The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

SPACE DATA LINK SECURITY PROTOCOL—EXTENDED PROCEDURES

RECOMMENDED STANDARD

CCSDS 355.1-B-1

BLUE BOOK

February 2020

The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

SPACE DATA LINK SECURITY PROTOCOL—EXTENDED PROCEDURES

RECOMMENDED STANDARD

CCSDS 355.1-B-1

BLUE BOOK

February 2020

AUTHORITY

Issue:	Recommended Standard, Issue 1
Date:	February 2020
Location:	Washington, DC, USA

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and represents the consensus technical agreement of the participating CCSDS Member Agencies. The procedure for review and authorization of CCSDS documents is detailed in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4), and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the email address below.

This document is published and maintained by:

CCSDS Secretariat
National Aeronautics and Space Administration
Washington, DC, USA
Email: secretariat@mailman.ccsds.org

STATEMENT OF INTENT

The Consultative Committee for Space Data Systems (CCSDS) is an organization officially established by the management of its members. The Committee meets periodically to address data systems problems that are common to all participants, and to formulate sound technical solutions to these problems. Inasmuch as participation in the CCSDS is completely voluntary, the results of Committee actions are termed **Recommended Standards** and are not considered binding on any Agency.

This **Recommended Standard** is issued by, and represents the consensus of, the CCSDS members. Endorsement of this **Recommendation** is entirely voluntary. Endorsement, however, indicates the following understandings:

- Whenever a member establishes a CCSDS-related **standard**, this **standard** will be in accord with the relevant **Recommended Standard**. Establishing such a **standard** does not preclude other provisions which a member may develop.
- Whenever a member establishes a CCSDS-related **standard**, that member will provide other CCSDS members with the following information:
  - The **standard** itself.
  - The anticipated date of initial operational capability.
  - The anticipated duration of operational service.
- Specific service arrangements shall be made via memoranda of agreement. Neither this **Recommended Standard** nor any ensuing **standard** is a substitute for a memorandum of agreement.

No later than five years from its date of issuance, this **Recommended Standard** will be reviewed by the CCSDS to determine whether it should: (1) remain in effect without change; (2) be changed to reflect the impact of new technologies, new requirements, or new directions; or (3) be retired or canceled.

In those instances when a new version of a **Recommended Standard** is issued, existing CCSDS-related member standards and implementations are not negated or deemed to be non-CCSDS compatible. It is the responsibility of each member to determine when such standards or implementations are to be modified. Each member is, however, strongly encouraged to direct planning for its new standards and implementations towards the later version of the Recommended Standard.

FOREWORD

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CCSDS has processes for identifying patent issues and for securing from the patent holder agreement that all licensing policies are reasonable and non-discriminatory. However, CCSDS does not have a patent law staff, and CCSDS shall not be held responsible for identifying any or all such patent rights.

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommended Standard is therefore subject to CCSDS document management and change control procedures, which are defined in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4). Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/

Questions relating to the contents or status of this document should be sent to the CCSDS Secretariat at the email address indicated on page i.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- China National Space Administration (CNSA)/People's Republic of China.
- Deutsches Zentrum für Luft- und Raumfahrt (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Federal Space Agency (FSA)/Russian Federation.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.
- UK Space Agency/United Kingdom.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Belgian Federal Science Policy Office (BFSPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
- Chinese Academy of Sciences (CAS)/China.
- China Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Danish National Space Center (DNSC)/Denmark.
- Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
- Electronics and Telecommunications Research Institute (ETRI)/Korea.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
- Hellenic National Space Committee (HNSC)/Greece.
- Hellenic Space Agency (HSA)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- Mohammed Bin Rashid Space Centre (MBRSC)/United Arab Emirates.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Agency of the Republic of Kazakhstan (NSARK)/Kazakhstan.
- National Space Organization (NSPO)/Chinese Taipei.
- Naval Center for Space Technology (NCST)/USA.
- Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
- South African National Space Agency (SANSA)/Republic of South Africa.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- Swiss Space Office (SSO)/Switzerland.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 355.1-B-1	Space Data Link Security Protocol—Extended Procedures, Recommended Standard, Issue 1	February 2020	Original issue

CONTENTS

CONTENTS (continued)

Section Page

Figure

CONTENTS (continued)

Section Page

Table

1 INTRODUCTION

1.1 PURPOSE

The purpose of this Recommended Standard is to specify the Space Data Link Security (SDLS) Protocol Extended Procedures (EP). It defines the Key Management, Security Association Management, SDLS Monitoring and Control Services, and data structures required to operate the SDLS protocol over a space link. Further, it defines the interfaces and required data structures for proper interaction with the Space Data Link (SDL) protocols and a security function status reporting mechanism.

1.2 SCOPE

This Recommended Standard defines the SDLS Extended Procedures in terms of

a) the protocol data units exchanged between the service initiator and service recipient;

b) the procedures performed by the service initiator and service recipient; and

c) the interfaces with the SDLS and SDL protocols.

It does not specify

a) individual implementations or products;

b) the implementation of service interfaces within real systems;

c) the methods or technologies required to perform the procedures; or

d) the space-link security protocol itself.

This Recommended Standard does not mandate the use of any particular cryptographic algorithm for key generation and management. CCSDS Cryptographic Algorithms (reference [7]) are to be considered for this purpose.

1.3 APPLICABILITY

This Recommended Standard applies to the creation of Agency standards and to data communications over space links between CCSDS Agencies in cross-support situations. The Recommended Standard includes comprehensive specification of the service for inter-Agency cross support. It is neither a specification of, nor a design, for real systems that may be implemented for existing or future missions.

The Recommended Standard specified in this document is to be invoked through the normal standards programs of each CCSDS Agency and is applicable to those missions for which cross support based on capabilities described in this Recommended Standard is anticipated. Where mandatory capabilities are clearly indicated in sections of the Recommended Standard, they must be implemented when this document is used as a basis for cross support.

Where options are allowed or implied, implementation of these options is subject to specific bilateral cross support agreements between the Agencies involved.

1.4 RATIONALE

The goals of this Recommended Standard are to

a) provide standard specifications for the Extended Procedures required to operate the SDLS protocol (reference [1]), in particular:
   - Key Management,
   - Security Associations Management, and
   - SDLS Monitoring and Control;

b) specify a new type of telemetry frame Operational Control Field (OCF), the Frame Security Report (FSR), for reporting of link security status events, fully compatible with the existing SDL protocols (TM, AOS, and USLP) (references [4], [5], and [9]); and

c) facilitate the development of common commercial implementations to improve interoperability across agencies.

More discussion of the goals of the SDLS Extended Procedures and design choices, including its interaction with other CCSDS services, may be found in reference [**C**13].

1.5 DOCUMENT STRUCTURE

Section 1 presents the purpose, scope, applicability, and rationale of this Recommended Standard and lists the conventions, definitions, and references used throughout the document.

