802.5); MAC bridges (IEEE Std 802.1, IEEE Std 802.6), etc. This is further complicated by the fact that these working groups often publish multiple standards for different media (e.g., coaxial cable, fiber optic, twisted pair). This implies that integration into the MAC sublayer would probably impact multiple standards, and thus would only apply to a very limited market. Since the security concerns are similar for the different MAC standards, and since a common interface is now provided by ISO/IEC 15802-1: 1995 [2D-5][20], the logical choice is to not integrate security services into the MAC sublayer.

Traffic flow analysis was not considered to be a serious threat by the 802.10 LAN Security Working Group, but the MAC sublayer is the only place where prevention against traffic flow analysis can be successfully implemented. If traffic flow analysis is a concern for a given implementation, the MAC sublayer would need to be more closely examined.

[20]The numbers in brackets correspond to those of the bibliography in 2D.7.

2D.3 Between LLC and MAC or lower LLC

The standard places a security entity at the bottom of LLC. With the exception of management, it can be viewed logically between the LLC and MAC sublayers. There are only three primitives that flow between the MAC and LLC Layers: MA_UNITDATA request, the MA_UNITDATA indication, and the MA_UNITDATA_STATUS indication. (The contents of these requests currently differ slightly between the various MAC protocols, but there is an effort to determine a common MAC interface.) The simplicity of the interface and the protocol is the biggest advantage of placing the protocol between the MAC and the LLC sublayers.

There are many existing protocols other than LLC that request services directly from MAC (MAC clients). Even though this protocol is referred to as being between LLC and MAC, any protocol that implements the MAC service primitives can reside above the security protocol. This will prove to be an advantage in providing security for existing systems that may not implement LLC.

The security services are as described in 2.3 of the standard.

2D.4 Integrated into upper LLC

Integration into LLC provides several advantages if it is done correctly. The granularity of security decisions and enforcements can now be at the granularity of the LSAP instead of the MAC addresses. While this provides added granularity, it must be realized exactly what this means. Normally LSAP addresses are reserved for applications, not processes. For instance, there is a reserved LSAP for ISO Network Layer, and another LSAP is reserved for the DoD Internet Protocol. There are also locally administered LSAPs. These LSAPs could be used to separate between security levels, but then, what about the need for different security levels for those applications running above the reserved LSAPs?

LLC provides three types of operation.[21] The first type is a connectionless-mode operation that provides service across a data link with minimum protocol complexity. The second type of operation provides a connection-oriented service across a data link comparable to high-level data link control (HDLC). This service includes support of flow control, sequencing, and error recovery. There is no substantial difference in the security services that can be provided over the connectionless-mode operations and those that can be provided by a protocol operating between LLC and MAC. The connection-oriented service, however, can provide additional security services and can allow different mechanisms.

What advantages are provided by connection-oriented security services? With regard to confidentiality, there is no discernible difference to the service requestor between connection and connectionless confidentiality. However, if encryption is the mechanism used to provide that confidentiality, several advantages are gained if a connection-oriented confidentiality is provided. The first is that the key granularity can be based upon the connection, and not simply between the two peer entities. This provides advantages, since a different key can be used for different connections, providing better security in some cases. Since the key granularity is based on the connection, the protocol can discard keys after receiving disconnect messages for the connection. This is an advantage over connectionless, since connectionless has no concept of connection and uses a key cache of all recently used keys. The judicial use of the disconnects can reduce the size of the key cache in many systems.

The second advantage occurs from the fact that most encryption algorithms chain encryption blocks. A typical block size of an encryption algorithm is 64 bits (such as ANSI X3.92-1981 [2D-1]). If every 64 bits were enciphered separately, then an attacker could look for repetitions of the 64 bits, and thus gain an advantage

[21]The third type (Connectionless Acknowledged) became a standard in 1998. The IEEE 802.10 Working Group believes that the Connectionless Acknowledged service will be handled in the same manner as LLC type 2.

in breaking the code. To prevent this, there are different modes of operation (such as ANSI X3.106-1983 [2D-2]). These modes of operation make each encryption block dependent upon the preceding block(s). While this is nice cryptographically, the order that the blocks are enciphered shall be the same as the order of decryption. If a connectionless service is used, this chaining must start over for each PDU received, since they are unordered. In a connection-oriented service, the chaining can continue across multiple PDUs, thus possibly reducing the overhead of reinitializing the cryptographic algorithm after each PDU.

Connection-oriented integrity is a distinctly different service than connectionless integrity. Connectionless integrity only assures the service-requestor that the chance of unauthorized modification to a single PDU is exceedingly small. Connection-oriented integrity ensures that the data units arrive in sequence, and that all the data units over the connection have arrived.

The effect of providing connection-oriented integrity in LLC is very similar to providing a connection-oriented LLC over a connectionless-integrity layer between LLC and MAC. Since the SDE SDU would be encapsulated in the MA_UNITDATA request, the sequence numbers as well as the data within the SDE SDU would be protected against modification. The only remaining integrity protection is against truncation. Truncation involves an active-wiretap deleting the last of a message in the hope that a security breach can be caused by the uncompleted transaction. Since the Disconnect is sent enciphered, the interloper cannot generate the Disconnect request. The Disconnect packet does not contain the last received PDU; however, the sender treats all unacknowledged PDUs as if they had been lost. The receiver has no idea that the connection has been truncated, since there is no method in LLC to prevent the receiver from thinking that all the valid data has been sent. There would need to be a special Disconnect PDU that contained the last sequence number. Unfortunately, that would involve changing the way that LLC processes, since ISO/IEC 8802-2: 1998 [2D-4] requires that all previously sent information PDUs “that are unacknowledged when this command [Disconnect] is actioned shall remain unacknowledged.”

One additional advantage to a connection-oriented service is a function of the implementation of the LLC protocol. The acknowledgment is provided for PDUs, so the service requestor knows the PDU has been delivered if data origin authentication and integrity are provided. Note, however, as was true with sequencing, the same service is provided by a connection-oriented LLC operating above a protocol providing secure connectionless integrity and data origin authentication.

While the integrated version of LLC appears quite attractive, there are some disadvantages that convinced the 802.10 LAN Security Working Group not to choose this option. The most important reason is that all of the existing implementations of LLC would need to change. The connection-oriented security services as described above require changing the way that the PDUs are processed. From a standards point of view, this means that changes to the existing LLC standards will be required. From a vendor’s point of view, existing equipment would be made obsolete, and migration to a secure version would become difficult.

There are more security services that should be provided by the integrated version than the MAC/LLC proposal. The question is whether these additional security services justify the problems and added complexity. The simplicity of the MAC interface allows a very simple protocol. The integrated LLC protocol shall provide for both modes of operation as well as be extensible to new types of operation that may be defined in the future. It is unclear if all of these security services should be provided at Layer 2. A much more conservative view of the security services that can be provided at Layer 2 is taken in ISO 7498-2: 1989 [2D-3]. It does not allow the provisioning of Access Control and Integrity. While the 802.10 LAN Security Working Group maintains that this is inappropriate, it should be remembered that the LLC protocol is only at Layer 2, and there may be other higher layers that are more suited for providing these additional services.

2D.5 Above LLC

The protocol operating above LLC must be cognizant of the different operational modes of LLC (connection and connectionless). It must tailor its security services to account for these. As such, it will probably not be

as simple a protocol as the MAC/LLC protocol. It does have the added benefit of having the granularity of LSAP addresses instead of MAC addresses as did the protocol integrated with LLC.

The reason that it was decided not to seriously consider the placement above LLC is that the only security service added other than finer granularity is the connection-oriented security services as described in the subclause on the integrated LLC protocol. If the protocol is operating above LLC, it must duplicate much of the LLC processing if it is to provide these services. For instance, assume a PDU is received that fails the integrity check because it has been modified during transit. If the protocol claims to provide connection-oriented integrity, it cannot deliver the PDU to the next layer. Obviously, its peer shall attempt to resend. Unfortunately, the LLC protocol's error detection did not catch the error and it has already sent an acknowledgment. This necessitates the protocol above LLC to buffer PDUs, and set up a window just like the LLC layer. This involves redundant processing, and eventually becomes almost as complicated as the LLC protocol.

The protocol above LLC could just provide the connectionless services and become much more simple, but then the only motivation for choosing it over the MAC/LLC protocol would be the LSAP granularity.

2D.6 Conclusion

For the reasons stated above, the 802.10 LAN Security Working Group concluded that the best approach was to define a protocol operating between LLC and MAC or lower LLC. Some specific applications will need the additional security services provided by higher layers.

2D.7 Bibliography

[2D-1] ANSI X3.92-1981 (R1987), Data Encryption Algorithm.

[2D-2] ANSI X3.106-1983 (R1996), Modes of Operation for the Data Encryption Algorithm.

[2D-3] ISO 7498-2: 1989, Information processing systems—Open Systems Interconnection—Basic Reference Model—Part 2: Security Architecture.

[2D-4] ISO/IEC 8802-2: 1998 [ANSI/IEEE Std 802.2, 1998 Edition], Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 2: Logical link control.

[2D-5] ISO/IEC 15802-1: 1995, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 1: Medium Access Control (MAC) service definition.

Annex 2E

(normative)

Fragmentation

2E.1 Introduction

The SDE protocol can add additional fields to the data received from the SDE service interface (the SDE SDU) and increase the length of the resulting MAC SDU. This additional length may produce a MAC SDU longer than the maximum allowed MAC SDU length. This is not acceptable, because it would force a MAC sublayer error.

