

IEEE Standards for Local and Metropolitan Area Networks:

Supplement to Standard for Interoperable LAN/MAN Security (SILS)—

Key Management (Clause 3)

Sponsor

**LAN MAN Standards Committee
of the
IEEE Computer Society**

Approved 21 January 1998

IEEE Standards Board

Abstract: A cryptographic key management model and a key management OSI Basic Reference Model Application Layer protocol are specified.

Keywords: association control service element (ACSE), asymmetric cryptographic algorithm, center-based key distribution, certificate-based key distribution, cryptographic keying material, key management model, key management protocol (KMP), manual key distribution, symmetric cryptographic algorithm

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

Copyright © 1998 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 1998. Printed in the United States of America.

ISBN 1-55937-955-3

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE 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 Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

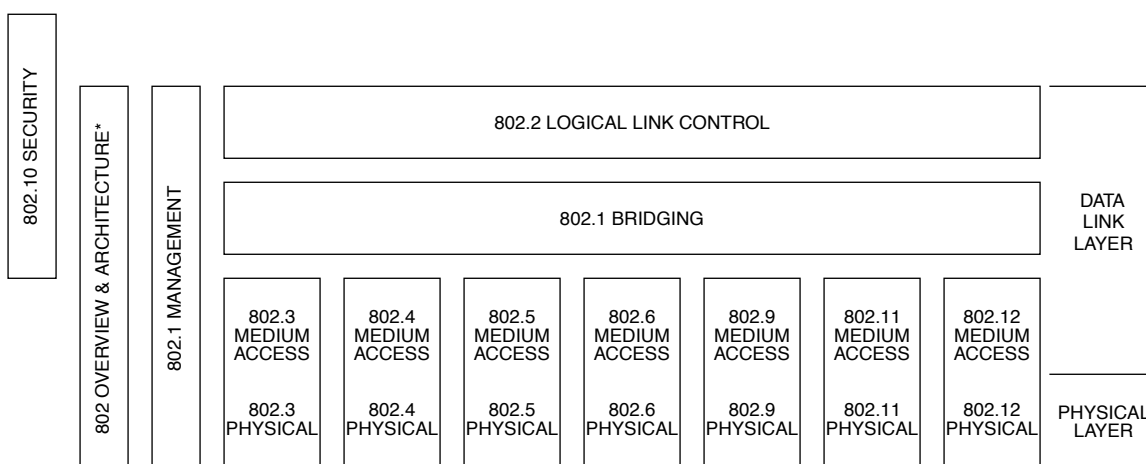
Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not 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.

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; (508) 750-8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Introduction

[This introduction is not part of IEEE Std 802.10c-1998, IEEE Standards for Local and Metropolitan Area Networks: Supplement to Standard for Interoperable LAN/MAN Security (SILS)—Key Management (Clause 3).]

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 access technologies 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 10038] *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 802.1G [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*
- 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 DIS 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 802.10c-1998

The IEEE 802.10 Working Group was formed in May 1988 to address the security of Local Area and Metropolitan Area Networks (LANs and MANs). Work on cryptographic key management began in May 1989. The IEEE 802.10 Working Group is sponsored by the LAN MAN Standards Committee. IEEE 802.10 provides interoperability standards that are compatible with existing IEEE 802 standards and Open Systems Interconnection (OSI) architectures. The working group is made up of representatives from the vendor, government, and user communities.

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

As standards evolve, several key areas will become critically important. One of these areas is network security. LANs and MANs must have the capability to exchange data in a secure manner. This is especially important in cases where disclosure of operational data to unauthorized parties would severely undermine an organization's effectiveness. A related critical area is data reliability. Should data become corrupted, either accidentally or maliciously, the effect on the organization could be grave. Both financial and government institutions have traditionally been aware of the importance of reliable data in a secure environment. However, recent widely publicized cases of computer fraud and related crimes have made this capability a goal for many other industries as well. Security has become a universal concern.

As the need for security on LANs and MANs becomes more widely recognized, the need for a security standards also becomes a priority. Work on security standards has already started; and where applicable, this standard incorporates this previous work.

This standard contains state-of-the-art material. The area covered by this standard is undergoing evolution. Revisions are anticipated within the next few years to clarify existing material, to correct possible errors, and to incorporate new related material. Information on the current revision state of this and other IEEE 802 standards may be obtained from

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

IEEE 802 committee working documents are available from

IEEE Document Distribution Service
AlphaGraphics #35 Attn: P. Thrush
10201 N. 35th Avenue
Phoenix, AZ 85051
USA

Participants

The following is a list of participants in the Standard for Interoperable LAN/MAN Security effort of the IEEE Project 802 Working Group. Voting members at the time of publication are marked with an asterisk (*).

Kenneth G. Alonge,* *Chair*
Joseph G. Maley,* *Recording Secretary*

Russell D. Housley,* *Vice Chair and Editor*
Richard McAllister,* *Executive Secretary*

Asher Altman
Kurt Augustine
Kirk Barker
Kym D. Blair
James Coyle
Dane Elliot
Joanne Evans
Warwick Ford
Kim Kirkpatrick
Robert D. Kolacki
Paul Lambert

Wen-Pai Lu
Massimo Mascoli
Michael Michnikov
Ken McCoy
Tassos Nakassis
Noel A. Nazario
Barbara Patrick
Brian Phillips
James Randall
Karen Randall
Amy Reiss
Robert Rosenthal

Anthony Sacco
Brian P. Schanning
Emile W. Soueid
Jeffrey Vignes
Dale L. Walters
David Wheeler
Michael White
Joseph B. Williamson
Peter Yee
Roberto Zamparo
James E. Zmuda

The following persons were on the balloting committee:

William B. Adams	Stephen Barton Kruger	Donal O'Mahony
Kenneth G. Alonge	Kenneth C. Kung	Roger Pandanda
Alan Arndt	Lanse M. Leach	Vahid Parsi
Kit Athul	Randolph S. Little	Lucy W. Person
Greg Bergren	Joseph C. J. Loo	Andreas Pfitzmann
James T. Carlo	Robert D. Love	Thomas L. Phinney
Michael H. Coden	Wen-Pai Lu	Vikram Punj
Robert S. Crowder	Joseph G. Maley	Karen T. Randall
Peter Ecclesine	Richard K. McAllister	Edouard Y. Rocher
Gregory Elkmann	Darrell B. McIndoe	James W. Romlein
James T. Ellis	Milan Merhar	Floyd E. Ross
John E. Emrich	Steve Messenger	Christoph Ruland
Philip H. Enslow	Bennett Meyer	Brian P. Schanning
Changxin Fan	Colin K. Mick	Roger R. Schell
John W. Fendrich	Ann Miller	Norman Schneidewind
Michael A. Fischer	Dale W. Miller	Donald A. Sheppard
Sandra J. Forney	David S. Millman	Dan Shia
Harvey A. Freeman	Refik Molva	Alex Soceanu
Robert J. Gagliano	Warren Monroe	Fred J. Strauss
Gautam Garai	John E. Montague	Michael L. Sutherland
Eli Herscovitz	Kinji Mori	Efstathios D. Sykas
Russell D. Housley	James R. Moulton	Geoffrey O. Thompson
Henry Hoyt	Wayne D. Moyers	Brian Tretick
Gilbert J. Huey	Michael John Nash	Mark-Rene Uchida
Thomas R. Hunwick	Noel Nazario	Dale Walters
Richard J. Iliff	Dan Nessett	Frank J. Weisser
Cynthia E. Irvine	Paul Nikolich	Clark Weissman
Peter M. Kelly	Richard O'Brien	Raymond P. Wenig
Gary C. Kessler	Robert O'Hara	Qian-Li Yang
Yongbum Kim		Oren Yuen

The final conditions for approval of this supplement were met on 21 January 1998. This supplement was conditionally approved by the IEEE Standards Board on 9 December 1997, with the following membership:

Donald C. Loughry, *Chair*

Richard J. Holleman, *Vice Chair*

Andrew G. Salem, *Secretary*

Clyde R. Camp	Lowell Johnson	Louis-François Pau
Stephen L. Diamond	Robert Kennelly	Gerald H. Peterson
Harold E. Epstein	E. G. "Al" Kiener	John W. Pope
Donald C. Fleckenstein	Joseph L. Koepfinger*	Jose R. Ramos
Jay Forster*	Stephen R. Lambert	Ronald H. Reimer
Thomas F. Garrity	Lawrence V. McCall	Ingo Rüschi
Donald N. Heirman	L. Bruce McClung	John S. Ryan
Jim Isaak	Marco W. Migliaro	Chee Kiow Tan
Ben C. Johnson		Howard L. Wolfman

*Member Emeritus

Also included are the following nonvoting IEEE Standards Board liaisons:

Satish K. Aggarwal
Alan H. Cookson

Valerie E. Zelenty
IEEE Standards Project Editor

Contents

<i>Changes and additions to IEEE Std 802.10-1992</i>	1
1.2.1 Acronyms.....	1
1.3 References.....	2
3. Key management	4
3.1 Overview.....	4
3.2 Key distribution techniques	5
3.3 Key management model	9
3.4 Service definition	18
3.5 Security exchanges.....	41
3.6 KMAE control function	56
Annex 3A (normative) Locating SDE key management entities	79
Annex 3B (informative) Certificate replacement.....	82
Annex 3C (informative) Compromised material lists	85
Annex 3D (informative) Key distribution scenarios	88
Annex 3E (normative) SDE attribute negotiations	94

IEEE Standards for Local and Metropolitan Area Networks:

Supplement to Standard for Interoperable LAN/MAN Security (SILS)—

Key Management (Clause 3)

Revisions to IEEE Std 802.10-1992

The contents of this document will be incorporated into IEEE Std 802.10 in a future edition. The clauses of this document are ordered to parallel the order of clauses in the base standard. This supplement is intended to be used in conjunction with IEEE Std 802.10-1992. Editing instructions necessary to incorporate this supplement into IEEE Std 802.10 are provided in *bold italics*.