Section 2 (informative) provides an overview of the Space Data Link Security Protocol Extended Procedures.

Section 3 (normative) defines the services provided by the protocol entity.

Section 4 (normative) specifies the interfaces between the SDLS Extended Procedures and the SDL and SDLS protocols.

Section 5 (normative) specifies the protocol data units provided for these services and the procedures employed by the service provider.

Section 6 (normative) specifies the managed parameters.

Section 7 (normative) specifies how to verify an implementation’s conformance with the Security Protocol.

Annex A (normative) provides a Protocol Implementation Conformance Statement (PICS) proforma for the Security Protocol.

Annex B (informative) provides an overview of security, SANA registry, and patent considerations related to this Recommended Standard.

Annex C (informative) provides a list of informative references.

Annex D (informative) defines baseline implementations suitable for a large range of space missions.

1.6 DEFINITIONS

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

Generic definitions for the security terminology applicable to this and other CCSDS documents are provided under reference [3].

Initiator: The Initiator of an SDLS Extended Procedure is one of the two peers involved in an SDLS communication session. The Initiator manages the SDLS session parameters, provides the necessary resources to execute a procedure, and always initiates a procedure.

Recipient: The Recipient of an SDLS Extended Procedure is one of the two peers involved in an SDLS communication session. It configures the SDLS session parameters based on procedures and instructions initiated by the Initiator.

Anti-Replay Sequence Number, ARSN: A counter field initialized to zero when a Security Association is activated between an Initiator and a Recipient, and then incremented for each transfer frame sent using that SA. This is used to provide protection against replay attacks.

NOTE – In most cases, the Initiator is the Mission Operations Center (MOC), and the Recipient is the spacecraft.

1.7 CONVENTIONS

1.7.1 NOMENCLATURE

The following conventions apply for the normative specifications in this Recommended Standard:

a) the words ‘shall’ and ‘must’ imply a binding and verifiable specification;

b) the word ‘should’ implies an optional, but desirable, specification;

c) the word ‘may’ implies an optional specification;

d) the words ‘is’, ‘are’, and ‘will’ imply statements of fact.

NOTE – These conventions do not imply constraints on diction in text that is clearly informative in nature.

1.7.2 INFORMATIVE TEXT

In the normative sections of this document, informative text is set off from the normative specifications either in notes or under one of the following subsection headings:

- Overview;
- Background;
- Rationale;
- Discussion.

1.8 REFERENCES

The following publications contain provisions, which, through reference in this text, constitute provisions of this document. At the time of publication, the editions indicated were valid. All publications are subject to revision, and users of this Recommended Standard are encouraged to investigate the possibility of applying the most recent editions of the publications indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS publications.

[1] *Space Data Link Security Protocol*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 355.0-B-1. Washington, D.C.: CCSDS, September 2015.

[2] *Symmetric Key Management*. Issue 1. Draft Recommendation for Space Data System Practices (Red Book), CCSDS 354.0-R-1. Washington, D.C.: CCSDS, June 2018.

[3] *Information Security Glossary of Terms*. Issue 2. Recommendation for Space Data System Practices (Magenta Book), CCSDS 350.8-M-2. Washington, D.C.: CCSDS, February 2020.

[4] *TM Space Data Link Protocol*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 132.0-B-2. Washington, D.C.: CCSDS, September 2015.

[5] *AOS Space Data Link Protocol*. Issue 3. Recommendation for Space Data System Standards (Blue Book), CCSDS 732.0-B-3. Washington, D.C.: CCSDS, September 2015.

[6] *TC Space Data Link Protocol*. Issue 3. Recommendation for Space Data System Standards (Blue Book), CCSDS 232.0-B-3. Washington, D.C.: CCSDS, September 2015.

[7] *CCSDS Cryptographic Algorithms*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 352.0-B-2. Washington, D.C.: CCSDS, August 2019.

[8] *Space Packet Protocol*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 133.0-B-1. Washington, D.C.: CCSDS, September 2003.

[9] *Unified Space Data Link Protocol*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 732.1-B-1. Washington, D.C.: CCSDS, October 2018.

NOTE – Informative references are listed in annex C.

2 OVERVIEW

2.1 CONCEPT OF SDLS EXTENDED PROCEDURES

The SDLS protocol (reference [1]) provides a structured method for applying data authentication and/or data confidentiality to the contents of TM, AOS, USLP, or TC Transfer Frames used by the Space Data Link Protocols over a space link.

The purpose of the SDLS Extended Procedures is to provide a standardized set of auxiliary services that are necessary to operate an implementation of the SDLS protocol. These services are categorized into Key Management, Security Association (SA) Management, and SDLS Monitoring & Control.

2.2 FEATURES OF SDLS EXTENDED PROCEDURES

The SDLS Extended Procedures specify, for each of the services provided, the following features:

- procedures description and breakdown (actions to be performed by Initiator and Recipient);
- data formats of the information exchanged as part of the procedures;
- means for signaling these procedures using existing, or extended, SDL protocol fields or features.

In addition, the SDLS Extended Procedures specify an FSR that is reported as an Operational Control Field within TM, AOS, or USLP frames and contains a brief report on the status of the Recipient Security Function.

2.3 SERVICES PROVIDED BY SDLS EXTENDED PROCEDURES

2.3.1 GENERAL

The SDLS Extended Procedures provide three different services:

a) Key Management Service;

b) Security Association Management Service; and

c) SDLS Monitoring & Control Service.

2.3.2 KEY MANAGEMENT SERVICE

The Key Management Service defines the concrete protocols and procedures to implement a subset of the abstract Key Management Procedures that are documented in the Symmetric

Key Management Recommended Practice (reference [2]). Thus terminology from that recommended practice is used within this specification.

The Key Management Services are designed to support a symmetric key management infrastructure for secure communications (authentication, confidentiality, and integrity) using the SDLS protocol. Key Management is a necessity to ensure that both communication end points (i.e., the Initiator and the Recipient) are synchronized in terms of cryptographic keys and key states. The Initiator and the Recipient share a common set of keys for all communication links between them.

A number of different key management infrastructure designs exist; however, for reasons of interoperability, scalability, security, and reduced complexity, the SDLS Key Management Services baseline is built around a two-tier symmetric key infrastructure, consisting of a master key (also called static key or key encryption key) tier and a session key (also called traffic protection key) tier. Master keys are used exclusively for the purpose of management (with few notable exceptions), while session keys are used to support the actual SDLS cryptographic operations; that is, they are traffic keys. The concept of Over-The-Air-Rekeying (OTAR) is used to ensure frequent updates of the session keys by distributing new session keys that have been generated by the Initiator to the Recipient.