There are two basic methods of ensuring that the SDE sublayer does not generate MAC SDUs that are too long:

a) Data Link users can adjust their maximum PDU size to take into account of the additional SDE overhead.
b) SDE can fragment and reassemble Data Link user PDUs transparently to the Data Link user.

The fragmentation and reassembly of SDE SDUs increases the complexity and reduces the performance of the SDE sublayer. Thus the adjustment of the Data Link user's maximum PDU size is the preferred solution. It is not always possible, however, to modify the Data Link user's maximum PDU size. This annex specifies a uniform method for the SDE sublayer to provide fragmentation and reassembly when the Data Link user's maximum PDU size cannot be modified. The use of a uniform method is required to maintain interoperability among implementations supporting fragmentation and reassembly. Support for fragmentation remains optional in all implementations of SDE.

2E.2 Overview

The fragmentation and reassembly procedures will be performed only if the security association indicates fragmentation support.

This fragmentation and reassembly procedure splits an SDE SDU into two parts. Each part or fragment will be transmitted as a separate SDE SDU. A PDU that is a fragment of an SDE SDU is identified by a "fragmented" field in the protected headed. This field is set true when the PDU contains a fragment of an SDE SDU. Each fragment of a given SDE SDU is assigned the same fragment identifier. The fragment identifier is stored in the SDE SDU's protected header. The fragment identifier must be unique for the duration of the crypto-period. When an SDE SDU is fragmented, the two parts are distinguished by a boolean field in the protected header called "more segments." The first fragment has the "more segments" field set true, and the second fragment has the field set false. The relationship between an SDE SDU and its fragments in Figure 2E.1.

On transmission, the procedure defined by this recommendation fragments the SDE SDU after the SDE entity finds a valid security association for a SDE_UNITDATA.request. Next, the proper fragmentation information is calculated and placed in the protected header of both fragments. Then each fragment is processed independently and finally forwarded to the MAC sublayer. Note that fragmentation occurs before encryption.

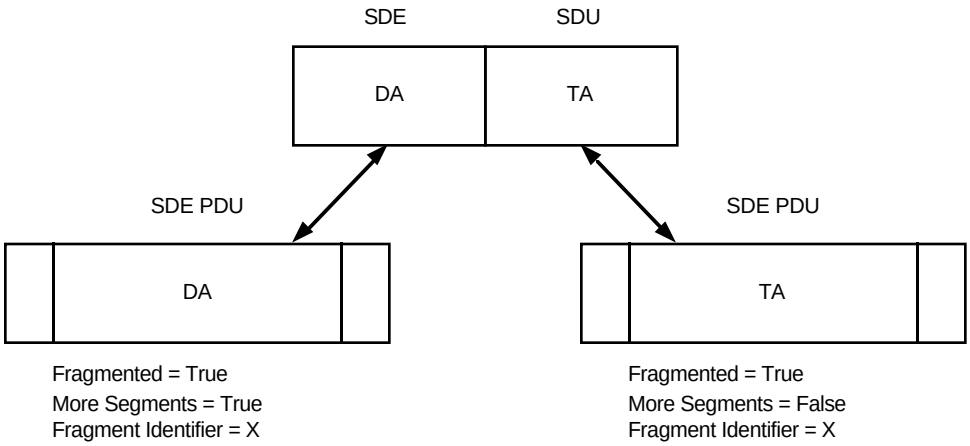


Figure 2E.1—Fragment association

On receipt of an SDE PDU from the MAC sublayer, all of the SDE PDU processing is performed before SDE PDU reassembly is attempted. Therefore, when necessary, the SDE PDUs have been decrypted and had their integrity verified. Each security association maintains a list of PDUs awaiting assembly. This set is searched for the other fragment of the SDE SDU. The other fragment is located by finding the SDE PDU with an equal fragment identifier and a different “more segments” value. If a match is found, the SDE SDU is reassembled using the information in the protected header and SDE security processing is continued as usual on the reassembled SDU. If a match is not found, the PDU is placed in the set of PDUs awaiting reassembly.

2E.3 Additional station objects

The SDE entity must be able to indicate whether it can support fragmentation. Therefore, implementations that support fragmentation must have the **Station_Fragmentation_Enabled** station object set to true. This object is used by key management when negotiating fragmentation support at security association initialization.

a) **Station_Reassembly_Timer**

INTEGER—The number of milliseconds an SDE entity will store a received SDE PDU that contains a fragment of an SDE SDU.

b) **Station_Reassembly_Expiration_Count**

INTEGER—The number of SDE PDUs that have been discarded by the SDE entity when the SDE reassembly timer has expired.

c) **Station_Receive_Fragment**

INTEGER—The number of SDE PDUs that contain SDE SDU fragments received by this station.

2E.4 SAP objects

The procedure defined in this annex must reference the SAP_Max_SDE_SDU SAP object. This object will be defined in 2.8, SDE sublayer management.

2E.5 Additional association object

The SDE must be able to determine whether a given association supports fragmentation. Therefore, stations that support fragmentation must have the **Assoc_Frag_Enab** object defined for each security association. Security associations that support fragmentation must have the **Assoc_Frag_Enab** object set to true.

2E.6 Additional Protected Header fields

When a security association supports fragmentation, the following two additional fields must be added to the protected header:

a) Flags
b) Fragment identifier

Two examples of the Protected Header formats for an association that supports fragmentation are shown in Figure 2E.2. Both examples assume that association object ID_pres is true. The first example shows the header format for a PDU that contains a fragment; the other example is for a PDU that does not contain a fragment.

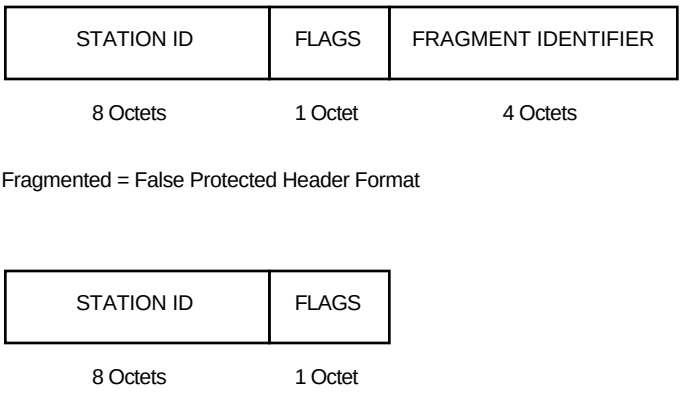


Figure 2E.2—Example of Protected Header formats

2E.6.1 Flags field

The Flags field is a mandatory field in the Protected Header when Assoc_Frag_Enab is true. The format of the field is shown in Figure 2E.2. If ID_pres is true, then the Flags field follows the Station ID field. If ID_pres is false, then the Flags field is the first field in the Protected Header. The Flags field contains two subfields used for fragmentation and reassembly: Fragmented and More Segments.

a) **Fragmented**

This is a boolean field. When the value of this field is true, it indicates that the SDE PDU is a fragment of an SDE SDU and that the Fragment Identifier field follows the Flag field.

b) **More Segments**

This is a boolean field that is only meaningful if the Fragmented field is true. This field is used to indicate the SDE PDU fragment number. If the value of this field is true, the SDE PDU contains the first fragment of the SDE SDU. If it is false, it contains the second part of a fragmented SDU SDU.

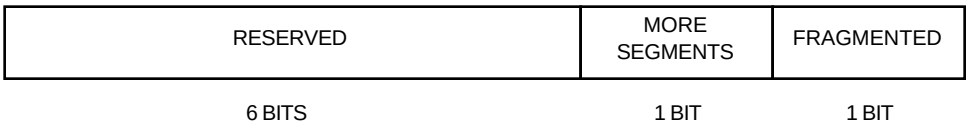


Figure 2E.3—Flags field format

2E.6.2 Fragment Identifier field

If the Flags field indicates that an SDE PDU contains a fragmented SDE SDU, then the Fragment Identifier field follows the Flags field in the Protected Header. The Fragment Identifier field is used to associate SDE PDUs with the SDE SDU from which they were derived. The fragment identifier is four octets long.

In order to protect against integrity attacks on fragments, the security association must be rekeyed before the Fragment Identifier field reuses identifier values. This implies that the SDE entity must be able to inform the key management entity when the fragment identifier space is exhausted.

2E.7 Detailed functional specification

2E.7.1 SDE_UNITDATA.request

The following steps are performed after finding the security association (2.7.4.1[22]) for the SDE_UNITDATA.request and when the security association has the Remote_SDE equal to true and Assoc_Frag_Enab equal to true.

a) If the length of the SDE SDU is greater than the maximum SDE SDU length (SAP_Max_SDE_SDU), then perform the following steps:
   1) Generate a fragment identifier. The value of this identifier must be different from all other values used on this association. If a unique fragment identifier value cannot be generated, then inform layer management. The handling of this event by layer management is a local manner.
   2) Split the SDE SDU (or the SDE_UNITDATA.request Data field) into two pieces. Each piece must be small enough so that resulting MAC SDUs are smaller than the maximum MAC SDU length when SDE processing is complete.
   3) Build the fragmentation part of the Protected Header with the Fragmented field set true, More Segments set true, and the value of the generated identifier in the Fragment Identifier field. Prepend these fields to the first fragment.
   4) Build the fragmentation part of the Protected Header with the Fragmented field set true, More Segments set false, and the value of the generated identifier in the Fragment Identifier field. Prepend these fields to the second fragment.

b) If the length of the SDE SDU is not greater than the maximum SDE SDU length (SAP_Max_SDE_SDU), then prepend the Flags field to the MSDU specified in the SDU_UNITDATA.request. The Flags field has the Fragmented field set to false.

c) Forward the outgoing PDU (or both PDUs if the SDU has been fragmented) to the Forming the Protected Header step (2.7.4.3).