1.2.1 Acronyms

Add the following to 1.2.1 of IEEE Std 802.10-1992:

ACSE	Association Control Service Element
ASE	Application Service Element
ASO	Application Service Object
CA	Certification Authority
CCITT	International Telegraph Telephone Consultative Committee (Renamed ITU-T)
CF	Control Function
CML	Compromised Material List
CRL	Certificate Revocation List
GULS	Generic Upper Layers Security
ICV	Integrity Check Value
IEC	International Electrotechnical Commission

ISO	International Organization for Standardization
ITU-T	International Telecommunications Union - Technical
KCASO	Key Center Application Service Object
KDC	Key Distribution Center
KEK	Key Encryption Key
KMAE	Key Management Application Entity
KMAP	Key Management Application Process
KMID	Keying Material IDentifier
KMP	Key Management Protocol
KMTI	Key Management Technique Identifier
KPASO	Key Peer Application Service Object
KTC	Key Translation Center
MKC	Multicast Key Center
NLSP	Network Layer Security Protocol
PDU	Protocol Data Unit
SA	Security Association
SAID	Security Association IDentifier
SDE	Secure Data Exchange
SE	Security Exchange
SEI	Security Exchange Item
SESE	Security Exchange Service Element
SMIB	Security Management Information Base
ST	Security Transformation
TLSP	Transport Layer Security Protocol

1.3 References

Add the following references to IEEE Std 802.10-1992:

ANSI X9.30-1 (1995), Public Key Cryptography Using Irreversible Algorithms for the Financial Services Industry—Part 1: The Digital Signature Algorithm (DSA).¹

ANSI X9.30-2 (1993), Public Key Cryptography Using Irreversible Algorithms for the Financial Services Industry—Part 2: The Secure Hash Algorithm (SHA).

ANSI DRAFT X9.42, Public Key Cryptography for the Financial Services Industry: Establishment of Symmetric Algorithm Keys Using Diffie-Hellman (draft dated 12 March 1998).

ANSI DRAFT X9.44, Public Key Cryptography Using Irreversible Algorithms for the Financial Services Industry: Transport of Symmetric Algorithm Keys Using RSA (draft dated September 1997).

ISO 7498-2: 1989, Information processing systems—Open Systems Interconnection—Basic Reference Model—Part 2: Security Architecture.² (ITU-T Recommendation X.800)

¹ANSI publications are available from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA.

²ISO and ISO/IEC publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse. ISO publications are also available in the United States from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA.

ISO 7498-3: 1997, Information technology—Open Systems Interconnection—Basic Reference Model—Part 3: Naming and Addressing.

ISO 8649: 1996, Information technology—Open Systems Interconnection—Service definition for the Association Control Service Element. (ITU-T Recommendation X.217)

ISO 8650-1: 1996, Information technology—Open Systems Interconnection—Connection-oriented protocol for the Association Control Service Element: Protocol specification.

ISO/IEC 7498-1: 1994, Information technology—Open Systems Interconnection—Basic Reference Model: The Basic Model. (ITU-T Recommendation X.200)

ISO/IEC 8824: 1990, Information technology—Open Systems Interconnection—Specification of Abstract Syntax Notation One (ASN.1). (ITU-T Recommendation X.208)

ISO/IEC 9545: 1994, Information technology—Open Systems Interconnection—Application Layer structure. (ITU-T Recommendation X.207)

ISO/IEC 9594-8: 1995, Information technology—Open Systems Interconnection—The Directory—Part 8: Authentication framework. (ITU-T Recommendation X.509)

ISO/IEC 10181-1: 1996, Information technology—Open Systems Interconnection—Security frameworks for open systems: Overview. (ITU-T Recommendation X.810)

ISO/IEC 10736: 1995, Information technology—Telecommunication and information exchange between systems—Transport layer security protocol. (ITU-T Recommendation X.274)

ISO/IEC 10745: 1995, Information technology—Open Systems Interconnection—Upper layers security model. (ITU-T Recommendation X.803)

ISO/IEC 11577: 1995, Information technology—Open Systems Interconnection—Network layer security protocol. (ITU-T Recommendation X.273)

ISO/IEC 11586-1: 1996, Information technology—Open Systems Interconnection—Generic upper layers security: Overview, models and notation. (ITU-T Recommendation X.830)

ISO/IEC 11586-2: 1996, Information technology—Open Systems Interconnection—Generic upper layers security: Security Exchange Service Element (SESE) service definition. (ITU-T Recommendation X.831)

ISO/IEC 11586-3: 1996, Information technology—Open Systems Interconnection—Generic upper layers security: Security Exchange Service Element (SESE) protocol specification. (ITU-T Recommendation X.832)

ISO/IEC 11586-4: 1996, Information technology—Open Systems Interconnection—Generic upper layers security: Protecting transfer syntax specification. (ITU-T Recommendation X.833)

3. Key management

Replace Clause 3 and add the following subclauses to IEEE Std 802.10-1992:

3.1 Overview

3.1.1 Scope and purpose

This clause specifies a cryptographic key management model and protocol. The Key Management Protocol (KMP) is an OSI Basic Reference Model Application Layer protocol. KMP uses the services provided by the Security Exchange Service Element (SESE), the Association Control Service Element (ACSE), and the Presentation service. In order to retrieve presentation addresses or to support a specific key management scheme, the Directory service may be needed. A KMP entity provides a service within the context of a defined security policy; it is outside the scope of this standard to define such security policies.

The KMP entity supports security protocol entities by managing the cryptographic keying material and security associations that are necessary for the provision of secure communication. KMP entities are responsible for creating the cooperative relationship and establishing the required cryptographic keying material. Security protocols supported by this KMP include the Secure Data Exchange security protocol, IEEE 802.10b; the Network Layer Security Protocol, ISO/IEC 11577; and the Transport Layer Security Protocol, ISO/IEC 10736. This KMP is intended to support other lower layer and upper layer security protocols, including those still under development by the Internet Engineering Task Force to provide security for the Internet Protocol (IP).

The key management model and protocol support three key distribution techniques: manual key distribution, center-based key distribution, and certificate-based key distribution. Symmetric cryptographic algorithms are normally used with center-based and manual key distribution techniques. Asymmetric cryptographic algorithms are always used with certificate-based key distribution techniques.

3.1.2 General

The KMP is used to create, spawn, and delete the security associations that are needed by security protocols as required by security policy. The security association is composed of symmetric keying material and security attributes.

Symmetric keying material is used in symmetric cryptographic algorithms. If two security protocol entities use the same symmetric algorithm, one security protocol entity uses the same cryptographic key to decrypt some or all of a protocol data unit (PDU) used by the other security protocol entity to encrypt the plaintext data.

The security attributes determine how the security protocol will use the symmetric keying material. For example, the security attributes tell which encryption algorithm and mode are used to provide confidentiality. Other security attributes tell which optional security protocol fields are present and which ones are absent.

This standard uses the functions and services defined in Generic Upper Layers Security (GULS), ISO/IEC 11586: 1996,³ to provide an application-level communications infrastructure for the development of this mechanism-independent KMP.

GULS defines a Security Exchange (SE) mechanism that permits the exchange of security information, such as keying material, in support of the various security services. The SE is defined as a sequence of Security

³Information on references can be found in 1.3.

Exchange Items (SEIs). The Security Exchange Service Element (SESE), an application level element, uses ASN.1 notational tools to allow the consistent specification of SEs. GULS also defines security transformations (STs) to provide integrity, confidentiality, or both, of application data communicated between open systems. STs consist of a transfer-syntax function and ASN.1 notational tools. A transfer-syntax function, the Generic Protecting Transfer Syntax, allows applications to signal their integrity and confidentiality requirements in a communication protocol. ASN.1 notational tools allow the consistent specification of STs, as well as a mechanism allowing an application to map its integrity and confidentiality requirements to specific security transformations that satisfy these requirements.

This standard uses the functions defined in GULS to define SEs and STs that are applicable to the key distribution techniques discussed in subclause 3.2 (of this standard). This standard also defines specific uses of the SESE, and application-contexts that define specific procedural rules and constraints pertaining to the overall use of the GULS functions in support of key management.

3.1.3 Security definitions

3.1.3.1 keying material: Keys, codes, authentication material, or state information necessary to establish and maintain cryptographic relationships.

3.1.3.2 nonce: A unique identifier. Examples of a nonce are a random number, a timestamp, or a counter, each of which is used only once.

3.1.3.3 pairwise: Something shared only by two parties.

3.1.3.4 rekey: Replacement of keying material by manual or electronic means. In certificate-based key management, rekey is the issuing of a replacement certificate, and may include the replacement of the asymmetric key pair.

3.1.3.5 spawn: Given one security association, to create another uniquely named security association by changing the keying material, the security attributes, or both.

3.1.3.6 update: A one-way transformation of a symmetric cryptographic key to form another symmetric cryptographic key.

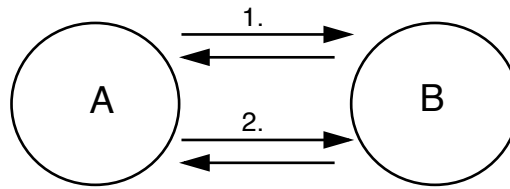
3.2 Key distribution techniques

This subclause describes the key distribution techniques that are supported by this protocol. Three classes of key distribution techniques are supported: manual distribution, center-based key distribution, and certificate-based distribution. The key center distribution techniques support both Key Distribution Centers (KDCs) and Key Translation Centers (KTCs). KDCs are sometimes called Centers for Key Distribution (CKDs); they are the same. KTCs are sometimes called Centers for Key Translation (CKTs); they are the same. The format of certificates used in the certificate-based distribution techniques is described in ITU-T Recommendation X.509.