The Key Management Procedures are built around a cryptographic key lifecycle, as shown in figure 2-1. The lifecycle is a state machine and a simplified implementation of the full key management lifecycle as specified in the CCSDS Symmetric Key Management Recommended Practice (reference [2]). While OTAR is recommended to be implemented, a space mission could also fly with pre-loaded keys only. In this case, OTAR is not required.

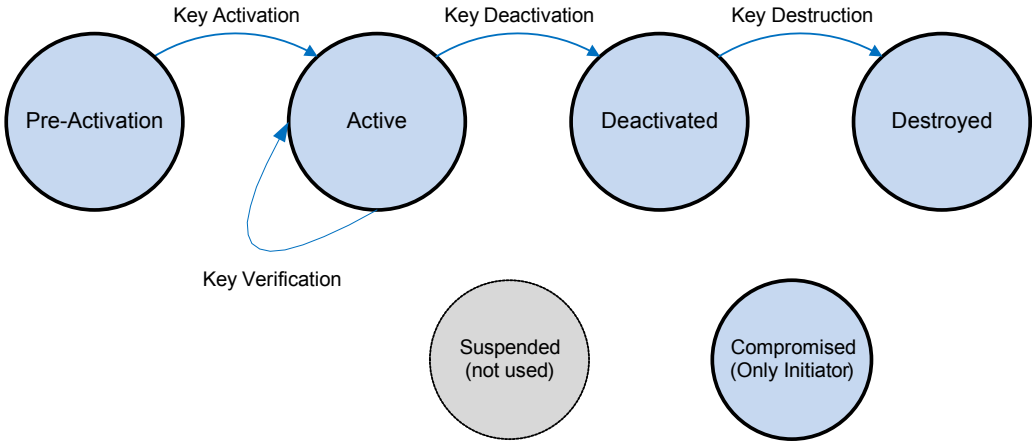

Figure 2-1: SDLS Extended Procedures Cryptographic Key Lifecycle

NOTE – The Suspended state as specified in reference [2] is not used in the SDLS Extended Procedures Recommended Standard. It is shown in grey in figure 2-1. The Compromised state is only applicable to the Initiator. There are no procedures associated with that state.

2.3.3 SECURITY ASSOCIATION MANAGEMENT SERVICE

The SDLS protocol provides encryption, authentication, or authenticated encryption for data link layer services of the TC, TM, AOS, and USLP protocols. The Security Association Management Service for the SDLS protocol is designed to carry out the most basic functions of Security Association setup, activation, status, and control necessary to command the configurable Security Association parameters of a remote system's SDLS implementation into a state suitable for operations.

The SA Management Service is designed to support an operational state model that may be simple or complex, as mission needs indicate. Many missions of ordinary duration and lower data rates can be satisfied with support for statically defined Security Associations and pre-loaded cryptographic keys and algorithms. For these, it is sufficient to choose which SA to use on a particular virtual channel along with all of its pre-loaded attributes.

It is anticipated that future complex or long-duration missions may need the capability to reuse and/or reconfigure Security Associations as the SAs and keys loaded into the system prior to the mission are used up over time. For this reason, the SA Management Service state model includes optional directives supporting OTAR and even on-demand instantiation of Security Associations.

Figure 2-2 illustrates the state model for Security Associations.

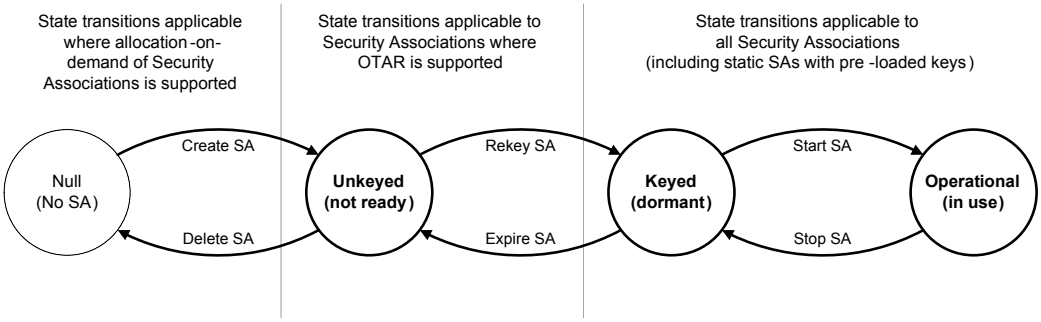

Figure 2-2: Variable State Model for Security Association Management

2.3.4 SDLS MONITORING & CONTROL SERVICE

The SDLS Monitoring and Control Service is designed to support the SDLS Monitoring and control of the Recipient Security Function. This is done via a set of messages sent to the security function (Commands) or received from it (Reports). These messages allow for complete control/command of the Recipient Security Function. A processor that implements the recipient security function may be a completely separate piece of equipment or a piece of software providing security functions, as defined in the SDLS standard, applied to one communication link: TC, TM, AOS, or USLP.

2.4 FRAME SECURITY REPORT

The SDLS Extended Procedures also specify a new type of Operational Control Field for the Space Data Link Layer protocols. This FSR contains information about the status of the security function and about the security processing (e.g., indicating a recent authentication failure). Since a TM, AOS, or USLP frame cannot contain two OCFs at the same time, the FSR insertion is multiplexed with the insertion of Command Link Control Word (CLCW). The multiplexing pattern is mission specific.

3 SERVICE DEFINITION

3.1 OVERVIEW

This section provides the service definition for the SDLS Extended Procedures.

The services provided by the SDLS Extended Procedures are defined as procedures with sequential execution steps to be executed by the Initiator or the Recipient. These procedural execution steps are independent of specific implementations. Usually, the Initiator is the operational control center while the Recipient is the spacecraft.

The parameters used by the procedures are specified in an abstract sense. They specify the information associated with a particular procedure step and are passed in either direction between the Initiator and the Recipient. The way in which those parameters are exchanged between the Initiator and the Recipient is specified in section 5 of this document.

This section defines the Key Management Service, the Security Association Management Service, and the SDLS Monitoring & Control Service.

3.2 KEY MANAGEMENT SERVICE

3.2.1 OVERVIEW

The Key Management Service Procedures specified in this section are an SDLS-specific instantiation of the abstract Key Management Procedures specified in reference [2].

The following SDLS-specific service procedures are defined:

- OTAR;
- Key Activation;
- Key Deactivation;
- Key Destruction;
- Key Verification; and
- Key Inventory.

3.2.2 SERVICE PARAMETERS

3.2.2.1 Over-the-Air-Rekeying

The OTAR procedure shall have the following service parameters:

a) Key ID of the Master Key used for the OTAR command protection;

b) Initialization Vector (IV) for the Protected Set of Upload Session Keys;

c) Set of Upload Session Keys;

d) Protected Set of Upload Session Keys, containing for each session key to be uploaded:

   1) Session Key ID,

   2) Encrypted Key;

NOTE – The length of the session keys is mission specific.

e) MAC of the Protected Set of Upload Session Keys.