[22]All clause numbers in this annex refer to Clause 2 of the standard.

2E.7.2 SDE_UNITDATA.indication

The following steps are performed following the Station ID step (2.7.5.5) in the reception procedures.

If Remote_SDE equals true, and Assoc_Frag_Enab equals true, and the Fragmented field in Flags field equals true, then perform the following:

a) Increment Station_Receiver_Fragment station object.
b) Each association has a set of SDE PDUs awaiting reassembly. This set is searched for an SDE PDU that has a fragment identifier that is equal to the fragment identifier of the received SDE PDU.
c) If a matching fragment identifier is found, then perform the following:
    1) Compare the More Segments subfield of the matching SDE PDUs.
    2) If the subfield values are different, perform the following:
        i) Stop the reassembly timer for the stored SDE PDU.
        ii) Concatenate the two fragments together.
        iii) Continue normal processing with the combined SDE PDUs.
d) If a matching fragment identifier is not found then perform the following:
    1) Place the SDE PDU in the set of PDUs awaiting reassembly.
    2) Start the reassembly timer for this SDE PDU.
    3) Signal layer management entity of a fragmentation event (Station_Receiver_Fragment).

2E.7.3 Build Protected Header

This recommendation requires a slight modification of the SDE Build Protected Header function (2.7.4.3). This function must always insert the flags field in the Protected Header if the security association supports fragmentation (Assoc_Frag_Enab = True).

2E.7.4 Station_Reassembly_Timer

The primary function of this timer is to provide a bound for which an SDE PDU will be held for reassembly. When a fragmented PDU is received, the reassembly timer is started. If the SDE PDU is not reassembled before the timer expires, the SDE PDU is discarded and the layer management entity is notified. The management entity will increment the Station_Reassembly_Expiration_Count object. The timer value is a station object (Station_Reassembly_Timer) and the value will be a local issue.

Annex 2F

(normative)

ASN.1 encodings

2F.1 Unrestricted ASN.1 types

Unrestricted datatypes in ASN.1 can be specified for simple types as follows:

boolean type	BOOLEAN
integer type	INTEGER
numerated type	ENUMERATED
real type	REAL
bit string type	BIT STRING
octet string type	OCTET STRING
null type	NULL
character string type	NumericString PrintableString VisibleString IA5String GraphicString GeneralString .....

2F.2 Imported encodings

The following registered object identifier root is used for the imported information objects:

ieee802-1partf = iso(1) member-body(2) us(840) ieee-802dot1partF(10011)

The following imported encodings are used to support the defined managed objects:

ieee802-1partf.asn1Module.commondefinitions.version1.MACAddress
ieee802-1partf.asn1Module.commondefinitions.version1.ResourceInfo
ieee802-1partf.asn1Module.commondefinitions.version1.ManufacturerOUI
ieee802-1partf.asn1Module.commondefinitions.version1.ManufactureName
ieee802-1partf.asn1Module.commondefinitions.version1.ManufacturerProductName
ieee802-1partf.asn1Module.commondefinitions.version1.ManufacturerProductVersion
ieee802-1partf.asn1Module.commondefinitions.version1.ResourceTypeIDName

2F.3 ASN.1 productions

An SDE managed object shall return an ASN.1 type NULL for any operation, object, managed object, or attribute that is not supported or not valid.

```
IEEE802.10_SDE_ASN1Module  {iso(1) member-body(2) us(840) ieee802.10(10022) asn1Module(2)
commondefinitions(1) }

DEFINITIONS ::= BEGIN

IMPORTS
        MACAddress,
        ResourceInfo,
        ManufacturerOUI,
        ManufacturerName,
        ManufacturerProductName,
        ManufacturerProductVersion,
        ResourceTypeIDName
        FROM IEEE802CommonDefinitions { iso(1) member-body(2) us(840)
ieee-802dot1partF(10011) asn1Module(2) commondefinitions(0) version1(0) } ;

IMPORTS
        AttributeId,
        ObjectInstance
        FROM CMIP-1 {joint-iso-ccitt ms(9) cmip(1) version1(1) protocol(3) } ;

IMPORTS
        AdditionalText,
        AdditionalInformation,
        AttributeIdentifierList,
        AttributeValueChangeDefinition,
        CorrelatedNotifications,
        Count,
        CounterThreshold,
        ManagementExtension,
        NotificationIdentifier,
        SourceIndicator
        From Attribute-ASN1Module {joint-iso-ccitt ms(9) smi(3) part2(2) asn1Module(2) 1 } ;

IMPORTS
        AttributeValueChangeInfo
        FROM Notification-ASN1Module {joint-iso-ccitt ms(9) smi(3) part2(2) asn1Module(2) 2 } ;

EXPORTS -- everything

SDE_Integer     ::= INTEGER

SDE_Boolean     ::= BOOLEAN

SDE_Ostring     ::= OCTET STRING

MACAddresses    ::= CHOICE      {
                        individual              [0] MACAddress,
                        groupaddress            [1] SEQUENCE OF MACAddress
                        }

Station_name    ::= GraphicString {"SDE_Station"}

SAP_ID          ::= GraphicString
```

```
SA_ID           ::= GraphicString

ALabel_Set      ::= OBJECT IDENTIFIER

Label_Values    ::= SEQUENCE OF OCTET STRING

SLabel_Sets     ::= SEQUENCE OF OBJECT IDENTIFIER

CreateError1    ::= ENUMERATED {

                        unknown                 (0),
                        unsupported             (1),
                        duplicate               (2),
                        outOfMemory             (3),
                        processError            (4),
                        infoMissing             (5),
                        bufferingUnsupported    (6),
                        dynamicUnsupported      (7),
                        fragmentationUnsupported (8) }

CreateError2    ::= ENUMERATED {

                        unknown                 (0),
                        unsupported             (1),
                        duplicate               (2),
                        outOfMemory             (3),
                        processError            (4),
                        infoMissing             (5),
                        limitExceeded           (6),
                        dynamicUnsupported      (7),
                        sduSizeUnsupported      (8) }

CreateError3    ::=  ENUMERATED {

                        unknown                 (0),
                        unsupported             (1),
                        duplicate               (2),
                        outOfMemory             (3),
                        processError            (4),
                        infoMissing             (5),
                        limitExceeded           (6),
                        dynamicUnsupported      (7),
                        remFragUnsupported      (8) }
END
```

Annex 2G

(normative)

Allocation of object identifier values

This annex contains a summary of all managed object identifier values that have been allocated by this standard.

Each table shows allocations related to a specific category of information object. The heading of the table identifies the category of information object, and shows the invariant part of the object identifier value allocated to the entries in the table. In cases of discrepancy between an identifier value in an allocation table and in the REGISTERED AS construct of a GDMO template, the template value takes precedence. The column labeled *Arc* shows the value allocated to the arc subsequent to the invariant part, which completes the object identifier value allocated. The column labeled *Purpose* contains a text description of the information object, and, in the case of current allocations, a reference to the location of the definition of the information object in the standard. The column labeled *Status* shows the status of the allocated values, using the following convention:

R *Reserved.* The object identifier value is reserved for future use by this standard.
C *Current.* The object identifier value has been allocated to an information object that is defined within the current revision of the standard.
D *Deprecated.* The object identifier value has been allocated to an information object that was defined in a previous revision of the standard, and whose use is now deprecated.

Allocations for standard-specific extension types.
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee-802dot10(10022) standardSpecificExtension(0)}

ARC	PURPOSE	STATUS
extension(0)	Reserved for future use	R
sdedefinitions(1)	Common definition module for the SDE sublayer	C

Allocations for ASN.1 module identifiers.
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee-802dot10(10022) asn1Module(2)}

ARC	PURPOSE	STATUS
extension(0)	Reserved for future use	R
sdedefinitions(1)	Common definition module for the SDE sublayer	C

Allocations for Managed Object classes.
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee-802dot10(10022) managedObjectClass(3)}

ARC	PURPOSE	STATUS
extension(0)	Reserved for future use	R
SDE_Station(1)	SDE Station managed object class name	C
SDE_SAP(2)	SDE SAP managed object class name	C
Security_Association(3)	Security Association object class name	C

Allocations for Package identifiers.
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee-802dot10(10022) package(4)}

ARC	PURPOSE	STATUS
extension(0)	Reserved for future use	R
station_package(1)	Mandatory package for the SDE Station	C
buffer_package(2)	Package for buffer management elements	C
sde_frag_package(3)	Package for fragmentation elements	C
sde_sap_package(4)	Mandatory package for SDE_SAP elements	C
oversize_pdu_package(5)	Package to handle oversize PDUs	C
said_package(6)	Mandatory package for security associations	C
remote_frag_package(7)	Package for remote fragmentation elements	C
dynamic_SDE_package(8)	Package for dynamic management of SDE	C
dynamic_SAP_package(9)	Package for dynamic management of SAP	C
dynamic_SA_package(10)	Package for dynamic management of SA	C
sde_security_label_package(11)	security label-related station information	C
security_label_package(12)	security label-related association information	C

Allocations for Parameter identifiers.
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee-802dot10(10022) parameter(5)}

ARC	PURPOSE	STATUS
extension(0)	Reserved for future use	R
createerror1(1)	Error information returned to manager	C
createerror2(2)	Error information returned to manager	C
createerror3(3)	Error information returned to manager	C