3.2.1 Manual key distribution techniques

Manual key distribution techniques may be used to establish pairwise or multicast cryptographic keying material. For this technique, cryptographic keying material is generated a-priori and stored on the peer key management systems via means other than KMP. The cryptographic keying material is stored in a cache of memory and accessed using Keying Material Identifiers (KMIDs). See Figure 3-1. The cache is usually

located in the Security Management Information Base (SMIB). The keying material cache with KMIDs is analogous to an array with indices. KMIDs are updated in the cache via means other than KMP.



1. A and B select a key that was previously distributed manually.
2. A and B use the key to protect the negotiation of the attributes for the operation of the security protocol.

Figure 3-1—Selecting a manually distributed key

Manual methods of key distribution can be cumbersome and have scalability issues. These methods use off-line delivery to intended users of keying material that must be kept confidential. KMP allows the use of pre-placed keying material or keying material generated by transforming preplaced keying material in the establishment and maintenance of security associations. The manual methods for preplacement of this keying material are beyond the scope of KMP.

In many cases, manual delivery of keying material is required only once per user. Distribution of additional keying material can be performed using the manually distributed key as a key encryption key (KEK) to encrypt the additional keying material. The encrypted keying material can then be distributed using any convenient method.

Manual key distribution methods do not provide any authentication other than that provided by the delivery method. Therefore, the strength of the procedures used for distribution of the keying material is extremely important.

Manual key distribution is suitable for multicast keying material distribution. In fact, it is often the most efficient way to distribute group or network-wide keying material, especially to large groups.

3.2.2 Center-based key distribution techniques

Center-based key distribution techniques may be used to establish pairwise or multicast cryptographic keying material between the participating parties via a trusted third party: the Key Distribution Center (KDC) or Key Translation Center (KTC).

Keying material established using a technique derived from the scheme originally described by Needham and Schroeder can be based on either a KDC or a KTC. The KDC and KTC protocols depend upon the manual distribution of KEKs to provide confidentiality and integrity protection for keying material. Dual-control or split-knowledge techniques may be used for initial secure distribution of KEKs, where two key components that will later be combined to form the cryptographic key are sent by separate paths. Each of the components conveys no knowledge of the final key. Once the manual KEKs have been distributed, cryptographic keying material may be electronically distributed. Confidentiality for the automatically distributed keys is provided by encryption using the manually distributed KEKs. Integrity may also be provided by the presence of an Integrity Check Value (ICV) or a digital signature.

Each party has a manually distributed secret KEK that is known only to itself and the KDC. One party, usually the initiator, requests the KDC to generate keying material to be shared by Party A and Party B. The

KDC generates the keying material and separately encrypts it in the manually distributed secret KEK for each party. The encrypted keying material is sent to the requesting party. The requesting party forwards the keying material to the other party encrypted in the other party's manually distributed secret KEK. As part of this transfer, the two parties demonstrate that they have received the same key from the KDC. See Figure 3-2.

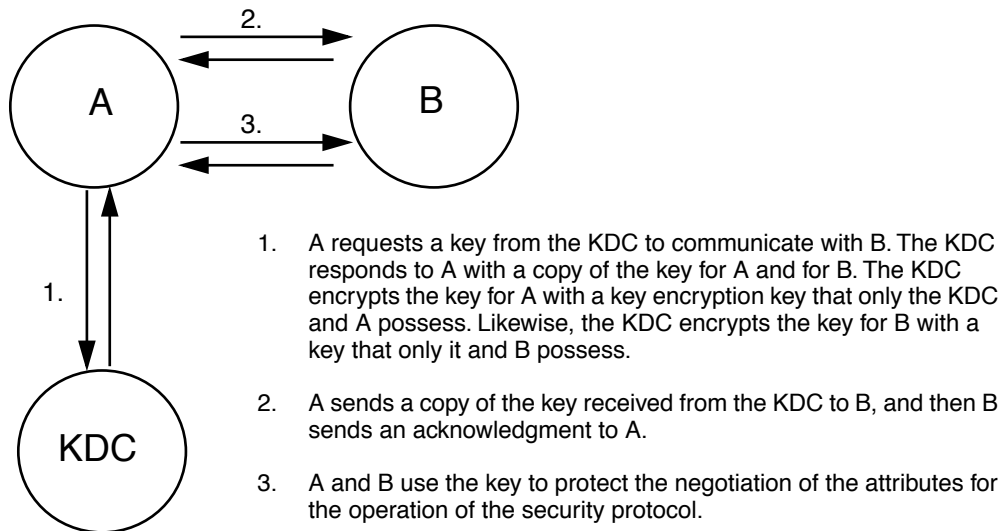


Figure 3-2—Center-based key distribution

The keying material that is generated by the KDC can be used directly by a security protocol, or it can be used as a KEK. In the latter case, the KDC-generated KEK is used to protect keying material generated by one party while it is transferred to the other party. When this scheme is adopted, the three tiers of keying material are used as follows:

Tier 1: *Manually distributed KEK.* Pairwise between one party and the KDC.

Tier 2: *Automatically generated KEK.* Generated by the KDC and shared by Party A and Party B.

Tier 3: *Keying material used by the security protocol.* Generated by either Party A or Party B, and then sent to the other party encrypted in the automatically generated KEK (Tier 2).

Unlike KDCs, KTCs do not generate keying material. With KTCs, each party has a manually distributed secret KEK that is known only to itself and the KTC. Party A generates the keying material, encrypts it in the manually distributed secret KEK shared with the KTC, and sends the encrypted keying material to the KTC. The KTC decrypts the keying material, and then the KTC translates it by reencrypting it in the manually distributed secret KEK shared with Party B. The translated key is sent back to the requestor, Party A. Party A forwards the translated keying material to Party B. As part of this transfer, the two parties demonstrate that they have the same key. See Figure 3-3.

In cases where the communications topology prohibits Party A from communicating directly with the KDC or KTC, Party B can communicate with the KDC or KTC and send the encrypted keying material to Party A.

3.2.3 Certificate-based distribution techniques

Certificate-based key distribution techniques may be used to establish pairwise cryptographic keying material. This standard does not support the use of certificate-based key distribution techniques to establish multicast cryptographic keying material; however, once pairwise cryptographic keying material is established, it can be used to protect the distribution of multicast cryptographic keying material. See Figures 3-4 and 3-5.

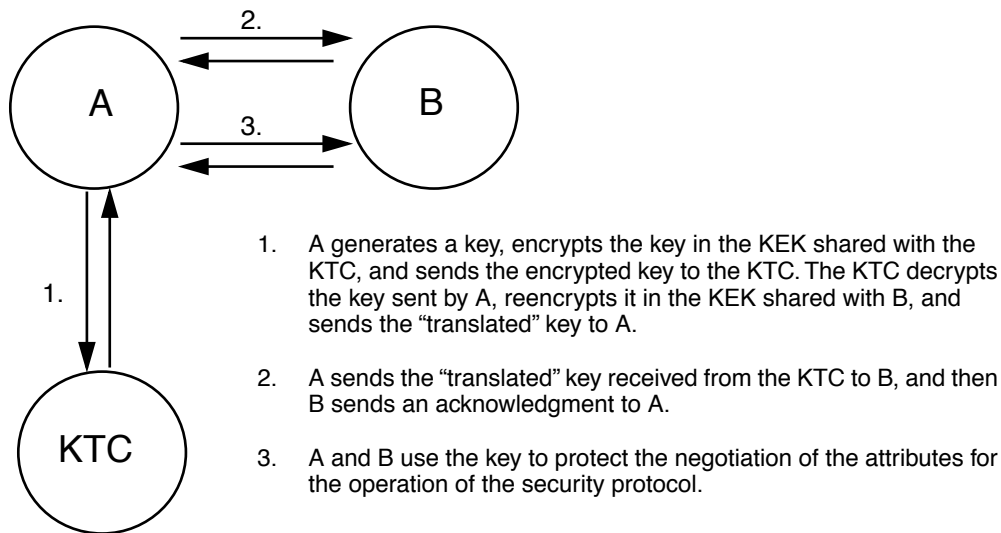


Figure 3-3—Center-based key translation

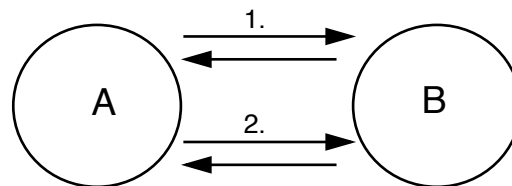


Figure 3-4—Certificate-based key distribution

There are two certificate-based distribution techniques:

- A public key cryptographic technique that uses asymmetric cryptography to encrypt locally generated keying material to protect it while it is being sent to the remote key management end system. This is commonly called *key transfer* or *key transportation*.
- A public key cryptographic technique that cooperatively generates common symmetric cryptographic keying material at both the local and remote key management end system. This is commonly called *key exchange*; it is also called *key agreement*.

This standard uses the formats for and management of certificates defined in the Directory Authentication Framework, ITU-T Recommendation X.509 (ISO/IEC 9594-8: 1995). The X.509 certificate contains the public component of the asymmetric key pair. Of course, the private component of the asymmetric key pair must be kept confidential. The certificate also contains the identity of the certificate owner; this identity takes the form of a distinguished name. Certificates are digitally signed by a recognized certification authority (CA). By signing the certificate, the CA is binding the public component of the asymmetric key pair and the identity. That is, the entity named by the distinguished name must hold the private component of the asymmetric key pair.