NOTES

1 The actual possible identifier values for the Session Key ID are mission specific.

2 The actual number of session keys is indicated by the length field of the Tag, Length, Value (TLV) format (see 5.3.1.2).

3.2.2.2 Key Activation

The Key Activation procedure shall have the following service parameter:

Set of Key IDs.

NOTES

1 The actual possible identifier values for the Key ID are mission specific.

2 The actual number of keys to be activated is indicated by the length field of the TLV format.

3.2.2.3 Key Deactivation

The Key Deactivation procedure shall have the following service parameter:

Set of Key IDs.

NOTES

1 The actual possible identifier values for the Key ID are mission specific.

2 The actual number of keys to be deactivated is indicated by the length field of the TLV format.

3.2.2.4 Key Destruction

The Key Destruction procedure shall have the following service parameter:

Set of Key IDs.

NOTES

1 The actual possible identifier values for the Key ID are mission specific.

2 The actual number of keys to be destroyed is indicated by the length field of the TLV format.

3.2.2.5 Key Verification

The Key Verification procedure shall have the following service parameters:

a) Set of Key IDs;

b) Set of Challenges;

c) Set of Challenge Responses (Session Key ID, Session Key IV, Response Encrypted Challenge, Response MAC).

NOTE – The actual number of session keys to be verified is defined through the Number of Session Keys parameter and indicated by the length field of the TLV format.

3.2.2.6 Key Inventory

The Key Inventory procedure shall have the following service parameters:

a) Range of Key IDs (First Key ID in range, Last Key ID in range);

b) Total number of Key IDs returned;

c) Set of Key state Responses (Key ID, Key state).

NOTE – The total number of keys in the specified range at the Recipient is indicated by the ‘total number of Key IDs returned’ field in the reply PDU.

3.2.3 SERVICE PROCEDURES

3.2.3.1 Over-the-Air-Rekeying

3.2.3.1.1 Overview

OTAR realizes the secure (encrypted and authenticated) transmission of new session keys over a communication channel from the Initiator to the Recipient. The implementation of the installation of the keys on the Recipient side is mission specific and not addressed by this Recommended Standard.

3.2.3.1.2 Preconditions for the Procedure

3.2.3.1.2.1 The Initiator shall have a set of session keys in pre-activation state available.

3.2.3.1.2.2 Both entities shall have an identical master key in pre-activation or active state.

NOTE – This is the master key that will be used to ensure authenticity and confidentiality of the session keys during transmission from the Initiator to the Recipient.

3.2.3.1.3 Procedural Steps

3.2.3.1.3.1 General

The OTAR procedure shall include the following mandatory execution steps:

a) Protection of Set of Upload Session Keys; Role: Initiator;

b) Signaling of Protected Set of Upload Session Keys; Role: Initiator;

c) Processing of Protected Set of Upload Session Keys; Role: Recipient.

3.2.3.1.3.2 Protection of Set of Upload Session Keys

The Protection of Set of Upload Session Keys step shall

a) be executed by the Initiator;

b) have the following inputs:

   1) Set of Upload Session Keys,

   2) Key ID of the Master Key;

c) have the following outputs:

   1) Protected Set of Upload Session Keys ready for upload,

2) Master Key in Active State; and

d) execute the following:

1) the State of the master key identified by the key ID of the master key shall be transitioned to active state if the master key is not already in active state,

2) authenticated encryption under the selected master key shall be applied to the complete set of pairs (Key ID, Key) to create the Protected Set of Upload Session Keys:

i) this shall be done using the agreed cryptographic algorithm and the master key identified by the master key ID,

ii) the initialization vector (if applicable) and MAC parameters shall be populated accordingly.

3.2.3.1.3.3 Signaling of Protected Set of Upload Session Keys

The Signaling of Protected Set of Upload Session Keys step shall

a) be executed by the Initiator;

b) have the following inputs:

1) Protected Set of Upload Session Keys,

2) Key ID of the Master Key;

c) have the following output: Protected Set of Upload Session Keys and the Key ID of the Master Key transmitted to the Recipient; and

d) execute the following: an OTAR Command PDU, as defined in 5.4.2.1, shall be created and transmitted to the Recipient using the interface specified in section 4.

3.2.3.1.3.4 Processing of Protected Set of Upload Session Keys

The Processing of Protected Set of Upload Session Keys step shall

a) be executed by the Recipient;

b) have the following inputs:

1) the Protected Set of Upload Session Keys, and

2) the Key ID of the Master Key received from the Initiator;

c) have the following output: Decrypted Set of Upload Session Keys stored in Pre-Active state; and

d) execute the following:

1) the Recipient shall perform the authentication and decryption of the Protected Set of Upload Session Keys using the Initialization Vector and MAC parameters as input to the authentication algorithm execution under the master key identified by the Master Key Id,

2) for each decrypted Upload Key, the Recipient shall store it in Pre-Active state using the indicated Key ID.

NOTE – Execution of this procedure will typically result in existing keys stored in the Recipient memory slots to be replaced by newly uploaded keys. Proper management of the on-board key memory is not the subject of this Recommended Standard and is mission specific.

3.2.3.2 Key Activation

3.2.3.2.1 Overview

The Key Activation procedure activates a set of keys at both ends of the communication channel (Initiator & Recipient) so that these keys are assigned the Active State and subsequently can be used for cryptographic operations.

3.2.3.2.2 Preconditions for the Procedure

Both entities shall have an identical set of keys in pre-activation state.

NOTE – A subset of these pre-active keys is activated by this procedure.

3.2.3.2.3 Procedural Steps

3.2.3.2.3.1 General

The Key Activation procedure shall include the following mandatory execution steps:

a) Activation of Initiator Keys; Role: Initiator;

b) Signaling of Key IDs for Keys to be activated; Role: Initiator;

c) Activation of Recipient Keys; Role: Recipient.

3.2.3.2.3.2 Activation of Initiator Session Keys

The Activation of Initiator Session Keys step shall

a) be executed by the Initiator;

b) have the following input: Set of Key IDs;

c) have the following output: all keys identified by the Set of Key IDs in State Activated; and

NOTE – Subsection 5.4.1.2 and reference [2] provide more information on key states.

d) execute the following: Keys identified by set of Key IDs shall be transitioned from Pre-Active State to Active State.

3.2.3.2.3.3 Signaling of Keys to Be Activated

The Signaling of Keys to be Activated step shall

a) be executed by the Initiator;

b) have the following input: Set of Key IDs of keys activated in Step 3.2.3.2.3.2;

c) have the following output: the Set of Key IDs of keys activated in Step 3.2.3.2.3.2 transmitted to the Recipient; and

NOTE – The signaling uses the interface to the SLP as described in section 4.

d) execute the following: a Key Activation Command PDU, as defined in 5.4.2.2, shall be created and transmitted to the Recipient using the interface specified in section 4.