Allocations for Name Binding identifiers.
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee-802dot10(10022) nameBinding(6)}

ARC	PURPOSE	STATUS
extension(0)	Reserved for future use	R
station-system(1)	Name binding for the SDE_Station managed object instance	C
sde_resourcetypeId(2)	ResourceTypeID managed object instance name binding for the SDE sublayer	C
sde_sap-sde_station(3)	Name binding for each instance of the SDE_SAP managed object	C
security_association-sde_sap(4)	Name binding for each instance of the Security Association managed object	C
station-system2(5)	Name binding for a dynamically created instance of a SDE_Station managed object	C
sde_sap-sde_station2(6)	Name binding for each dynamically created instance of the SDE_SAP managed object	C
security_association-sde_sap2(7)	Name binding for each dynamically created instance of the SA managed object	C

Allocations for Attribute identifiers.
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee-802dot10(10022) attribute(7)}

ARC	PURPOSE	STATUS
extension(0)	Reserved for future use	R
sde_station_name(1)	SDE Station Name attribute	C
station_clear_header(2)	Station Clear Header boolean attribute	C
station_mdf(3)	Station MDF boolean attribute	C
fragmentation_enable(4)	Station Fragmentation Enable boolean attribute	C
max_mac_sdu_size(5)	Maximum size of a PDU before fragmentation	C
station_reassembly_timer(6)	Fragmentation reassembly timer in milliseconds	C
station_reassembly_expiration_ count(7)	Count of fragmentation reassembly timeouts	C
station_receive_fragment(8)	Count of received fragments	C
buffersize(9)	Total buffer space in octets	C
maxbufferusesize(10)	Maximum of buffer usage at any time	C
avgbufferusesize(11)	Average buffer usage in octets	C
sap_ID(12)	SDE_SAP naming attribute	C
sap_worst_case_expansion(13)	Maximum size an SDU can be expanded	C
sap_max_SDE_sdu(14)	Maximum sdu size that can be sent to MAC layer	C

Allocations for Attribute identifiers. ***(continued)***
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee-802dot10(10022) attribute(7)}

ARC	PURPOSE	STATUS
sap_exceed_size_cnt(15)	Count of SDUs that have been oversize	C
sap_exceed_size_thres(16)	Threshold for oversize event trigger	C
sap_max_oversize_short(17)	Size of shortest oversize PDU	C
sap_max_oversize_long(18)	Size of largest oversize PDU	C
sa_ID(19)	Security Association naming attribute	C
local_SAID(20)	Unique number identifying a security association	C
remote_SAID(21)	Number identifying remote security association	C
assoc_MDF(22)	Boolean for MDF presence	C
remote_MDF(23)	MDF value for outgoing SDE PDU	C
confid(24)	Data confidentiality presence boolean	C
integ(25)	Data integrity presence boolean	C
padding_pres(26)	Padding presence boolean	C
ID_pres(27)	Station Id presence boolean	C
confid_alg_ID(28)	Confidentiality algorithm label attribute	C
integ_alg_ID(29)	Integrity algorithm label attribute	C
sde_sap(30)	Association SAP attribute	C
remote_sde(31)	Remote SDE station presence boolean	C
outgoing_source_mac_address(32)	Source address for outgoing packet	C
outgoing_destination_mac_address(33)	Destination address for outgoing packet	C
incoming_source_mac_address (34)	Local station address attribute	C
incoming_destination_mac_address(35)	Sender’s address attribute	C
assoc_frag_enab(36)	Remote station fragmentation boolean	C
badpdusICVcount(37)	Count of PDUs with bad ICVs	C
badpdusICVthreshold(38)	Threshold count for PDUs with bad ICVs	C
badpdusSAIDcount(39)	Count of PDUs with bad SAID processing	C
badpdusSAIDthreshold(40)	Threshold count for PDUs with bad SAID processing	C

Allocations for Attribute identifiers. ***(continued)***
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee-802dot10(10022) attribute(7)}

ARC	PURPOSE	STATUS
badpduscount(41)	Count of PDUs with an invalid SAID	C
badpdusthreshold(42)	Threshold count for PDUs with an invalid SAID	C
bufferproblemscount(43)	Count of PDUs discarded due to buffer problems	C
bufferproblemsthreshold(44)	Threshold count for PDUs discarded due to buffer problems	C
station_security_label_enabled(45)	station supports security labels	C
station_security_label_sets_allowed(46)	station-supported security label sets	C
station_security_label_values(47)	station-supported security label values	C
assoc_label_set(48)	security label set for the association	C
assoc_label_value(49)	label value(s) for the association	C
assoc_label_explicit(50)	association requires labels on all SDE PDUs	C

Allocations for Attribute Group identifiers.
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee-802dot10(10022) attribute Group(8)}

ARC	PURPOSE	STATUS
extension(0)	Reserved for future use	R

Allocations for Action types.
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee-802dot10(10022) action(9)}

ARC	PURPOSE	STATUS
extension(0)	Reserved for future use	R

Allocations for Notification types.
Invariant part of object identifier value =
{iso(1) member-body(2) us(840) ieee-802dot10(10022) notification(10)}

ARC	PURPOSE	STATUS
extension(0)	Reserved for future use	R
bufferproblemsevent(1)	Event indicating reception of a threshold number of PDUs causing buffering problems	C
badpdusSAIDdiscarded(2)	Event indicating reception of a threshold number of SDE PDUs with valid ICVs but causing processing errors	C
sap_exceed_size_event(3)	Event indicating threshold of oversize PDUs	C
badpdusICVdiscarded(4)	Event indicating reception of a threshold number of SDE PDUs with invalid integrity check values	C
badpdusdiscarded(5)	Event indicating reception of a threshold number of invalid SDE PDUs without a valid security association	C

Annex 2H

(informative)

Recommended practice for SDE with IEEE 802.3 Type-encoded frames

2H.1 Introduction and scope

IEEE Std 802.10-1998 defines a standard for provisioning security services over IEEE 802 LANs and MANs. The standard provides facilities for the protection of traffic between stations attached to IEEE 802 LANs of different MAC types. This recommended practice extends those facilities to include the protection of 802.3 Type-encoded traffic [2H-2][23], also known as Ethernet V2.0 [2H-5].

If this recommended practice is not followed and the Ethernet Type is carried in plain text in the Ethernet header, then

a) No confidentiality or integrity services are provided for the Ethernet Type.
b) Management stations that promiscuously read LAN/MAN traffic will report a significant number of errors because the SDE ciphertext does not conform to the PDU format specified by the Ethernet Type.

For the purpose of facilitating interoperability of bridges and end stations that implement SDE in environments composed of a mixture of IEEE 802.3 Type-encoded and Length-encoded implementations, this recommended practice specifies extensions to the behavior of SDE implementations in order to support protection of both IEEE 802.3 Type-encoded and Length-encoded frame formats on a single LAN.

It is not the intent of this recommended practice to provide protected communications between two end stations on the same physical LAN that are not otherwise capable of communicating.

2H.2 Processing steps

Frames requiring Ethernet Types are mapped onto IEEE 802 MAC frames with SDE protection applied as shown in Figure 2H.1.

2H.2.1 Conversion of IEEE 802.3 Typed-encoded frames

The steps involved for conversion are as follows:

a) To form the LLC header, use the SNAP SAP values for the DSAP and SSAP fields and unnumbered information (UI) command in the Control field. The SNAP Protocol Control Information is formatted using the ISO/IEC 11802-5: 1997 [2H-4] OUI (00-00-F8) as octets 0, 1, and 2; and the Ethernet Protocol Type as octets 3 and 4.
b) Copy the Ethernet SNAP PDU to the LLC Data field immediately following the LLC Protocol Control Information.
c) Perform SDE transmission processing.

[23]Information on bibliographic references can be found in 2H.3.

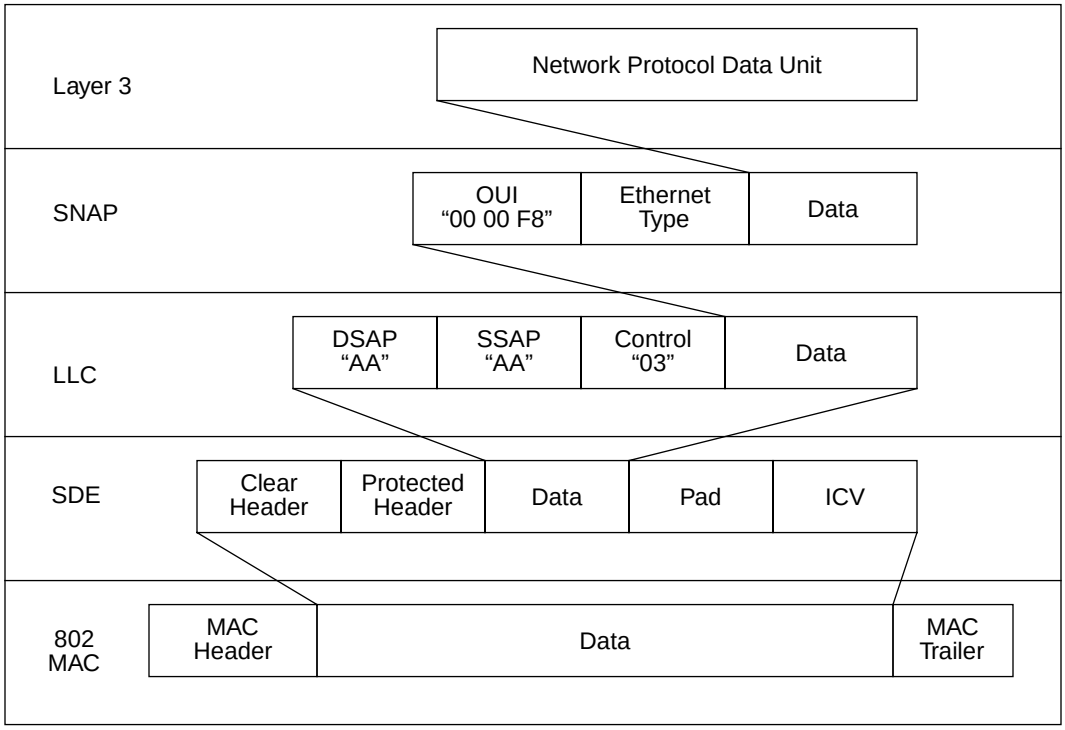


Figure 2H.1—LLC with SNAP protected by SDE

2H.2.2 Reception processing of SDE-protected IEEE 802.3 Type-encoded MAC frames

The steps involved in conversion are as follows:

a) Perform SDE reception processing.
b) If the receiving entity is implemented in an SDE end station or an SDE bridge that attaches to an Ethernet LAN, then perform the following additional two steps:
   1) Copy the Ethernet Protocol Type from octets 3 and 4 of the SNAP SAP Protocol Identifier.
   2) Copy the Network Protocol Data Unit to the Ethernet Data.
c) If the receiving entity is not an SDE end station or an SDE bridge connected to an Ethernet LAN, then the SNAP protocol information remains. The decapsulation will be subsequently performed by a bridge in accordance with ISO/IEC 11802-5: 1997 [2H-4].

2H.3 Bibliography

[2H-1] IAB RFC 1042, February 1988, Postel, J. B. and Reynold, J. K., A Standard for the Transmission of IP Datagrams over IEEE 802 Networks.

[2H-2] IEEE Std 802.3, 1998 Edition, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications (subclause 3.2.6).

[2H-3] ISO/IEC 8802-2: 1998 [ANSI/IEEE Std 802.2, 1998 Edition], Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 2: Logical link control.

[2H-4] ISO/IEC 11802-5: 1997 (ANSI/IEEE Std 802.1H, 1997 Edition), Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Technical reports and guidelines—Part 5: Media Access Control (MAC) Bridging of Ethernet V2.0 in Local Area Networks.

[2H-5] “The Ethernet—A Local Area Network: Data Link Layer and Physical Layer (Version 2.0),” November 1982, Digital Equipment Corporation, Intel Corporation, and Xerox Corporation.

Annex 2I

(normative)

Secure data exchange security label

2I.1 Introduction

This annex specifies a security label option for SDE. The optional security label carries information that specifies protective measures and identifies handling conventions required by a communications security policy. The security label consists of a named set of security tags whose use is defined via registration. The registration process associates a unique name to each security tag set definition enabling implementations to identify the semantics for the processing of the labels. A common register, identified in Annex 2J, may be used for the registration of tag sets for this label option. The set of security tags used on SDE PDUs is determined during the establishment of a security association and remains constant for the life of the association. The Tag Set Name for each security association is maintained in the Security Management Information Base (SMIB) along with other association attributes.

Support for security labels is optional in all implementations of SDE.

2I.2 Overview

The SDE security label is based on the Standard Security Label (SSL) (FIPS PUB 188: 1994). The SSL is defined as a collection of one or more Named Tag Sets. Every Named Tag Set contains a set of tags carrying security information preceded by an identifier (Tag Set Name). That identifier gives a reference to a register entry where the tag set and its associated semantics are defined. The security tags carry security attributes of the data being exchanged. Security attributes may be represented in several ways, thus the need for several tag types. A Named Tag Set and its components are shown in Figure 2I.1.

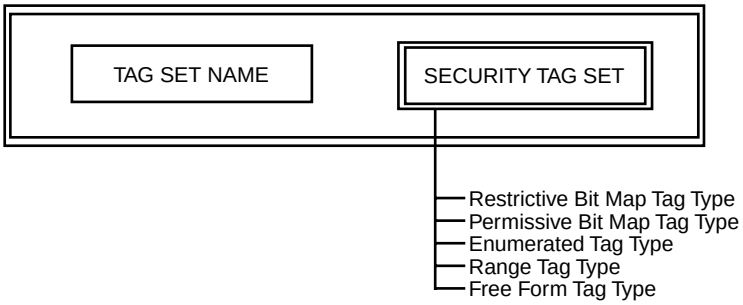


Figure 2I.1—SSL Named Tag Set

The SDE security label includes a single Named Tag Set. If present, the label follows the Station ID, Flags, and Fragment Identifier fields in the SDE Protected Header. Figure 2I.2 shows the placement of the security label in the SDE PDU structure when all optional fields are present.

In contrast to SSL, the SDE label supports a single tag set per label. The Tag Set Name identifier for a label is given by the security association and not carried on every SDE PDU. The set of security tags and the allowable values are established by the communicating parties as part of the establishment of the security

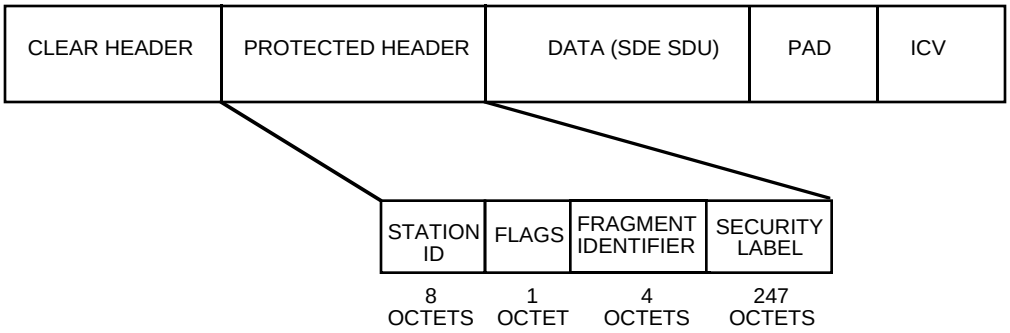


Figure 2I.2—SDE PDU structure

association. A label need not appear on every SDE PDU even if one is negotiated as part of a security association. For instance, if the security association accepts only one label value, that value is implied by the association and need not be carried on every PDU. Implementors should be aware that the use of security labels could increase the possibility of exceeding the maximum PDU size, thus increasing the need to support fragmentation.

2I.3 SDE security label option

The security label option begins with a one-octet length field followed by a variable-length security tag set. Figure 2I.3 shows the format for the security label. For all fields shown, assume that the most significant bit (MSB) is on the left and the least significant bit (LSB) is on the right.

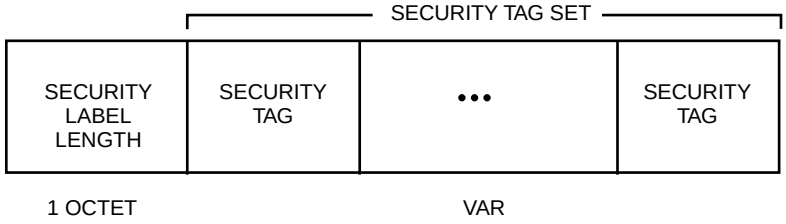


Figure 2I.3—SDE security label format

As stated in Annex 2E, adding fields (e.g., a security label) to an SDE SDU increases the length of the resulting MAC SDU. The additional length may cause a MAC sublayer error if the resulting SDU is longer than the maximum allowed MAC SDU length. One method for preventing the generation of MAC SDUs that are too long is for SDE to fragment and reassemble Data Link user PDUs transparently to the user. Upon fragmentation, the label shall be copied onto all generated SDE PDUs.

2I.4 Security label length

This field is one octet in length. Its value is the total length of the label option in octets including the length field. The maximum length value is 247. Note that maximum length may be further restricted by the failure to support fragmentation and other implementation-specific tradeoffs.

2I.5 Security tags

A common format for passing security related data is necessary for interoperability. This standard defines five types of security tags to carry security attributes of the data in a PDU. Each Security Tag has a one-octet type field and a one-octet length field plus a variable size data field. Only tags 1, 2, 5, 6, and 7 are defined; this standard reserves all other tag types for future use. Figure 2I.4 shows the general format for security tags.

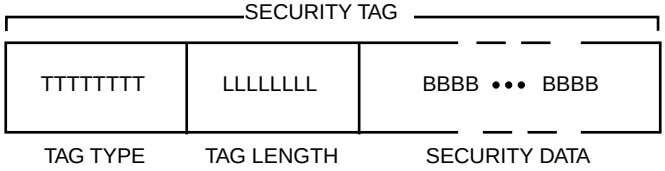


Figure 2I.4—General security tag format

2I.5.1 Tag type

This field is 1 octet in length and identifies the format used to represent the security data, e.g., bit map, list of two-octet attribute numbers, pairs of attribute numbers, etc.

2I.5.2 Tag length

This field is 1 octet in length. Its value is the total length of the tag including the type and length fields.

2I.5.3 Security data

This is a variable-length field. It carries security attributes of the data in the PDU. For Tag Types 1, 2, 5, and 6, the Security Data field begins with a one-octet subfield containing a security level. Its value may range from 0 to 255. The values are ordered with 0 being the lowest security level and 255 representing the highest security level. This field is used to convey a restrictive hierarchical security attribute. The format of the rest of the Security Data field is different for each of these four tag types and is defined below. The format of the Security Data field of Type 7 tags is defined through registration for each Named Tag Set.