Each party must have a trusted public key. Generation and distribution of this trusted public key is beyond the scope of this standard. A party validates a certificate, in part, by validating that the current date is within the validity period of the certificate and by validating the certificate's signature using the public key of the CA. Likewise, the party validates the CA's certificate, which contains its public key. This process continues until the party validates a certificate using the trusted public key. This list of certificates composes a certification path.

CAs may revoke a certificate if the binding between the public component of the asymmetric key pair and the identity is no longer appropriate or if the private component of the asymmetric key pair is disclosed. As part of the certificate validation process, parties must ensure that a certificate has not been revoked by contacting the CA or by checking a recent Certificate Revocation List (CRL) issued and signed by the CA.

When certificates are cached, their validity should be periodically confirmed to ensure that they have not expired or been revoked. If after checking the validity of a certificate it is found to be revoked, then the certificate should be discarded from the cache and security associations that were established using the revoked certificate should be deleted.

3.2.4 Multicast key distribution techniques

Multicast key distribution relies on one of the key distribution techniques, manual, center-based, or certificate-based, to establish pairwise cryptographic keying material between the local key management end system and the Multicast Key Center (MKC). Using the pairwise cryptographic material, the local key management system requests the broadcast or multicast keying material and its associated attributes from the MKC. The MKC determines if the local key management system is authorized to receive the requested multicast keying material; if it is authorized, the MKC sends the multicast keying material and its associated attributes to the local key management system.

Each security protocol entity instigates obtaining the broadcast or multicast keying material as its needs dictate. The MKC does not push the broadcast or multicast keying material to every member of the group. Figure 3-5 depicts multicast key distribution using the three key distribution techniques.

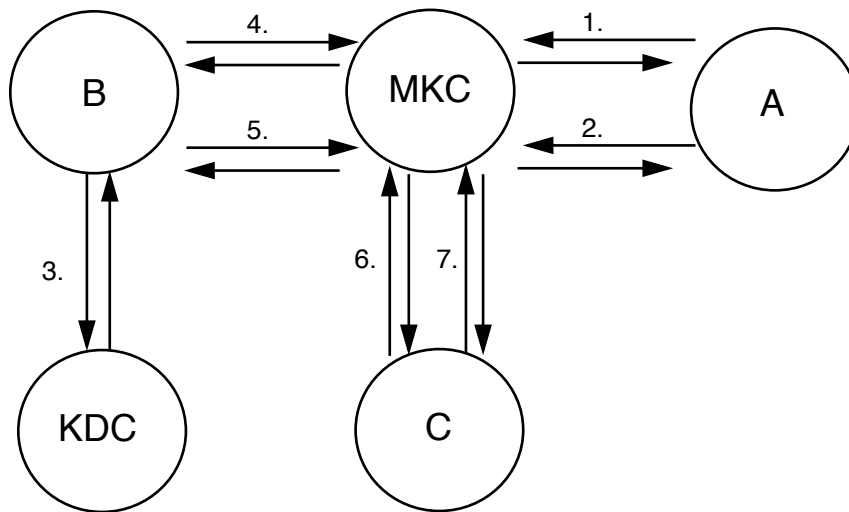
3.3 Key management model

The cryptographic keying material and security attributes that define a security association are represented as managed objects and stored in the SMIB. Security-related managed objects in the SMIB are available to security protocols across all layers of the OSI Reference Model. The KMP described here allows for the creating, spawning, and deleting of security associations within the SMIB. Creating a security association establishes new keying material and security attributes within the SMIB, spawning uses information from a security association in the SMIB to generate a new one, and deleting a security association makes the keying material and security attributes unavailable. Deleting a security association may remove the keying material or security attributes from the SMIB, if local security policy permits.

KMP supports two types of security associations: pairwise and multicast. Pairwise security associations support protected communication between two parties that share the same cryptographic keying material and security attributes. Multicast security associations support communication among two or more parties, where all parties share the same cryptographic keying material and security attributes.

3.3.1 Security association lifecycle

The creation of security associations is accomplished in two phases: establishment of cryptographic keying material, and negotiation of security attributes. Once cryptographic keying material is established, it is used to protect the security attribute negotiation.



1. A and the MKC select a key that was previously distributed manually.
2. A gets the multicast key from the MKC.
3. B requests a key from the KDC to communicate with the MKC.
4. B sends a copy of the key received from the KDC to the MKC. The key is protected with a key encryption key that only the MKC knows.
5. B gets the multicast key from the MKC.
6. C and the MKC exchange key generation information, which they use to make a secret key.
7. C gets the multicast key from the MKC.

Figure 3-5—Multicast key distribution

The negotiated security attributes determine how the security association will be used. In other words, the security attributes determine which Security Protocol will use the security association and how that Security Protocol will use the cryptographic keying material.

After a security association is created, it may be used to spawn other security associations. Spawn operations allow the generation of a new security association with new cryptographic keying material derived from that of the parent association and the same security attributes; the same cryptographic keying material and new security attributes; or new cryptographic keying material derived from that of the parent association and new security attributes. In every case, spawn operations generate a new security association with a new Security Association Identifier (SAID). Spawn operations do not affect the parent security association.

Either key management end system may determine that a security association is no longer needed. Once the determination is made, the local system notifies the remote system, and deletes the association. No confirmation is necessary.

3.3.2 Key management application entity structure

The key management application requires communication with remote key management applications. To support this communication, the KMP relies on OSI communication services provided by the ACSE and SESE. These service elements interface to the OSI Presentation Layer to communicate with the remote key management end system.

The Key Management Application Process (KMAP) is not specified by this standard, but the KMAP uses the OSI communications services to establish cryptographic keying material, negotiate attributes for the cryptographic keying material, and delete the cryptographic keying material in accordance with locally defined security policy. The KMAP provides the top level interface to the user. The KMAP may use one or more of the supported cryptographic keying material distribution techniques: manual key distribution, center-based key distribution, and certificate-based key distribution. A single attribute negotiation technique is used with all three cryptographic keying material distribution techniques. This protocol is used to establish security associations that provide security services.

The Key Management Application Entity (KMAE) is an Application Service Object (ASO) that performs cryptographic keying material management. As shown in Figure 3-6, the KMAE comprises two application service objects and a control function. The Key Peer Application Service Object (KPASO) interacts with the remote key management end system to establish security associations. The Key Center Application Service Object (KCASO) interacts with a KDC or a KTC to obtain cryptographic keying material. The Control Function (CF) coordinates interactions between the KPASO and the KCASO. The service interface to the KMAE is described in 3.4.1.

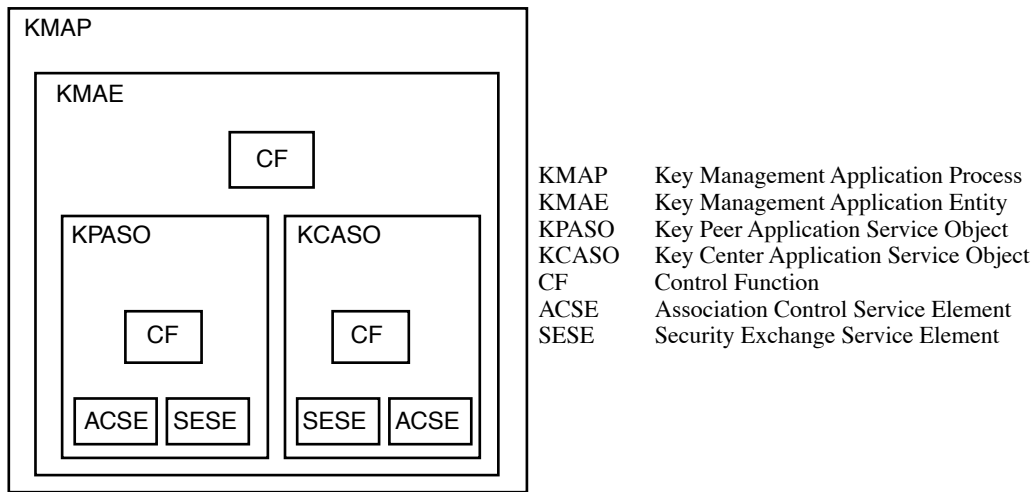


Figure 3-6—Key management application structure

The KPASO comprises the ACSE, the SESE, and a CF. The ACSE is a common application service element defined by ISO/IEC 8649: 1996, and it establishes and terminates application-associations. The SESE is a common application service element defined by ISO/IEC 11586: 1996, and it performs security exchanges and security transformations. The SESE provides the SE-Transfer operation to the CF in order to transport a Security Exchange Item (SEI). Each protocol data unit in this protocol is represented as an SEI. The CF coordinates interactions between ACSE and SESE. The service interface to the KPASO is described in 3.4.2.

The KMAE may invoke the KPASO multiple times, once for each remote key management end system. Therefore, each KPASO invocation uses a single application-association. Limiting the KPASO to one application-association simplifies addressing.

The KCASO, like the KPASO, comprises the ACSE, the SESE, and a CF. Again, the CF coordinates interactions between ACSE and SESE. The service interface to the KCASO is described in 3.4.3.

The KMAE may invoke the KCASO multiple times, once for a KDC or for a KTC. Therefore, each KCASO invocation uses a single application-association. Again, limiting the KCASO to one application-association simplifies addressing.

3.3.3 Sequencing of application layer services

The KMAE does not implement any security exchanges; rather, the KMAE invokes the KPASO and the KCASO, which perform the necessary security exchanges. Communication between the KMAE and the KPASO, communication between the KMAE and the KCASO, and sequencing of these communications are controlled by the CF within the KMAE; however, these communications are an implementation detail and are not specified.

The KMAE may optionally use the KPASO to negotiate the key distribution technique and the algorithm that will be used to establish symmetric cryptographic keying material. Regardless of how the symmetric keying material is established, the KPASO uses the symmetric keying material to protect the security attribute negotiation exchange necessary to complete the creation of the security association.