3.2.3.2.3.4 Activation of Recipient Session Keys

The Activation of Recipient Session Keys step shall

a) be executed by the Recipient;

b) have the following input: the Set of Key IDs of keys activated in Step 3.2.3.2.3.2 received from the Initiator;

c) have the following output: all session keys identified by the set of key IDs in State Active; and

d) execute the following: the keys identified by the Key IDs in the set of Key IDs shall be transitioned from Pre-Active State to Active State.

3.2.3.3 Key Deactivation

3.2.3.3.1 Overview

The Key Deactivation (or revocation) procedure deactivates a set of previously uploaded keys at both ends of the communication channel (Initiator & Recipient) so that these keys are assigned the Deactivated State and subsequently cannot be used for cryptographic operations anymore. The keys are not destroyed (erased) by this procedure but can be used only to decrypt formerly encrypted data.

3.2.3.3.2 Preconditions for the Procedure

Both entities shall have an identical set of keys in active state.

NOTE – A subset of these active keys is revoked by this procedure.

3.2.3.3.3 Procedural Steps

3.2.3.3.3.1 General

The Key Deactivation procedure shall include the following mandatory execution steps:

a) Deactivation of Initiator keys; Role: Initiator;

b) Signaling of Key IDs of the keys to be deactivated; Role: Initiator;

c) Deactivation of Recipient keys; Role: Recipient.

3.2.3.3.3.2 Deactivation of Initiator Keys

The Deactivation of Initiator Keys step shall

a) be executed by the Initiator;

b) have the following input: the set of key IDs of keys to be deactivated;

c) have the following output: all keys identified by the set of key IDs in State Deactivated; and

NOTE – Subsection 5.4.1.2 and reference [2] provide more information on key states.

d) execute the following: the keys identified by the Key IDs in the set of Key IDs shall be transitioned from Active State to Deactivated State.

3.2.3.3.3.3 Signaling of Keys to be deactivated

The Signaling of Keys to be deactivated step shall

a) be executed by the Initiator.

b) have the following input: the set of Key IDs of keys deactivated in Step 3.2.3.3.3.2;

c) have the following outputs: the set of Key IDs of keys deactivated in Step 3.2.3.3.3.2 transmitted to the Recipient; and

NOTE – The signaling uses the interface to the SLP as described in section 4.

d) execute the following: a Key Deactivation Command PDU, as defined in 5.4.2.3, shall be created and transmitted to the Recipient using the interface specified in section 4.

3.2.3.3.3.4 Deactivation of Recipient Keys

The Deactivation of Recipient Keys step shall

a) be executed by the Recipient;

b) have the following input: the set of Key IDs of keys deactivated in Step 3.2.3.3.3.2 received from the Initiator;

c) have the following output: all keys identified by the set of key IDs in State Deactivated;

d) execute the following: the keys identified by the Key IDs in the set of Key IDs shall be transitioned from Active State to Deactivated State.

3.2.3.4 Key Destruction

3.2.3.4.1 Overview

The Key Destruction deletes a number of keys from both Initiator and Recipient key databases. No record of the destroyed keys is retained. After keys are destroyed they can no longer be used even to decrypt formerly encrypted data (see 3.2.3.3.1).

3.2.3.4.2 Preconditions for the Procedure

Both entities shall have an identical set of keys in deactivated state.

NOTE – A subset of these deactivated keys is deleted/destroyed by this procedure.

3.2.3.4.3 Procedural Steps

3.2.3.4.3.1 General

The Key Destruction procedure shall include the following mandatory execution steps:

a) Destruction of Initiator session keys; Role: Initiator;

b) Signaling of session Key IDs to be destroyed; Role: Initiator;

c) Destruction of Recipient session keys; Role: Recipient.

3.2.3.4.3.2 Destruction of Initiator Session Keys

The Destruction of Initiator Session Keys step shall

a) be executed by the Initiator;

b) have the following input: the set of key IDs of keys to be destroyed;

c) have the following output: all keys identified by the set of key IDs in State Destroyed; and

NOTE – Subsection 5.4.1.2 and reference [2] provide more information on key states.

d) execute the following: the session keys identified by the Key IDs in the set of Key IDs shall be transitioned from Deactivated State to Destroyed State.

3.2.3.4.3.3 Signaling of Keys to Be Destroyed

The Signaling of Keys to Be Destroyed step shall

a) be executed by the Initiator;

b) have the following input: the set of Key IDs of keys destroyed in Step 3.2.3.4.3.2;

c) have the following output: the set of Key IDs of keys destroyed in Step 3.2.3.4.3.2 transmitted to the Recipient; and

NOTE – The signaling uses the interface to the SLP as described in section 4.

d) execute the following: a Key Destruction Command PDU, as defined in 5.4.2.4, shall be created and transmitted to the Recipient using the interface specified in section 4.

3.2.3.4.3.4 Destruction of Recipient Session Keys

The Destruction of Recipient Session Keys step shall

a) be executed by the Recipient;

b) have the following input: the set of Key IDs of keys destroyed in Step 3.2.3.4.3.2 received from the Initiator;

NOTE – The signaling uses the interface to the SLP as described in section 4.

c) have the following output: all keys identified by the set of key IDs in State Destroyed; and

d) execute the following: the session keys identified by the Key IDs in the set of Key IDs shall be transitioned from Deactivated State to Destroyed State.

3.2.3.5 Key Verification

3.2.3.5.1 Overview

The Key Verification procedure allows the verification of a set of active keys at the Recipient. This gives confirmation to the Initiator that the keys are not corrupted or modified and are fully operational.

3.2.3.5.2 Preconditions for the Procedure

Both entities shall have an identical set of keys in Active State.

NOTE – A subset of these keys is verified by this procedure.

3.2.3.5.3 Procedural Steps

3.2.3.5.3.1 General

The Key Verification procedure shall include the following mandatory execution steps:

a) Challenge Creation; Role: Initiator;

b) Signaling of Challenges and Key IDs to be verified; Role: Initiator;

c) Computation of Challenge Responses; Role: Recipient;

d) Signaling of Challenge Responses; Role: Recipient;

e) Response Verification; Role: Initiator.

3.2.3.5.3.2 Challenge Creation

The Challenge Creation step shall

a) be executed by the Initiator;

b) have the following input: the Set of Key IDs of keys to be verified;

c) have the following output: the Set of Challenges corresponding to the number of keys to be verified;

d) execute the following: for each key in the set of Key IDs, a challenge shall be created in the Set of Challenges; each challenge shall be associated with a Key ID; and

e) The specification of the algorithm for the creation of the Challenges is outside the scope of this Recommended Standard; however, associated authentication algorithms shall be compliant with those approved in reference [7].