2I.6 Security Tag Type 1

Tag Type 1 is the Restrictive Bit Map Tag Type. Tags of this type are used to convey restrictive security parameters, such as compartments and protection categories, that may be selected from a set by setting a one-bit flag. Security attributes conveyed by this tag type are used to limit the entities allowed to access the data in the PDU to those whose security level is higher or equal to the level of the data and whose set of non-hierarchical attributes is a superset of that for the data. The format of this tag type is as shown in Figure 2I.5.

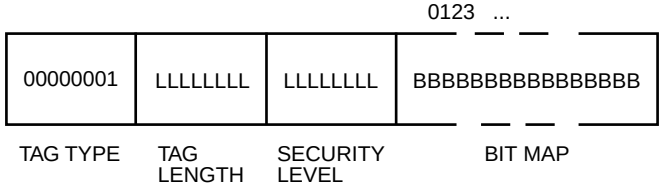


Figure 2I.5—Security Tag Type 1 format

2I.6.1 Restrictive security attribute bit map

The length of this field is variable. The maximum length is 243 octets and the minimum is 0 octets. The ordering of the bits is left to right or MSB to LSB. For example, security attribute 0 is represented by the MSB of the first octet and security attribute 15 is represented by the LSB of the second octet. Bit maps shall be padded with zeros to the right (i.e., up to the LSB of the last octet), if necessary. Figure 2I.6 graphically shows this ordering.

Bit *n* is binary 1 if attribute *n* is part of the label for the PDU, and bit *n* is binary 0 if attribute *n* is not part of the label.

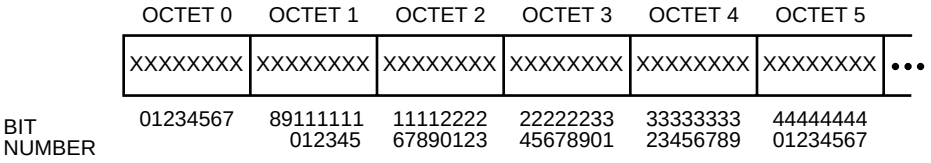


Figure 2I.6—Bit ordering for Bit Map Tags

2I.7 Security Tag Type 2

Tag Type 2 is the Enumerated Tag Type. Tags of this type are used when only a few security attributes, out of a large set, apply to the data in a given PDU. This is done by assigning a two-octet non-negative binary number to each security attribute and enumerating those attributes that apply. The enumerated attributes in Type 2 tags may be either permissive or restrictive. The registered specification for the tag set shall indicate whether the attributes on a Type 2 tag are permissive or restrictive. Permissive and restrictive attributes shall not be combined on a single tag. The format of this tag type is as shown in Figure 2I.7.

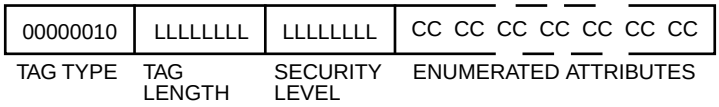


Figure 2I.7—Security Tag Type 2 format

2I.7.1 Enumerated attributes

In tags of this type, security attributes are listed by their assigned number value rather than by their position within a bit field. A two-octet number is used to identify each security attribute. Valid values for security attributes are 0 to 65 534. Attribute value 65 535 is not a valid attribute value. The maximum length for this field is 242 octets, i.e., up to 121 attributes may be listed.

Note that the two-octet numbers could be used to convey ASCII character pairs as an alternative way of identifying security attributes.

2I.8 Security Tag Type 5

Tag Type 5 is referred to as the Range Tag Type. It is used to represent labels where all attributes in a range, or set of ranges, apply to the data in a PDU. The attribute ranges in Type 5 tags may be either permissive or

restrictive. The registered specification for the Named Tag Set shall indicate whether the attributes on a Type 5 tag are permissive or restrictive. Permissive and restrictive attributes shall not be combined on a single tag. The format of this tag type is as shown in Figure 2I.8.

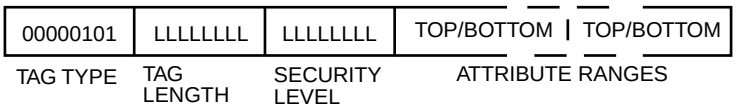


Figure 2I.8—Security Tag Type 5 format

2I.8.1 Security attribute ranges

Attribute ranges are pairs of two-octet values that represent the top and bottom security attributes of a range, respectively. These range endpoints are included within the range of attributes. All attributes within a range apply to the data in the PDU. The ranges must be non-overlapping and be listed in descending order. The bottom attribute endpoint for the last pair in the tag may be omitted when its value is 0. The maximum length of this field is 242 octets. This allows for a maximum of 61 ranges of which the last one has a lower bound of 0. Valid values for security attributes range from 65 534 to 0. Attribute value 65 535 is not a valid attribute value.

2I.9 Security Tag Type 6

Tag Type 6 is the Permissive Bit Map Tag Type. The bit map on tags of this type conveys permissive security attributes, such as release markings, that may be selected from a set by resetting a one-bit flag. Permissive security attributes conveyed by this tag type are used to indicate groups of entities allowed to access the data in the PDU. The format of this tag type is as shown in Figure 2I.9.

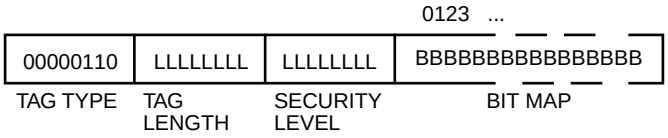


Figure 2I.9—Security Tag Type 6 format

2I.9.1 Permissive Security Attribute Bit Map

The length of this field is variable. The maximum length is 243 octets and the minimum is 0 octets. Bits in the map shall be numbered left to right starting with the MSB of the first transmitted octet. For example, security attribute 0 would be represented by the MSB of the first octet while security attribute 15 would be represented by the LSB of the second octet. Figure 2I.6 graphically shows this ordering. Bit maps shall be padded with ones to the right (i.e., up to the LSB of the last octet), if necessary.

Bit *n* is binary 0 if entities in group *n* are allowed to access the data in the PDU, and bit *n* is binary 1 if entities in group *n* are not allowed access.

2I.10 Security Tag Type 7

Tag Type 7 is the Free Form Tag Type. Tags of this type are used to convey a free format field of up to 244 octets. The Security Data field of this tag (i.e., Free Form field) may hold character strings, or any user-

defined data relevant to Layer 3 processing. See “Security Labeling Framework for the Internet” (Internet RFC 1457: 1993) for a discussion on relevant security data. The format of that data must be specified via registration.

The format of this tag type is as shown in Figure 2I.10.

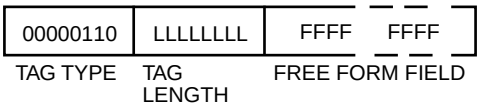


Figure 2I-10—Security Tag Type 7 format

2I.11 Additional station objects

The use of security labels on SDE-protected communications is optional and depends on the functionality supported by an implementation and the governing communications security policy. Stations that implement security labels shall have the Station_Security_Label_Enabled station object set to TRUE to signal to Key Management that a security label should be negotiated when attempting to establish a security association. The following station objects shall also be available to Key Management for the negotiation of the use of labels under a specific security association:

a) *Station_Security_Label_Sets_Allowed.* An ordered list of object identifiers indicating the registered Tag Set Names supported by the station. By being present on the list it is assumed that the security policy approves of the use of a label set. The order of the list should indicate order of preference of use. Only one of the Tag Set Names on the list may be successfully negotiated for any security association.

b) *Station_Security_Label_Values.* An ordered list of octet strings containing security attribute values accepted by the station. The registered definition of a Tag Set provides a format for conveying security information. The Station_Security_Label_Values list gives the policy-approved set of values that may be represented by the Tag Set. The security attribute values negotiated for any security association must be a subset of this list.

2I.12 Additional association objects

SDE relies on the information stored in the SMIB to determine the appropriate processing for SDE-protected PDUs. To enable label processing, the following objects must be created during the establishment of the security association:

a) *Assoc_Label_Set.* An object identifier corresponding to the Tag Set Name for the registered security label syntax specification. This value must be a subset of the Station_Security_Label_Sets_Allowed object.

b) *Assoc_Label_Value.* An ordered list of octet strings containing security attribute values negotiated for the security association. Whether this object contains a single label value or a range of values, it must be a subset of the Station_Security_Label_Values object.

c) *Assoc_Label_Explicit.* This Boolean variable indicates whether or not the security label negotiated for the association shall be carried on every PDU. If Assoc_Label_Explicit = FALSE, the negotiated label value shall apply implicitly to all PDUs protected by the security association.