When using center-based techniques, the KMAE uses the KCASO to obtain symmetric cryptographic keying material from a KDC, or to translate a key using a KTC. This symmetric cryptographic keying material is subsequently used by the KPASO to complete the creation of the security association.

3.3.3.1 Manually distributed key

To select cryptographic keying material from a collection of pre-positioned material or to select cryptographic keying material generated by transforming pre-positioned material, a Create-SA service request is issued to the KMAE.

In order to implement manually distributed cryptographic keying material, it is generated a-priori and stored on the peer key management systems via means other than KMP. The cryptographic keying material is stored in a memory cache and accessed using Keying Material Identifiers (KMIDs). Validity periods may be associated with the cached cryptographic keying material. The cache is usually located in the SMIB. The cache with KMIDs is analogous to an array with indices; see Table 3-1. If cryptographic keying material transformation is supported, then the cache must include a counter that tells the number of times a transformation has been applied to the manually distributed cryptographic keying material. Keeping this counter allows earlier versions of the cryptographic keying material, including the original, to be discarded when they are no longer needed.

Table 3-1—Cryptographic keying material cache

KMID	Cryptographic keying material	Transform counter
1	0x0123456789ABCDEF	0
2	0XF0DEBC9A78563412	48

The KPASO CF generates the proper SEIs from the parameters, and then passes the SEIs to SESE via the SE-Transfer service. The KPASO CF passes the SEIs to SESE only after an application-association is in place for the SESE to use. If an application-association is not already in place, the KPASO CF invokes the A-Associate service provided by ACSE. If the Create-SA parameters include a Key Management Technique Identifier (KMTI) List with multiple techniques (see 3.4.1.1.3), or the Security Policy Identifier must be announced (see 3.4.1.1.4), then the KMAE CF invokes the KPASO Pick-KM-Alg service. Once the KMAE CF has selected the manual key distribution technique, the KMAE CF issues the Select-Key service to the KPASO. Then, the KMAE CF uses the Security Protocol Parameters List to issue the Pick-SA-Attrs. If further key management communications are likely, the KMAE CF leaves the application-association in place; otherwise the KMAE CF invokes the Release-P service, which releases the application-association.

Figure 3-7 shows the relationship between the KMAE and the KPASO. As mentioned previously, the Release-P service is optional, but has been included in the figure for completeness. Only service requests are shown; confirmations have been omitted to simplify the figure.

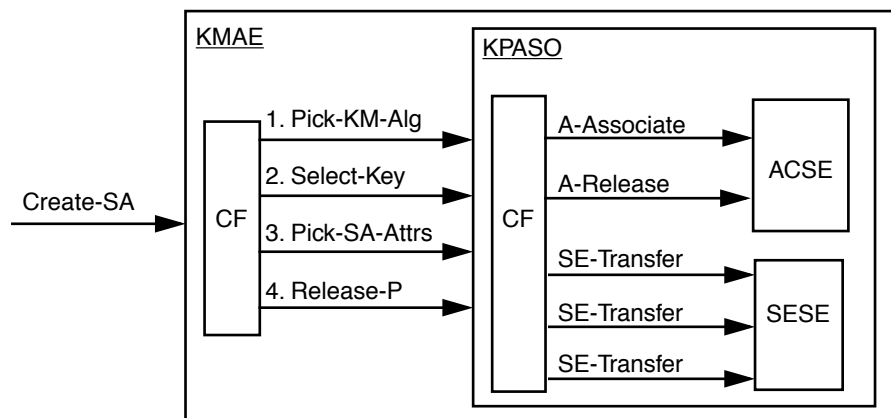


Figure 3-7—Create an SA by selecting manually distributed key

3.3.3.2 Key center distribution

A Create-SA service request is issued to the KMAE to obtain cryptographic keying material from a KDC or to translate cryptographic keying material using a KTC. If the Create-SA parameters include the KMTI List with multiple techniques (see 3.4.1.1.3), or the Security Policy Identifier must be announced (see 3.4.1.1.4), then the KMAE CF invokes the KPASO Pick-KM-Alg service. The KPASO generates the proper SEIs from the parameters, and then passes the SEIs to SESE via the SE-Transfer service. The KPASO CF passes the SEIs to SESE only after an application-association is in place for the SESE to use with its peer in the remote key management end system. If an application-association is not already in place, the KPASO CF invokes the A-Associate service provided by ACSE.

Once the KMAE CF has selected the key center or translation center distribution technique, the KMAE CF issues the Request-Key or Translate-Key service to the KCASO. In the same manner as the KPASO, the KCASO CF passes the SEIs to SESE only after an application-association is in place for the SESE to use with its peer in the remote key or translation center end system. If an application-association with the KDC or KTC does not already exist, the KCASO CF invokes the A-Associate service provided by ACSE. After the KMAE CF receives confirmation from the Request-Key or Translate-Key service, the KMAE CF issues the Send-Key service to the KPASO. Then the KMAE CF uses the Security Protocol Parameters List to issue the Pick-SA-Attrs. If further key management communications are likely, the KMAE CF leaves both application-associations in place; otherwise the KMAE CF invokes the Release-P and Release-C service, which releases the application-associations.

Figure 3-8 shows the relationship between the KMAE, the KPASO, and the KCASO. As mentioned previously, the Release-P and Release-C services are optional, but have been included in the figure for completeness. Only service requests are shown; confirmations have been omitted to simplify the figure.

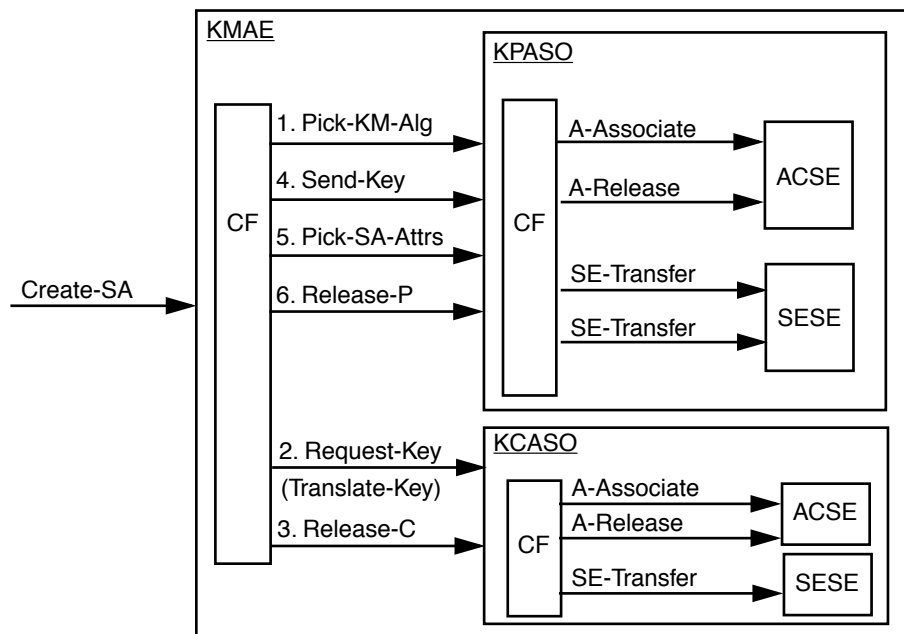


Figure 3-8—Create an SA using center-based key distribution

In cases where the communications topology prohibits the local key management end system from communicating directly with the KDC or KTC, the local KMAE CF issues the Please-Send-Key service to the KPASO requesting the remote key management end system to communicate with the KDC or KTC. In response, the remote KMAE CF issues the Request-Key or Translate-Key service to the remote KCASO, and then the remote KMAE CF issues the Send-Key service to the remote KPASO. Afterwards, the local KMAE CF uses the Security Protocol Parameters List to issue the Pick-SA-Attrs. If further key management communications are likely, the KMAE CF leaves application-association in place; otherwise the KMAE CF invokes the Release-P, which releases the application-association. If further key management communications are likely between the remote key management end system and the KDC or KTC, the remote KMAE CF leaves application-association in place; otherwise the remote KMAE CF invokes the Release-C, which releases the application-association.

3.3.3.3 Certificate-based key distribution

To generate a key using certificate-based techniques, a Create-SA service request is issued to the KMAE. If the Create-SA parameters include the KMTI List with multiple techniques (see 3.4.1.1.3) or the Security Policy Identifier must be announced (see 3.4.1.1.4), then the KMAE CF invokes the KPASO Pick-KM-Alg service. The KPASO generates the proper SEIs from the parameters, and then passes the SEIs to SESE via the SE-Transfer service. The KPASO CF passes the SEIs to SESE only after an application-association is in place for the SESE to use. If an application-association is not already in place, the KPASO CF invokes the A-Associate service provided by ACSE. Once the KMAE CF has selected the certificate-based key distribution technique, the KMAE CF issues the Make-Key service to the KPASO. Then the KMAE CF uses the Security Protocol Parameters List to issue the Pick-SA-Attrs. If further key management communications are likely, the KMAE CF leaves the application-association in place; otherwise the KMAE CF invokes the Release-P service, which releases the application-association.

Figure 3-9 shows the relationship between the KMAE and the KPASO. As mentioned previously, the Release-P service is optional, but has been included in the figure for completeness. Only service requests are shown; confirmations have been omitted to simplify the figure.