3.2.3.5.3.3 Signaling of Challenges and Key IDs to Be Verified

The Signaling of Challenges and Key IDs to Be Verified step shall

a) be executed by the Initiator;

b) have the following input: the set of Key IDs to be verified and the Set of Challenges created in Step 3.2.3.5.3.2;

c) have the following output: the Key IDs to be verified and the Set of Challenges transmitted to the Recipient; and

NOTE – The signaling uses the interface to the SLP as described in section 4.

d) execute the following: a Key Verification Command PDU, as defined in 5.4.2.5, shall be created and transmitted to the Recipient using the interface specified in section 4.

3.2.3.5.3.4 Computation of Challenge Responses

The Computation of Challenge Responses step shall

a) be executed by the Recipient;

b) have the following input: the Key IDs to be verified and the Set of Challenges received from the Initiator;

c) have the following output: the Set of Challenge Responses; and

d) execute the following: for each key in the set of Key IDs and each associated Challenge in the Set of Challenges, a response shall be created in the Set of Challenge Responses.

3.2.3.5.3.5 Signaling of Challenge Responses

The Signaling of Challenge Responses step shall

a) be executed by the Recipient;

b) have the following input: the set of Key IDs and the Set of Challenge Responses created in Step 3.2.3.5.3.4;

c) have the following output: the Key IDs and the Set of Responses transmitted to the Initiator; and

NOTE – The signaling uses the interface to the SLP as described in section 4.

d) execute the following: a Key Verification Reply PDU, as defined in 5.4.2.5, shall be created and transmitted to the Initiator using the interface specified in section 4.

3.2.3.5.3.6 Challenge Response Verification

The Challenge Response Verification step shall

a) be executed by the Initiator;

b) have the following input: the Key IDs and the Set of Challenge Responses transmitted to the Recipient;

c) have the following output: Verification results associated with each key in the Set of Key IDs; and

d) execute the following:

   1) for each key in the set of Key IDs and each associated response in the Set of Challenge Responses, the challenge shall be computed and compared with the associated challenge in the Set of Challenges,

   2) in case of a match, the Key shall be declared verified.

3.2.3.6 Key Inventory

3.2.3.6.1 Overview

The Key Inventory procedure allows the Initiator to get the list of Key IDs present at the Recipient together with the corresponding Key states. This enables the Initiator to

– check the synchronization of its Key data base with the Recipient; and

– have confirmation of the reception of keys and corresponding key states as a verification for the key management procedures.

3.2.3.6.2 Preconditions for the Procedure

There are no preconditions for the procedure.

3.2.3.6.3 Procedural Steps

3.2.3.6.3.1 General

The Key Inventory procedure shall include the following mandatory execution steps:

a) Signaling of the Key Inventory Request; Role: Initiator;

b) Generation of the Key Inventory Response; Role: Recipient;

c) Signaling of Key Inventory Response; Role: Recipient.

3.2.3.6.3.2 Signaling of the Key Inventory Request

The Signaling of the Key Inventory Request step shall

a) be executed by the Initiator;

b) have the following input: range of Key IDs (first Key ID in range, last Key ID in range) to be inventoried;

c) have the following output: the Key Inventory Request transmitted to the Recipient; and

d) execute the following: a Key Inventory Request Command PDU, as defined in 5.4.2.6, shall be created and transmitted to the Recipient using the interface specified in section.

3.2.3.6.3.3 Generation of the Key Inventory Response

The Generation of the Key Inventory Response step shall

a) be executed by the Recipient;

b) have the following input: the range of Key IDs to be inventoried;

c) have the following output: the Key Inventory Response;

d) execute the following:

1) compute the total number of Keys, in the specified Key ID range, present in the Recipient Key data base;

2) for each Key in the range, extract corresponding Key State;

3) generate the Key Inventory Response comprising the above-mentioned information: total number of keys in specified range (Key ID, Key State).

3.2.3.6.3.4 Signaling of Key Inventory Response

The Signaling of the Key Inventory Response step shall

a) be executed by the Recipient;

b) have the following input: the Key Inventory Response created at Step 3.2.3.6.3.3;

c) have the following output: Key Inventory Response transmitted to the Initiator; and

d) execute the following: a Key Inventory Reply PDU, as defined in 5.4.2.6, shall be created and transmitted to the Initiator using the interface specified in section 4.

3.3 SECURITY ASSOCIATION MANAGEMENT SERVICE

3.3.1 OVERVIEW

The Security Association Management Service establishes the context of a Security Association for a particular Global Virtual Channel and/or Global MAP ID or a set of Global Virtual Channel IDs (GVCIDs) or Global MAP IDs (GMAP IDs). The user manages the operations of a Security Association by invoking the service primitives defined below.

The following service procedures are specified:

- Start SA;
- Stop SA;
- Rekey SA;
- Expire SA;
- Create SA;
- Delete SA;
- Set Anti-Replay Sequence Number;
- Set Anti-Replay Sequence Number window;
- SA Status Request;
- Read Anti-Replay Sequence Number.

3.3.2 SERVICE PARAMETERS

3.3.2.1 Start SA

The Start SA directive shall have the following service parameters:

a) Security Parameter Index (SPI);

b) Global Virtual Channel (GVC)/GMAP ID(s) with which the SA is to be used.

3.3.2.2 Stop SA

The Stop SA directive shall have the following service parameter:

SPI.

3.3.2.3 Rekey SA

The Rekey SA directive shall have the following service parameters:

a) SPI;

b) Anti-Replay Sequence Number (ARSN);

c) IV;

d) Encryption Key ID;

e) Authentication Key ID.

3.3.2.4 Expire SA

The Expire SA directive shall have the following service parameter:

SPI.

3.3.2.5 Create SA

The Create SA directive shall have the following service parameters:

a) SPI;

b) SA Service Type;

c) Lengths for Security Header IV, SN, and PL fields;

d) Length for Security Trailer MAC field;

e) Encryption cipher suite length and identifier;

f) IV length and initial value;

g) Authentication cipher suite length and identifier;

h) Authentication bit mask length and value;

i) ARSN length and initial value;

j) ARSN window length and value.

3.3.2.6 Delete SA

The Delete SA directive shall have the following service parameter:

SPI.

3.3.2.7 Set Anti-Replay Sequence Number

The Set Anti-Replay Sequence Number directive shall have the following service parameters:

a) SPI;

b) Anti-Replay Sequence Number.

3.3.2.8 Set Anti-Replay Sequence Number Window

The Set Anti-Replay Sequence Number Window directive shall have the following service parameters:

a) SPI;

b) Anti-Replay Sequence Number window value.

3.3.2.9 SA Status Request

The SA Status Request directive shall have the following service parameters:

a) SPI;

b) procedure Identification of most recent SA state transition directive.

3.3.2.10 Read Anti-Replay Sequence Number

The Read Anti-Replay Sequence Number procedure shall have the following service parameters:

a) Security Parameter Index (16 bit);

NOTE – This is the SPI of the SA to which the ARSN belongs.

b) Anti-Replay Sequence Number (bit field length managed by the SA).