2I.13 Additional Protected Header field

When present in the SDE Protected Header, the security label is the last option to appear in the field. Figure 2I.11 illustrates the six possible configurations of an SDE Protected Header containing a security label.

a) ID_pres = FALSE
Assoc_Frag_Enab = FALSE (or not defined)
Assoc_Label_Explicit = TRUE

b) ID_pres = TRUE
Assoc_Frag_Enab = FALSE (or not defined)
Assoc_Label_Explicit = TRUE

c) ID_pres = TRUE
Assoc_Frag_Enab = TRUE
Assoc_Label_Explicit = TRUE

d) ID_pres = TRUE
Assoc_Frag_Enab = TRUE
Assoc_Label_Explicit = TRUE

e) ID_pres = FALSE
Assoc_Frag_Enab = TRUE
Assoc_Label_Explicit = TRUE

f) ID_pres = FALSE
Assoc_Frag_Enab = TRUE
Assoc_Label_Explicit = TRUE

Figure 2I.11—Protected Header configurations

2I.14 Detailed functional specification

2I.14.1 SDE_UNITDATA.request

The following steps are performed as part of the formation of the Protected SDE Header (2.6.4.3) when the security association has both Remote_SDE and Assoc_Label_Explicit equal to TRUE. Processing continues as follows after the check of the ID_Pres Boolean.

a) Build the Security Label part of the Protected Header according to the registered specifications for the label set selected for the association and the label values in Assoc_Label_Value. (The definition of the label set, and the selection/calculation of label values from local information are accomplished by means outside the scope of this standard).

b) If Assoc_Frag_Enable equals TRUE, continue with the steps outlined in 2E.7.1.

NOTE—The addition of a label to an SDE PDU may cause the size of the PDU to exceed an implementation's limit. If Assoc_Frag_Enable equals FALSE, discard the PDU and report the error to local management. If Assoc_Frag_Enable equals TRUE, upon fragmentation the label shall be copied onto all fragments.

2I.14.2 SDE_UNITDATA.indication

The following steps are performed following the Station ID step (2.6.5.4) in the reception procedures if Remote_SDE equals TRUE and Assoc_Label_Explicit equals TRUE.

Check the security label and its values against the label set specified by the security association (Assoc_ Label_Set) and the allowed label values (or their bounds) (Assoc_Label_Value). If the check is successful, discard the label and continue PDU processing. If the check is successful, continue PDU processing.

If Assoc_Label_Explicit equals FALSE, no security label shall appear on any incoming PDU.

Annex 2J

(informative)

Security label set registration service

This standard relies on the availability of a registration service to assign Tag Set Names and serve as the repository of the semantics, special handling rules, and other details required for the implementation and use of security policy-specific label sets. One such service has been established by NIST. The Computer Security Objects Register (CSOR) is defined in NISTIR 5308: 1993, General Procedures for Registering Computer Security Objects.[24] The document contains generic and object-specific registration procedures for security labels and other security objects. The following are examples of the items whose value and significance shall be provided upon registration of a Named Tag Set:

— Number of tags
— Length of the set
— Length of each tag
— Ordering of tags
— Full format and semantics for Type 7 tags
— Security-relevant conditions

[24]This report may be obtained via the World Wide Web at the following URL: http://csrc.nist.gov/nistir/ir5308.txt. It may also be obtained by writing to the National Technical Information Service (NTIS), 5285 Port Royal Road, Springfield, VA 22161 USA, tel. (703) 605-6000.

Annex 2K

(informative)

Basic processing rules for security labels

2K.1 Introduction

This annex contains processing rules that support a common set of processing requirements for labeled data. The information provided here is consistent with processing rules in FIPS PUB 188: 1994, Standard Security Label for Information Transfer. These processing rules are offered as a basic set consistent with common practice in label processing at the time of publication; the set is thus expected to meet the requirements of many user communities.

2K.2 Trustworthiness of transmitted labels

Security labels used in communication systems are intended as an extension to end system labels. It is therefore necessary to ensure the integrity of the labels and their binding to the corresponding data units. This implies that implementations should support a mechanism for verifying the integrity of the labels and their binding to the labeled data.

2K.3 Originator requirements

a) Security label information shall be obtained from the Security Management Information Base (SMIB). The establishment of the reliability of the label information at the originating system and the translation of end system labels and attributes into the SDE label format are local matters.

b) The Named Tag Set negotiated for the security association shall be used on all SDE PDUs protected under that association. The label value on every outbound PDU must be within the range established for the security association. Notice that the range of values for the association may contain a single value.

c) An attempt to send data outside the value range established by the security association constitutes a security-relevant event and shall be reported to layer management. Implementations shall provide the option to log the event in an audit trail and to notify the SDE user of the error.

d) The system administrator shall be able to establish, at configuration time, thresholds, and parameters such as whether a type of security-relevant event is to be audited and whether notification shall be provided to the SDE user. This determination is a policy-based decision.

e) If all required security information cannot fit in the appropriate tags, the whole message shall be discarded and audited. Implementations shall provide for an optional error message to be passed to the SDE user.

f) The security label cannot fit in the corresponding protocol header, the whole message shall be discarded and audited. Implementations shall provide for an optional error message to be passed to the SDE user.

2K.4 Receiver requirements

a) Upon receipt, labels shall be parsed based on the semantic rules pointed to by the value in the Tag Set Name field of the label.

b) PDUs with missing labels, label errors (i.e., any part of the label fails to follow the SDE option format or the registered specifications), or out-of-bounds label values shall be discarded. Receipt of such PDUs may require audit. An optional error report PDU could be returned if allowed by the applicable security policy.

c) All PDUs shall contain at most one label. Detection of more than one label will cause an error. A PDU with multiple labels shall not be accepted.

d) Whether or not a label must be present on a PDU depends on the attributes for the security association under which the PDU is protected.

e) Implementations shall be able to log security-relevant events, such as label errors, in an audit trail and to return error report PDUs. The system administrator shall be able to establish, at configuration time, thresholds and parameters such as whether a type of security-relevant event is to be audited and whether to return an error report PDU. This determination is a policy-based decision.

2K.5 Error reports

The following set of possible events shall be reported to layer management upon occurrence:

a) Inbound violations.

   1) *Out-of-bounds label.* At least one security attribute value on a label is outside of the range of accepted values for the corresponding security association.

   2) *Bad label.* At least one error is found when parsing the incoming label. This includes the case when an integrity check fails.

   3) *Label missing.* No security label is found although one is required by the security association.

b) Outbound violations. Outbound messages shall meet the requirements in 2K.3. Label attributes for outbound messages shall be within the acceptable range for the security association that will be used for their protection. The verification, reporting, and handling of errors may be accomplished by means external to the SDE implementation; they are therefore local matters.

2K.6 Policy-based processing rules

The following policy-based processing rules are similar to those found in some networks run by the U.S. Department of Defense (DoD) and are consistent with DoD requirements. The Trusted Network Interpretation [2K-1][25] separates all system elements into *objects* and *subjects*. The data carried in a PDU is an object and anything that can send or receive objects (e.g., a host, an application) is a subject. Each object has a “sensitivity” label associated with it. Every subject is assigned a range of labels that it is authorized to send and a range of labels that it is authorized to receive. These labels have a hierarchical “sensitivity level” and a set of “sensitivity categories.” For a subject to have access to an object, the following criteria must be met:

a) The upper level of the subject’s “receive range” must be greater than or equal to the sensitivity level of the object.

[25]The numbers in brackets correspond to those of the bibliography in 2K.7.

b) The lower level of the subject's "receive range" must be less than or equal to the sensitivity level of the object.

c) The set of sensitivity categories for the subject must include all the sensitivity categories for the object.

Tag Type 1 supports this "restrictive" security policy and should be processed accordingly. Tag Types 2 and 5 may also support the same policy if specified explicitly and unambiguously through registration.

A complementary security policy based on release authorizations or "release markings" is also available. Under such policy, objects and subjects have a list of release categories. For a subject to have access to an object it must have at least one release category in common with the object. For instance, subjects acting on behalf of an organization's Personnel Department staff can have access to objects carrying a release marking for that department.

Tag Type 6 supports this "permissive" security policy and should be processed accordingly. Tag Types 2 and 5 may also support the same policy if specified explicitly and unambiguously through registration.

It is expected that most security label implementations will support these policies. Named Tag Sets that support either or both of these policies may be defined and registered. The specific ranges for an instance of communication should be established a priori when setting up a security association between the communicating ends. When tags supporting both policies appear on a label, the restrictive tag (supporting the sensitivity policy) is processed first. If that succeeds, then the permissive tag (supporting the release policy) is processed. In a restrictive security tag, the hierarchical component (sensitivity level) is processed first. Only if the sensitivity level is within the valid range for the security association are the sensitivity categories tested. In a permissive tag the security level field could carry a sensitivity level, as in the restrictive tags, or a null value. If both restrictive and permissive tags are carried on the same label, only the security level field in the restrictive tag shall contain significant information; the permissive tag must carry a null value in that field. The release markings may only be processed after all other tests are passed.

Security policies may require that one kind of error suppress reporting of others. For example, failure to meet the tests on restrictive attributes may disqualify the sender or recipient from receiving other kinds of diagnostics. Similarly, failure of the test on a security level may disqualify recipients from receiving diagnostics about other restrictive attributes.

2K.7 Bibliography

[2K-1] NCSC-TG-005 Version 1, Trusted Network Interpretation of the Department of Defense Trusted Computer System Evaluation Criteria, July 1987.

Annex 2L

(normative)

Secure Data Exchange (SDE) Protocol Implementation Conformance Statement (PICS) proforma[26]

2L.1 Introduction

The supplier of a protocol implementation that is claimed to conform to IEEE Standard 802.10-1998, Clause 2, Secure Data Exchange (SDE) shall complete the following protocol implementation conformance statement (PICS) proforma.

A completed PICS Proforma is the PICS for the implementation. The PICS is a statement identifying the capabilities and options of the protocol that have been implemented. The PICS can serve a number of purposes, including its use as

a) A checklist for the protocol implementor to reduce the risk of failure to conform to the standard through oversight.

b) A detailed indication of the protocol's capabilities for the supplier and receiver of the implementation, stated in terms relative to the common basis provided by the standard PICS proforma.

c) A basis for the user of the implementation to check the possibility of interworking with another implementation.

d) The basis for a protocol tester to select appropriate tests against which to assess the claim for conformance to the implementation.