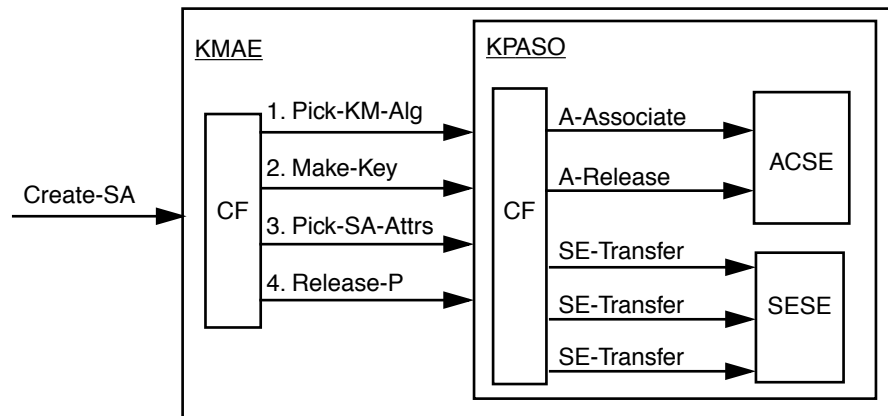


Figure 3-9—Create an SA using certificate-based key distribution

3.3.3.4 Multicast key distribution

The MKC generates the multicast keying material and the associated attributes, and then distributes them to the broadcast or multicast group members. The MKC determines the period of time that multicast keying material may be used. The KMAE uses the Create-MSA service to obtain multicast keying material from the MKC. The KMAE uses the Spawn-MSA service to obtain subsequent multicast keying material from the MKC.

There is no need for a Delete-MSA service. With pairwise security associations, the Delete-SA service is used to notify the remote KMAE that pairwise cryptographic keying material is being removed from the local SMIB. This notification is important in the pairwise case because communications associated with that cryptographic keying material are no longer possible. This is not the case with multicast keying material. When one KMAE deletes multicast keying material from the local SMIB, other members of the broadcast or multicast group can continue to communicate using that cryptographic keying material. The local deletion of multicast keying material has the effect of resigning from the broadcast or multicast group. The method used to delete multicast security associations is an implementation detail.

The security association identifier and all other security association attributes are assigned by the MKC so that the same ones will be used by all Security Protocol entities who possess the multicast keying material.

3.3.3.4.1 Create multicast security association

For Multicast keys, a Create-MSA service request is issued to the KMAE. If the Create-SA parameters include the KMTI List with multiple techniques (see 3.4.1.1.3), or the Security Policy Identifier must be announced (see 3.4.1.1.4), then the KMAE CF invokes the KPASO Pick-KM-Alg service. The KPASO generates the proper SEIs from the parameters, and then passes the SEIs to SESE via the SE-Transfer service. The KPASO CF passes the SEIs to SESE only after an application-association is in place for the SESE to use with the MKC. If an application-association is not already in place, the KPASO CF invokes the A-Associate service provided by ACSE.

Once the key distribution technique has been determined, the KMAE CF issues either the Make-Key, Select-Key, Send-Key, or Please-Send-Key service to the KPASO in order to establish keying material with the MKC. The Protected-Make-Key service may optionally be invoked following the completion of any of these services. After the KMAE CF has received confirmation from the KPASO for the chosen service, the KMAE CF issues the Get-MKey with a list of broadcast and multicast addresses. The MKC returns the list of approved broadcast and multicast addresses with a corresponding list of multicast tokens. The multicast token includes the multicast SAID. If further key management communications are likely, the application-

association is left in place; otherwise the KMAE CF invokes the Release-P service, which releases the application-association.

Figure 3-10 shows the relationship between the KMAE and the KPASO. As mentioned previously, the Release-P service is optional, but has been included in the figure for completeness. Only service requests are shown; confirmations have been omitted to simplify the figure.

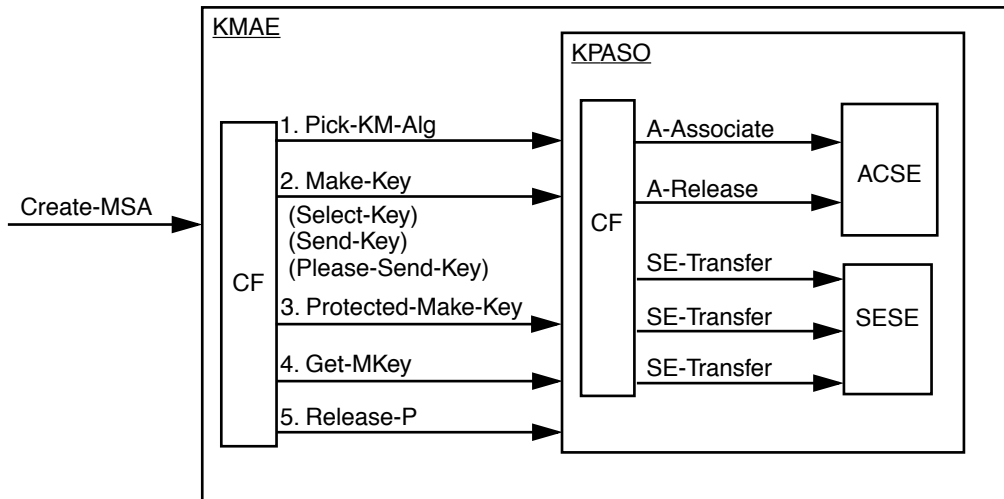


Figure 3-10—Multicast key distribution

3.3.3.4.2 Spawn multicast security association

To generate a new multicast security association from an existing one, a Spawn-MSA service request is issued to the KMAE. The KMAE CF uses a previously established multicast security association to protect the transfer of the new multicast token from the MKC. The KMAE CF invokes the KPASO service Get-Next-MKey to the MKC. The MKC returns the new multicast token, which includes the multicast SAID. The KPASO generates the proper SEIs from the passed parameters, and then passes the SEIs to the SESE via the SE-Transfer service. The KPASO CF passes the SEIs to SESE only after an application-association is in place for the SESE to use with the MKC. If an application-association is not already in place, the KPASO CF invokes the A-Associate service provided by ACSE. If further key management communications are likely, the application-association is left in place; otherwise the KMAE CF invokes the Release-P service, which releases the application-association.

Figure 3-11 shows the relationship between the KMAE and the KPASO. As mentioned previously, the Release-P service is optional, but has been included in the figure for completeness. Only service requests are shown; confirmations have been omitted to simplify the figure.

3.3.3.5 Spawn security association

A Spawn-SA service request is issued to the KMAE to generate a new security association from an existing security association by one of the following Spawn Options:

- Transform keying material and use the same attributes;
- Generate or transfer new keying material under the protection of an existing security association and use the same attributes;
- Transform keying material and negotiate new attributes;
- Generate or transfer new keying material under the protection of an existing security association and negotiate new attributes; or
- Use the same keying material, and negotiate new attributes.

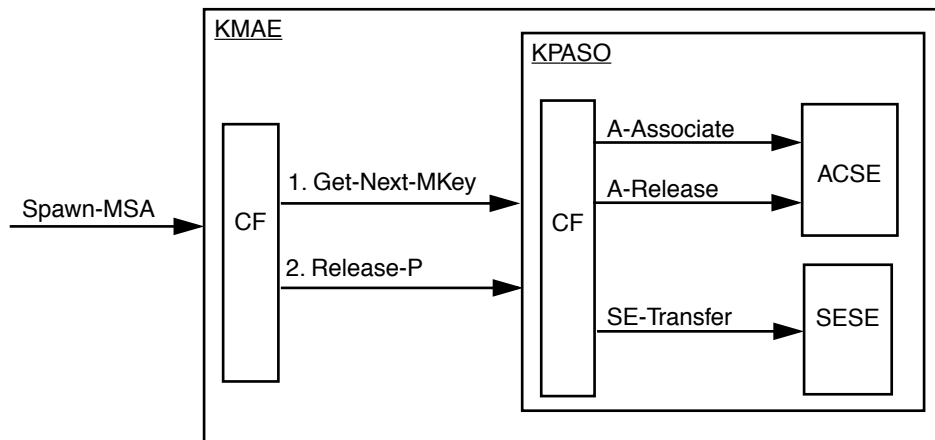


Figure 3-11 – Multicast spawn

The KMAE CF uses information from the Spawn Option, the Key Transformation Algorithm Identifier, and the previously established calling and called SAIDs to invoke either the Spawn-Key, Protected-Make-Key, or Send-Key KPASO service. All of these services assign a new SAID.

To change only the attributes while preserving the keying material, the KMAE must invoke the Spawn-Key service with an identity key transformation algorithm.

The KPASO generates the proper SEIs from the parameters, and then passes the SEIs to the SESE via the SE-Transfer service. The KPASO CF passes the SEIs to SESE only after an application-association is in place for the SESE to use with its peer in the remote key management end system. If an application-association is not already in place, the KPASO CF invokes the A-Associate service provided by ACSE. The KMAE CF uses the Security Protocol Parameters List to issue the Pick-SA-Attrs to the KPASO. If further key management communications are likely, the application-association is left in place; otherwise the KMAE CF invokes the Release-P service, which releases it.

Figure 3-12 shows the relationship between the KMAE and the KPASO. As mentioned previously, the Release-P service is optional, but has been included in the figure for completeness. Only service requests are shown; confirmations have been omitted to simplify the figure.