NOTE – The length of the ARSN is a managed parameter within the SA. It can vary; however, the full value of the ARSN must be reported without truncation.

3.3.3 SERVICE PROCEDURES

3.3.3.1 Start SA

3.3.3.1.1 Overview

The Start SA directive is used to begin using a particular Security Association on a channel.

3.3.3.1.2 Preconditions for the Procedure

The Security Association must be in the ‘Keyed’ state.

3.3.3.1.3 Procedural Steps

3.3.3.1.3.1 General

The Start SA procedure shall include the following mandatory execution steps:

a) Execution of Start SA; Role: Initiator;

b) Signaling of Start SA Request; Role: Initiator;

c) Execution of Start SA; Role: Recipient.

3.3.3.1.3.2 Execution of Start SA

The Execution of Start SA step shall

a) be executed by the Initiator;

b) have the following inputs:

   1) SPI of an existing Security Association which is in the ‘Keyed’ state,

   2) The specified GVC/GMAP ID(s) with which to use the SA;

c) have the following output: the SA transitions from ‘Keyed’ to ‘Operational’ state; and

d) execute the following:

   1) for each specified GVC/GMAP ID, verify that the SA is applicable and/or authorized for use,

   2) add the GVC/GMAP ID(s) into the SA.

3.3.3.1.3.3 Signaling of Start SA Request

The Signaling of Start SA Request step shall

a) be executed by the Initiator;

b) have the following inputs:

   1) The SPI of the Security Association to activate,

   2) The GVC/GMAP ID(s) upon which to activate the SA;

c) have the following output: SPI and specified GVC/GMAP ID(s) transmitted to the Recipient; and

d) execute the following: a Start SA PDU, as defined in 5.5.1.2, shall be created and transmitted to the Recipient using the interface specified in section 4.

3.3.3.1.3.4 Execution of Start SA

The Execution of Start SA step shall

a) be executed by the Recipient;

b) have the following input: SPI and specified GVC/GMAP ID(s) received from the Initiator;

c) have the following output: the SA transitions from 'Keyed' to 'Operational' state; and

d) execute the following:

1) verify that the specified SA exists and is in the 'Keyed' state,

2) for each specified GVC/GMAP ID, verify that the SA is applicable / authorized for use,

3) add the GVC/GMAP ID(s) into the SA.

3.3.3.2 Stop SA

3.3.3.2.1 Overview

The Stop SA directive is used to stop using a particular Security Association on a channel.

3.3.3.2.2 Preconditions for the Procedure

The Security Association must be in the 'Operational' state.

3.3.3.2.3 Procedural Steps

3.3.3.2.3.1 General

The Stop SA procedure shall include the following mandatory execution steps:

a) Execution of Stop SA; Role: Initiator;

b) Signaling of Stop SA Request; Role: Initiator;

c) Execution of Stop SA; Role: Recipient.

3.3.3.2.3.2 Execution of Stop SA

The Execution of Stop SA step shall

a) be executed by the Initiator;

b) have the following input: SPI of an existing Security Association that is in the 'Operational' state;

c) have the following output: the SA transitions from 'Operational' to 'Keyed' state;

d) execute the following:

   1) verify that the specified SA exists and is in the 'Operational' state,

   2) remove all GVC/GMAP ID(s) from the SA.

3.3.3.2.3.3 Signaling of Stop SA Request

The Signaling of Stop SA Request step shall

a) be executed by the Initiator.

b) have the following input: the SPI of the Security Association to stop;

c) have the following output: SPI transmitted to the Recipient; and

d) execute the following: a Stop SA PDU, as defined in 5.5.1.3, shall be created and transmitted to the Recipient using the interface specified in section 4.

3.3.3.2.3.4 Execution of Stop SA

The Execution of Stop SA step shall

a) be executed by the Recipient;

b) have the following input: SPI received from the Initiator;

c) have the following output: the SA transitions from 'Operational' to 'Keyed' state; and

d) execute the following:

   1) verify that the specified SA exists and is in the 'Operational' state,

   2) remove all GVC/GMAP ID(s) from the SA.

3.3.3.3 Rekey SA

3.3.3.3.1 Overview

The Rekey SA directive is used to associate a cryptographic key with a particular Security Association prior to the Security Association being activated for use on a channel.

3.3.3.3.2 Preconditions for the Procedure

The Security Association shall be in the ‘Unkeyed’ state, and the new key shall be in the Active State.

3.3.3.3.3 Procedural Steps

3.3.3.3.3.1 General

The Rekey SA procedure shall include the following mandatory execution steps:

a) Execution of Rekey SA; Role: Initiator;

b) Signaling of Rekey SA Request; Role: Initiator;

c) Execution of Rekey SA; Role: Recipient.

3.3.3.3.3.2 Execution of Rekey SA

The Execution of Rekey SA step shall

a) be executed by the Initiator;

b) have the following inputs:

   1) SPI of an existing Security Association which is in the ‘Unkeyed’ state,

   2) ARSN of an existing Security Association that is in the ‘Unkeyed’ state,

   3) the specified key ID(s),

   4) corresponding IV(s) to use;

c) have the following output: the SA transitions from ‘Unkeyed’ to ‘Keyed’ state; and

d) execute the following:

   1) verify that the specified SA exists and is in the ‘Unkeyed’ state,

   2) import the requested key(s) (identified by key ID) into the SA.

3.3.3.3.3.3 Signaling of Rekey SA Request

The Signaling of Rekey SA Request step shall

a) be executed by the Initiator;

b) have the following inputs:

   1) the SPI of the Security Association to rekey,

   2) the ARSN of the Security Association to rekey,

   3) the specified key ID(s),

   4) corresponding IV(s) to use;

c) have the following output: SPI transmitted to the Recipient; and

d) execute the following: a Rekey SA PDU, as defined in 5.5.1.4, shall be created and transmitted to the Recipient using the interface specified in section 4.

3.3.3.3.3.4 Execution of Rekey SA

The Execution of Rekey SA step shall

a) be executed by the Recipient;

b) have the following input: SPI received from the Initiator;

c) have the following output: the SA transitions from ‘Unkeyed’ to ‘Keyed’ state; and

d) execute the following:

   1) verify that the specified SA exists and is in the ‘Unkeyed’ state,

   2) import the requested key(s) (identified by key ID), ARSN, IV(s) into the SA.

3.3.3.4 Expire SA

3.3.3.4.1 Overview

The Expire SA directive is used to de-associate a cryptographic key from a particular Security Association in order that another key may be associated with that same SA using the ‘Rekey SA’ directive.

3.3.3.4.2 Preconditions for the Procedure

The Security Association must be in the ‘Keyed’ state.