Implementors of SDE are expected to provide products that, at least, are capable of meeting the minimum essential requirements (MERs), and the SDE protocol data unit (PDU) mandatory fields and parameters as specified in the SDE standard.

Implementations are also expected to provide the services and service primitives used by SDE. SDE uses only two service primitives at its boundaries, UNITDATA.request and UNITDATA.indication, as defined in ISO/IEC 15802-1: 1995. These two primitives are prefixed as follows:

— At the upper boundary of SDE with "SDE" as SDE_UNITDATA.request and SDE_UNITDATA.indication.
— At the lower or MAC sublayer boundary with "MA" as MA_UNITDATA.request and MA_UNITDATA.indication.

The services provided at the upper SDE boundary include those provided by the MAC sublayer with the addition of those services provided transparently by the SDE sublayer.

[26]*Copyright release for PICS Proformas:* Users of this standard may freely reproduce the PICS proforma in this annex so that it can be used for its intended purpose and may further publish the completed PICS.

2L.2 Abbreviations and special symbols

M	Mandatory
O	Optional
O.1	Optional, but support of at least one of the group of options labeled in this way is required
<item>	Conditional-item symbol, dependent upon the support marked for <item>: (see 2L.3.4)

2L.3 Instructions for completing the PICS proforma

2L.3.1 General structure

The tables in 2L.4 through 2L.7 of this SDE PICS proforma provide a fixed format for providing the essential information to complete an SDE PICS. SDE implementors are expected to complete the tables in 2L.4 through 2L.7.

2L.3.2 Additional information

Items of Additional Information allow a supplier to provide further information intended to assist the interpretation of the SDE PICS. It is not intended or expected that a large quantity of information will be supplied, and the SDE PICS can be considered complete without any such information. An example might be an outline of the ways in which a single implementation can be set up to operate in a variety of environments and configurations.

References to items of Additional Information may be indicated by providing an index, in the form (Ai), to additional parts of the PICS next to answers to questions, or in the constraints fields. There are no other restrictions on the format or presentation of the Additional Information.

2L.3.3 Exception information

Occasionally, it may be necessary for a supplier to enter an answer to an item, which normally requires an answer of “Yes” (since it is mandatory), in such a way that conflicts with the indicated requirement. A possible reason might be that a defect in the standard has been reported, a correction for which is expected to change the requirement not met by the implementation. References to items of Exception Information may be indicated by providing an index, in the form (Xi), to additional parts of the PICS next to the answer in part 2L.5, or in the constraints fields. There are no other restrictions on the format or presentation of the Exception Information.

2L.3.4 Conditional items

The SDE PICS proforma contains a number of conditional items. These are items for which the applicability of the item itself and its status, if it does apply (mandatory or optional), are dependent upon whether or not other items are supported.

Individual conditional items are indicated by a conditional symbol of the form “<item>:S” where <item> is an item reference that appears in the first column of the table for some other item, and S is one of the status symbols M or O. If the item referred to by the conditional symbol is marked as supported or implemented, the conditional item is applicable and its status is given by S. For example, if CLEAR HEADER is implemented, then SDE Designator is mandatory (ch:M).

2L.3.5 Specific instructions

The table in 2L.4, Identification, is to be completed as indicated with the information necessary to identify fully both the supplier and the implementation. Notes are provided giving additional instructions.

The table in 2L.5, Amendments, corrigenda, and exceptions, provides the format for identifying the specific standard and any amendments or corrections to the standard and the SDE PICS proforma that were completed. The implementor must declare any exceptions (see 2L.3.3) to the standard and provide the date of preparation of the PICS.

Items in the table in 2L.6, SDE feature declarations, are organized to indicate which features have been implemented by a Yes or No answer. The implementor must support at least one of the items marked “O.1”.

Items in the table in 2L.7, SDE PDU declarations, are organized by PDU fields and subfields in sequential order. Each PDU field and its affiliated subfields are separately indicated as Transmit (sender) or Receive (receiver) under which the field/subfield is defined in the SDE standard. Entries listed under the STD columns in the table are designated as optional (O) or mandatory (M). The SDE DATA field is the only mandatory field. Each field/subfield is followed by a constraints column. The constraints column refers to the applicable IEEE Std 802.10-1998, Clause 2 subclause or annex, and/or contains the number of octets for each field/subfield. The IMP fields are used by the implementor to indicate, by entering an “X” in the field, that a PDU item has been implemented. The implementor must observe the conditional item indications in the STD columns.

2L.4 Identification

Supplier	
Point of contact for queries about the PICS	
Implementation Name(s) and Version(s)	
Other information necessary for full identification [e.g., name(s) and version(s) for machine(s), and/or operating systems; System Name(s)]	
NOTE 1—Only the first three items are required for all implementations; the other information may be completed as appropriate in meeting the requirements for full identification. NOTE 2—The terms Name and Version should be interpreted appropriately to correspond with a supplier’s terminology (e.g., Type, Series, Model).	

2L.5 Amendments, corrigenda, and exceptions

Identification of protocol specification	IEEE Std 802.10-1998, Clause 2, Annexes 2A–2K
Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS	IEEE Std 802.10-1998, Clause 2, Annex 2L Amd: Corr: Amd: Corr:
Have any exception items been required? (The answer Yes means that the implementation does not conform.)	No [ ] Yes [ ]
Date of statement	

2L.6 SDE feature declarations

Item	Feature	Status	Support	Constraints
	Does this implementation provide transparency?	M	Yes [ ] No [ ]	ref: 2.2.2, 2.3
	Does this implementation provide correct error handling of erroneous incoming PDUs?	M	Yes [ ] No [ ]	ref: 2.3
	Does this implementation provide pass through of incoming PDUs with a clear header but Remote_SDE is False?	M	Yes [ ] No [ ]	ref: 2.6.5.1.1
	SAID Reception	M[a]	Yes [ ] No [ ]	ref: 2.7.3, 2.6.3.4
	Security Services	M	Yes [ ] No [ ]	ref: 2.3
	Data Confidentiality	O.1	Yes [ ] No [ ]	ref: 2.3, 2.7.4
ci	Connectionless Integrity	O.1	Yes [ ] No [ ]	ref: 2.3, 2.7.4
	Data Origin Authentication	O	Yes [ ] No [ ]	ref: 2.3
ac	Access Control (labeling)	O	Yes [ ] No [ ]	ref: 2.3, Annex 2I
	Tag Type 1	ac:O.1	Yes [ ] No [ ]	
	Tag Type 2	ac:O.1	Yes [ ] No [ ]	
	Tag Type 5	ac:O.1	Yes [ ] No [ ]	
	Tag Type 6	ac:O.1	Yes [ ] No [ ]	
	Tag Type 7	ac:O.1	Yes [ ] No [ ]	
	SDE on Ethernet V2.0	O	Yes [ ] No [ ]	ref: Annex 2H
	Fragmentation	O	Yes [ ] No [ ]	ref: Annex 2E

[a]Affects SAID in table in 2L.7.

2L.7 SDE PDU declarations

Item	SDE PDU fields	Transmit		Receive		Constraints
		STD	IMP	STD	IMP	
ch	CLEAR HEADER[a]	O		M		≤ 27 octets—ref: 2.5.2.1
	SDE Designator	ch:M		ch:M		3 octets—ref: 2.5.2.1.1
	SDE Reserved LSAP	ch:M		ch:M		"$0A0A_{16}$"
	Unnumbered Information control field	ch:M		ch:M		"03_{16}"
	SAID	ch:M		M[b]		4 octets—ref: 2.5.2.1.2
	Management Defined Field	ch:O		ch:O		≤ 20 octets—ref: 2.5.2.1.3
ph	PROTECTED HEADER	O		O		8, 9, 13, 16 or more, but ≤ 260 octets—ref: 2.5.2.2
	Station Identifier	ph:O		ph:O		8 octets—ref: 2.3, 2.5.2.2
	Flags	ph:O		ph:O		1 octet—ref: Annex 2E
	Fragment Identifier	ph:O		ph:O		4 octets—ref: Annex 2E
sl	Security Label	ph:O		ph:O		3 to 247 octets ref: 2.3, Annexes 2I, 2J, 2K
	Security Label Length	sl:M		sl:M		1 octet—ref: 2I.4
	Security Tags	sl:M		sl:M		ref: Annex 2I
	Tag Type	sl:M		sl:M		1 octet—ref: 2I.5.1
	Tag Length	sl:M		sl:M		1 octet—ref: 2I.5.2
	Security Data	sl:M		sl:M		≤ 244 octets—ref: 2I.5.3
	DATA (SDE SDU)	M		M		≥ 1 octet—ref: 2.5.2.3, 2.5.3
pd	PAD	O		O		ref: 2.5.2.4, 2.5.3
	Pad Field	pd:M		pd:M		≤ 255 octets— ref: 2.5.2.4.1, 2.5.3
	Pad Length Field	pd:M		pd:M		1 octet—ref: 2.5.2.4.2, 2.5.3
	INTEGRITY CHECK VALUE (ICV)	O		O		≥ 1 octet—ref: 2.3, 2.5.2.5, 2.5.3

[a]Some bridges operate on data outside the MAC header. As a result, SDE frames that do not include the SDE Designator may not be processed correctly. Therefore, when bridges of this type are employed, the option for Clear Header should be selected.
[b]From table in 2L.6 SAID reception is mandatory.