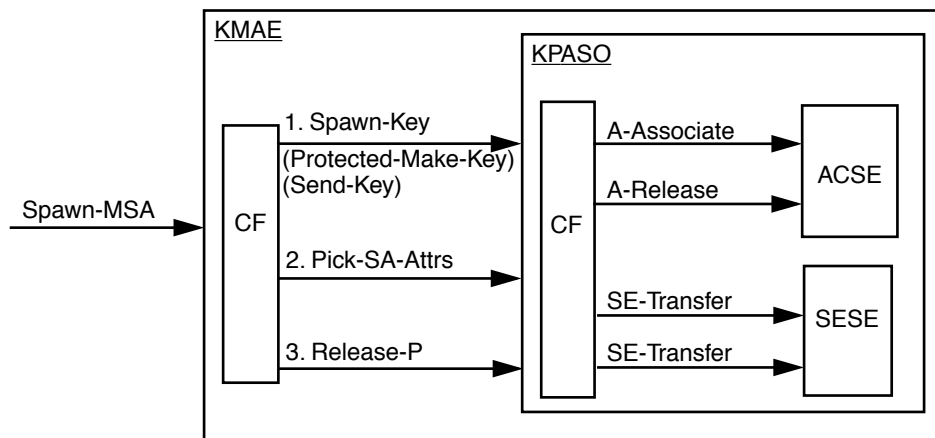


Figure 3-12—Spawn SA

3.3.3.6 Delete security association

To delete a security association, a Delete-SA service is issued to the KMAE. The KMAE CF uses the Calling and Called SAIDs to issue the Delete-Key service to the KPASO. The KPASO generates the proper SEIs from the parameters, and then passes the SEIs to the SESE via the SE-Transfer service. The KPASO CF passes the SEIs to SESE only after an application-association is in place for the SESE to use with its peer in the remote key management end system. If an application-association is not already in place, the KPASO CF invokes the A-Associate service provided by ACSE. If further key management communications are likely, the application-association is left in place; otherwise the KMAE CF invokes the Release-P service, which releases it.

Figure 3-13 shows the relationship between the KMAE and the KPASO. As mentioned previously, the Release-P service is optional, but has been included in the figure for completeness. Only service requests are shown; confirmations have been omitted to simplify the figure.

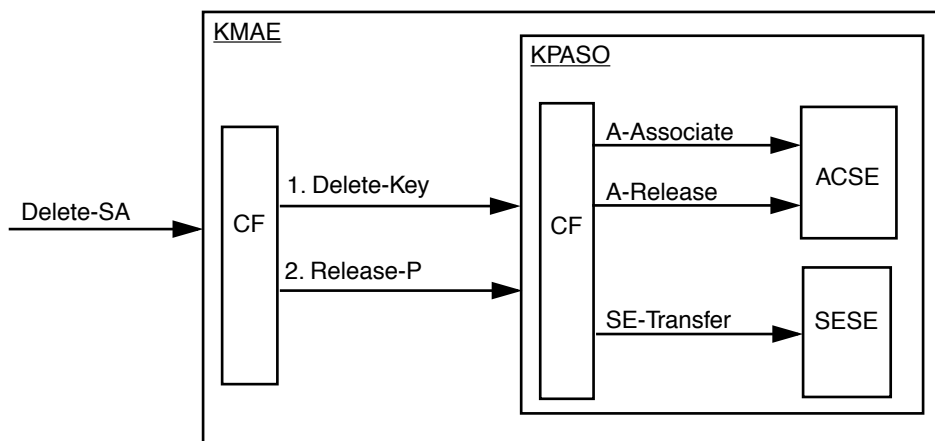


Figure 3-13—Delete SA

3.4 Service definition

The following abbreviations are used in this subclause:

- M Mandatory
- U User Optional
- C Conditional
- (=) If present, the parameter value must be the same as the parameter value in the previous column

The KMAE provides services to the KMAP using a service interface with a very high level of abstraction. The KPASO and KCASO provide service to the KMAE using a service interface with a lower level of abstraction. The KMAE CF determines when application-associations should be established and released. The time at which application-associations should be released is not addressed by this standard; it is a local matter.

3.4.1 Key management application entity (KMAE) services

The KMAE provides five services: Create-SA, Spawn-SA, Delete-SA, Create-MSA, and Spawn-MSA.

3.4.1.1 Create security association (Create-SA)

The Create-SA service establishes the cryptographic keying material and associated attributes to form a security association. Create-SA provides a confirmed service, and it securely establishes the same symmetric cryptographic keying material in the SMIB of each key management end system. The specific attributes that are associated with the cryptographic keying material depend on the Security Protocol that will use the security association. Issuing a Create-SA.conf announces that the security association is enabled on both end systems for use by the security protocol.

The parameters of Create-SA are defined in the Table 3-2.

Table 3-2—Create-SA parameters

Create-SA parameter name	.req	.ind	.rsp	.conf	Reference
Calling AE-Title	M	M (=)	—	—	3.4.1.1.1
Called AE-Title	M	M (=)	M	M (=)	3.4.1.1.2
Key Management Technique Identifier List	U	C (=)	—	—	3.4.1.1.3
Security Policy Identifier	U	C (=)	—	—	3.4.1.1.4
Security Association Attributes List	M	M (=)	—	—	3.4.1.1.5
Security Association Attributes	—	—	M	M (=)	3.4.1.1.6
Calling SAID	—	M	M (=)	M (=)	3.4.1.1.7
Called SAID	—	M	M (=)	M (=)	3.4.1.1.8
Result	—	—	M	M (=)	3.4.1.1.9

3.4.1.1.1 Calling AE-Title

The Calling AE-Title comprises the Application Title and Application Entity Qualifier of the calling KMAE, both of which are necessary during the application-association establishment and are passed from the calling key management application. The form of the AE-Title could either be the distinguished name of the application or an object identifier for the application.

CallingAETitle ::= AETitle

AETitle ::= CHOICE {
 AETitle-form1,
 AETitle-form2 }

AETitle-form1 ::= [0] SEQUENCE {
 COMPONENTS OF {
 APTtitle-form1,
 AEQualifier-form1 }}}

APTtitle-form1 ::= IMPLICIT SEQUENCE OF
 RelativeDistinguishedName

AEQualifier-form1 ::= [1] IMPLICIT
 RelativeDistinguishedName

AETitle-form2 ::= [1] SEQUENCE {

```
APTtitle-form2,  
AEQualifier-form2 }
```

```
APTtitle-form2 ::= [2] IMPLICIT OBJECT IDENTIFIER
```

```
AEQualifier-form2 ::= [3] IMPLICIT INTEGER
```

3.4.1.1.2 Called AE-Title

The Called AE-Title comprises the Application Title and Application Entity Qualifier of the called KMAE, both of which are necessary during the application-association establishment and passed from the calling key management application to indicate the intended called KMAE. The form of the AE-Title can either be the distinguished name of the application or an object identifier for the application.

```
CalledAETitle ::= AETitle -- AETitle is defined in 3.4.1.1.1
```

3.4.1.1.3 Key management technique identifier list

The Key Management Technique Identifier (KMTI) List contains a list of alternatives, in order of preference, regarding the type of key management distribution and the algorithms and values needed for key generation, confidentiality, integrity, and key transformation.

If the KMTI List contains more than one alternative or the Security Policy Identifier must be announced, then the KMAE CF must invoke Pick-KM-Alg. In many cases, the calling key management end systems support only a single key management technique; thus negotiation is unnecessary, and the KMAE CF may omit invocation of Pick-KM-Alg.

The KMTI is the selection made by the called KMAE from the list of alternatives in the KMTI List. The selection is the first acceptable alternative from the list in the KMTI List. The selection includes the type of key management distribution and the algorithms and values needed, if any, for key generation. Based on the value of the technique, the KMAE CF subsequently invokes Select-Key if the technique is equal to 0, Send-Key if the technique is equal to 1 (after it invokes either a Request-Key or a Translate-Key), Please-Send-Key if the technique is equal to 2, or Make-Key if the technique is equal to 3. Also, included in the selection is the confidentiality and integrity algorithms that the KMAE uses in subsequent exchanges during the establishment of the Security Association, i.e., within the Pick-SA-Attrs exchange. Also, the KMAE may subsequently use the key transformation algorithm within the Spawn-SA exchanges.

```
KMTechniqueIdList ::= SEQUENCE OF KMTechniqueId
```

```
KMTechniqueId ::= SEQUENCE {  
    technique          Technique,  
    keyCreationAlg      KeyCreationAlg,  
    attrConfidAlg       AttrConfidAlg,  
    attrIntegrityAlg    AttrIntegrityAlg,  
    transformationAlg   TransformationAlg OPTIONAL }
```

```
Technique ::= INTEGER {  
    manual (0),  
    requesterCallsCenter (1),  
    responderCallsCenter (2),  
    certificate (3) }
```

```
KeyCreationAlg ::= AlgorithmIdentifier
```

```
AttrConfidAlg ::= AlgorithmIdentifier
```

```
AttrIntegrityAlg ::= AlgorithmIdentifier
TransformationAlg ::= AlgorithmIdentifier
```

3.4.1.1.4 Security policy identifier

The Security Policy Identifier is used by the calling KMAE to indicate a specific security policy to the called KMAE. In addition, authentication information can follow the policy identifier. The Security Policy Identifier is maintained in the SMIB, and it can be used to implicitly state security association requirements such as security service requirements, keying material selection, security protocol selection, and security protocol attributes. This parameter is optional.

```
SecurityPolicy ::= SEQUENCE {
    policyId          OBJECT IDENTIFIER,
    policyVersion     INTEGER OPTIONAL,
    authenticationInfo OCTET STRING OPTIONAL }
```

3.4.1.1.5 Security association attributes list

The Security Association Attributes List contains information regarding the suite of security attributes required for the security association. For example, the security attributes could be the attributes for a security protocol. The information is placed in the list in order of preference by the calling KMAE. The values are determined by the security policy and maintained in the SMIB.

```
SAAattrsList ::= SEQUENCE of SAAattrs
```

3.4.1.1.6 Security association attributes

The Security Association Attributes is the selection made by the called KMAE from the Security Association Attributes List. The selection is made by the first match between the calling and called KMAE. The security attributes may be enumerated or they may be specified by referencing a previously established security association. If desired, padding may be included to make all of the alternatives approximately the same size. Any padding is ignored by the remote key management end system. An example of attributes for the SDE security protocol is contained in Annex 3E. The definition of the Object Class SP-ATTRS is contained in 3.5.5.2.

```
SAAattrs ::= CHOICE {
    sameAs      [0] ReferencesSA,
               [1] SPAttrs }
ReferencesSA ::= SEQUENCE {
    callingSAID Said,
    calledSAID  Said,
    padding     OCTET STRING OPTIONAL }
SPAttrs ::= SEQUENCE {
    spAttrs      SP-ATTRS.&attrs-Id
    attrs        SP-ATTRS.&Sp-Attrs {@spAttrs} OPTIONAL }
```

3.4.1.1.7 Calling SAID

The Calling SAID is a unique identifier for the Security Association and is determined by the calling KMAE. The calling and called KMAE determine their own SAIDs; therefore, the calling and called SAIDs may differ. The KMAE creates the SAID during the key establishment process.

The SAID is used by this KMP, and the format of the calling and called SAID is identical to the format of the SAID defined in 2.5.2.1.2 of IEEE Std 802.10-1992. If a security protocol requires its own SAID with a different format, then this SAID must be established with the selection of security attributes.

The SDE SAID format provides an indicator, the G-bit, for identifying security association types. For the security association created by this service, the KMAE must set the G-bit to FALSE, indicating a pairwise security association.

CallingSAID ::= OCTET STRING

3.4.1.1.8 Called SAID

The Called SAID is a unique identifier for the Security Association and is determined by the called KMAE. The calling and called KMAE determine their own SAIDs; therefore, the calling and called SAIDs may differ. The SAID is created during the key generation process.

CalledSAID :: = OCTET STRING

3.4.1.1.9 Result

The Result contains the exit status of the KMAE, KPASO, or KCASO services. A zero indicates success and one of the positive integers listed below indicates failure and corresponds to an error code.

```
Result ::= INTEGER {
  success(0),
  -- The service has completed successfully.

  -- General Error Codes
  failed_no_reason(1),
  -- The service did not complete successfully with no reason given.
  service_not_available(2),
  -- The underlying service provider is not available.
  service_not_available_at_this_time(3),
  -- The underlying service is not available at the particular moment and
  -- the caller is encouraged to try again at a later time.
  illegal_ASN1_value_bad_value(4),
  -- The ASN values passed from the calling party contain a bad value and
  -- the call is rejected.
  illegal_ASN1_value_missing_field(5),
  -- The ASN values passed from the calling party do not contain fields
  -- that are mandatory and the call is rejected.
  illegal_ASN1_value_unknown_field(6),
  -- The valid ASN values passed from the calling party contain an unknown
  -- field and the call is rejected.
  provider_abort_received(7),
  -- The underlying service provider aborted causing the service to fail.
  user_abort_received(8),
  -- The user service on the local side initiated an abort causing the
  -- service to fail.

  -- KMP Specific Error Codes
  negotiation_failure(64),
  -- The optional negotiation phase (Pick-KM-Algs) was not successful.
  error_generating_key_material(65),
  -- The key generation process (Select-Key, Send-Key, Make-Key, or
  -- Protected-Make-Key) could not successfully generate the new keying
  -- material.
  error_retrieving_key_material(67),
```

```

-- The key retrieval process (Request-Key or Translate-Key) could not
-- successfully retrieve the new keying material from the key center.
not_ready_for_protected_service(68),
-- Even though the key generation process completed successfully, the
-- protective security mechanism was not ready.
error_retrieving_multicast_key(69),
-- The key retrieval process (Get-MKey or Get-Next-Mkey) could not
-- successfully retrieve the new multicast key from the MKC.
unknown_security_transformation(70),
-- The security transformation was unknown during the Spawn-Key service.
error_processing_SA_attributes(71),
-- The attribute negotiation process (Pick-SA-Attrs) was not successful.
invalid_called_SAID(72),
-- The called SAID was invalid.
invalid_calling_SAID(73),
-- The calling SAID was invalid.
release_peer_urgent(74),
-- The KPASO generated an urgent release request reason.
release_peer_user_defined_rq(75),
-- The KPASO generated an user defined release request reason.
release_peer_not_finished(76),
-- The KPASO generated a not finished release response reason.
release_peer_user_defined_rp(77),
-- The KPASO generated an user defined release response.
release_center_urgent(78),
-- The KCASO generated an urgent release request reason.
release_center_user_defined_rq(79),
-- The KCASO generated a user defined release request reason.
release_center_not_finished(80),
-- The KCASO generated a not finished release response reason.
release_center_user_defined_rp(81),
-- The KCASO generated a user defined release response.
abort_peer_user(82),
-- The KPASO generated an user abort.
abort_peer_provider(83),
-- The KPASO generated a provider abort.
abort_center_user(84),
-- The KCASO generated an user abort.
abort_center_provider(85),
-- The KCASO generated a provider abort.
expired_key(86),
-- The Select-Key request referenced keying material that has expired.
unavailable_key(87),
-- The referenced keying material is not available at this time.
transform_count(256-511)
-- Contains the number of transformations to the base keying material
-- needed to derive shared keying material. To determine this value,
-- subtract 256 from the error code received.
}

```

3.4.1.2 Spawn security association (Spawn-SA)

The Spawn-SA service uses a previously established security association to establish a new security association. Spawn-SA can create a new security association from an existing security association in one of the following Spawn Options:

- Transform keying material and use the same attributes;
- Generate or transfer new keying material under the protection of an existing security association and use the same attributes;
- Transform keying material and negotiate new attributes;
- Generate or transfer new keying material under the protection of an existing security association and negotiate new attributes; or
- Use the same keying material, and negotiate new attributes.

Even if the keying material and all of the attributes are the same, the security association identifier is different for the two security associations. The previously established security association remains unaffected by the Spawn-SA service. Spawn-SA provides a confirmed service. The enabling of cryptographic keying material and associated attributes within the SMIB for use by the Security Protocol is a local matter. However, when the Spawn-SA.conf is provided, the cryptographic keying material and associated attributes have been enabled for use by the Security Protocols on both of the peer key management end systems.

The Spawn-SA exchange is protected by the previously established security association. This protection may provide peer entity authentication. Other protection depends on the services provided by the previously established security association; confidentiality, integrity, or both may be provided.

The parameters of Spawn-SA are defined in Table 3-3.

Table 3-3—Spawn-SA parameters

Spawn-SA parameter name	.req	.ind	.rsp	.conf	Reference
Calling AE-Title	M	M (=)	—	—	3.4.1.1.1
Called AE-Title	M	M (=)	M	M (=)	3.4.1.1.2
Spawn Option	M	M (=)	M	M (=)	3.4.1.2.1
Key Transformation Algorithm Identifier	U	C (=)	—	—	3.4.1.2.2
Security Association Attributes List	U	C (=)	—	—	3.4.1.1.5
Security Association Attributes	—	—	U	C (=)	3.4.1.1.6
Previously Established Calling SAID	M	M (=)			3.4.1.2.3
Previously Established Called SAID	M	M (=)			3.4.1.2.4
Calling SAID	—	M	M (=)	M (=)	3.4.1.1.7
Called SAID	—	—	M	M (=)	3.4.1.1.8
Result	—	—	M	M (=)	3.4.1.1.9

3.4.1.2.1 Spawn option

The Spawn Option indicates whether the Spawn-SA service is to negotiate new attributes for use with existing keying material associated with a previously established security association, transform the symmetric

keying material, make new keying material while under the protection of an existing security association, or transfer newly generated keying material under the protection of a previously established security association. The Spawn Option values are samekey for new attributes with existing key, update for transforming symmetric keying material, create for generating new keying material under the protection of an existing security association, and transfer for sending keying material under the protection of a previously established security association. The value for Spawn Option is selected by the Key Management Application. When the value is samekey (0), the KMAE invokes the Spawn-Key service with the identity key transformation algorithm, in order to assign a new SAID. When the value is update (1), the KMAE invokes the Spawn-Key service with the transformation algorithm identifier. When the value is create (2), the KMAE invokes the Protected-Make-Key service. When the value is transfer (3), the KMAE invokes the Send-Key service.

```
SpawnOption ::= INTEGER {
    samekey      (0),
    update       (1),
    create       (2),
    transfer     (3) }
```

3.4.1.2.2 Key transformation algorithm identifier

The Key Transformation Algorithm Identifier specifies the one-way transformation to be used by both the calling and called KMAEs to transform existing keying material into the new keying material for the security association. The Key Transformation Algorithm Identifier is set by the calling KMAE and is maintained in the SMIB or passed from the Key Management Application. The Key Transformation Algorithm Identifier is composed of an OBJECT IDENTIFIER and an optional INTEGER parameter, called transformCount. The transformCount specifies the number of times that the one-way transformation is applied to the original keying material.

The Key Transformation Algorithm Identifier was negotiated when the KMAE created the previous security association. If multiple keys comprise the previously established security association, the identifier tells the KMAE how to transform each key.

```
KeyTransformationAlgorithmIdentifier ::= AlgorithmIdentifier
```

3.4.1.2.3 Previously established calling SAID

The Previously Established Calling SAID is the unique identifier for the calling Security Association that is to be used as a template for the spawned security association. The previously established security association remains unaffected during the spawn process. The Previously Established Calling SAID and the Previously Established Called SAID must reference the same security association. The value for the Previously Established Calling SAID is maintained in the SMIB and is set by the calling KMAE.

```
PreviouslyEstablishedCallingSAID ::= OCTET STRING
```

3.4.1.2.4 Previously established called SAID

The Previously Established Called SAID is the unique identifier for the called Security Association that is to be used as a template for the spawned security association. The previously established security association remains unaffected during the spawn process. The Previously Established Calling SAID and the Previously Established Called SAID must reference the same security association. The value for the Previously Established Called SAID is maintained in the SMIB and is provided by the calling KMAE.

```
PreviouslyEstablishedCalledSAID ::= OCTET STRING
```