3.3.3.4.3 Procedural Steps

3.3.3.4.3.1 General

The Expire SA procedure shall include the following mandatory execution steps:

a) Execution of Expire SA; Role: Initiator;

b) Signaling of Expire SA Request; Role: Initiator;

c) Execution of Expire SA; Role: Recipient.

3.3.3.4.3.2 Execution of Expire SA

The Execution of Expire SA step shall

a) be executed by the Initiator;

b) have the following input: SPI of an existing Security Association which is in the 'Keyed' state;

c) have the following output: the SA transitions from 'Keyed' to 'Unkeyed' state; and

d) execute the following: remove all keys from the SA.

3.3.3.4.3.3 Signaling of Expire SA Request.

The Signaling of Expire SA Request step shall

a) be executed by the Initiator;

b) have the following input: the SPI of the Security Association to expire;

c) have the following output: SPI transmitted to the Recipient; and

d) execute the following: an Expire SA PDU, as defined in 5.5.1.5.2, shall be created and transmitted to the Recipient using the interface specified in section 4.

3.3.3.4.3.4 Execution of Expire SA

The Execution of Expire SA step shall

a) be executed by the Recipient;

b) have the following input: SPI received from the Initiator;

c) have the following output: the SA transitions from 'Keyed' to 'Unkeyed' state; and

d) execute the following:

1) verify that the specified SA exists and is in the ‘Keyed’ state,
2) remove all keys from the SA.

3.3.3.5 Create SA

3.3.3.5.1 Overview

The Create SA directive is used to initialize a Security Association with the parameters supplied by the service user.

3.3.3.5.2 Preconditions for the Procedure

The Security Association must not exist.

3.3.3.5.3 Procedural Steps

3.3.3.5.3.1 General

The Create SA procedure shall include the following mandatory execution steps:

a) Execution of Create SA; Role: Initiator;
b) Signaling of Create SA Request; Role: Initiator;
c) Execution of Create SA; Role: Recipient.

3.3.3.5.3.2 Execution of Create SA

The Execution of Create SA step shall

a) be executed by the Initiator;
b) have the following input: SPI of a nonexistent Security Association;
c) have the following output: SA in the ‘Unkeyed’ state; and
d) execute the following:
    1) verify that the specified SA does not exist,
    2) initialize a Security Association (SA) having the specified SPI,
    3) add the SA Service Type into the SA,
    4) add the Lengths for Security Header IV, SN, and PL fields into the SA,
    5) add the Length for Security Trailer MAC field into the SA,

6) add the Encryption cipher suite identifier into the SA,
7) add the IV length and initial value into the SA,
8) add the Authentication cipher suite identifier into the SA,
9) add the Authentication bit mask length and value into the SA,
10) add the ARSN length and initial value into the SA,
11) add the Anti-Replay Sequence Number window length and value into the SA.

3.3.3.5.3.3 Signaling of Create SA Request

The Signaling of Create SA Request step shall

a) be executed by the Initiator;
b) have the following inputs:
    1) the SPI of the Security Association to create,
    2) the parameters of the Security Association to create:
        i) SA Service Type,
        ii) lengths for Security Header fields,
        iii) length for Security Trailer MAC field,
        iv) encryption cipher suite identifier,
        v) IV length and initial value,
        vi) authentication cipher suite identifier,
        vii) Authentication bit mask length and value,
        viii) ARSN length and initial value,
        ix) ARSN window length and value;
c) have the following output: SPI and SA parameters transmitted to the Recipient; and
d) execute the following: a Create SA PDU, as defined in 5.5.1.6, shall be created and transmitted to the Recipient using the interface specified in section 4.

3.3.3.5.3.4 Execution of Create SA

The Execution of Create SA step shall

a) be executed by the Recipient;

b) have the following input: fields in the Create SA PDU received from the Initiator;

c) have the following outputs: SA in the ‘Unkeyed’ state; and

d) execute the following:

1) verify that the specified SA does not exist,

2) initialize an SA having the specified SPI,

3) add the SA Service Type into the SA,

4) add the Lengths for Security Header IV, SN, and PL fields into the SA,

5) add the Length for Security Trailer MAC field into the SA,

6) add the Encryption cipher suite identifier into the SA,

7) add the IV length and initial value into the SA,

8) add the Authentication cipher suite identifier into the SA,

9) add the Authentication bit mask length and value into the SA,

10) add the ARSN length and initial value into the SA,

11) add the Anti-Replay Sequence Number window length and value into the SA.

3.3.3.6 Delete SA

3.3.3.6.1 Overview

The Delete SA directive is used to remove a Security Association entirely.

3.3.3.6.2 Preconditions for the Procedure

The Security Association must be in the ‘Unkeyed’ state.

3.3.3.6.3 Procedural steps

3.3.3.6.3.1 General

The Delete SA procedure shall include the following mandatory execution steps:

a) Execution of Delete SA Request; Role: Initiator;

b) Signaling of Delete SA Request; Role: Initiator;

c) Execution of Delete SA Request; Role: Recipient.

3.3.3.6.3.2 Execution of Delete SA Request (Initiator)

The Execution of Delete SA Request (Initiator) step shall

a) be executed by the Initiator;

b) have the following input: SPI of an existing Security Association that is in the 'Unkeyed' state;

c) have the following output: the SA transitions to a null state ('No SA');

d) execute the following: erase all managed parameters of the SA.

3.3.3.6.3.3 Signaling of Delete SA Request

The Signaling of Delete SA Request step shall

a) be executed by the Initiator;

b) have the following input: the SPI of the Security Association to delete;

c) have the following output: SPI transmitted to the Recipient; and

d) execute the following: a Delete SA PDU, as defined in 5.5.1.7, shall be created and transmitted to the Recipient using the interface specified in section 4.

3.3.3.6.3.4 Execution of Delete SA Request (Recipient)

The Execution of Delete SA Request (Recipient) step shall

a) be executed by the Recipient;

b) have the following input: SPI received from the Initiator;

c) have the following output: the SA transitions to a null state ('No SA'); and

d) execute the following:

   1) verify that the specified SA exists and is in the 'Unkeyed' state,

   2) erase all managed parameters of the SA.

3.3.3.7 Set Anti-Replay Sequence Number

3.3.3.7.1 Overview

The Set Anti-Replay Sequence Number directive is used to initialize the managed Anti-Replay Sequence Number for a Security Association to the value supplied by the service user.

3.3.3.7.2 Preconditions for the Procedure

The Security Association service type must be Authentication or Authenticated Encryption.

3.3.3.7.3 Procedural Steps

3.3.3.7.3.1 General

The Set Anti-Replay Sequence Number procedure shall include the following mandatory execution steps:

a) Execution of Set ARSN; Role: Initiator;

b) Signaling of Set ARSN Request; Role: Initiator;

c) Execution of Set ARSN; Role: Recipient.

3.3.3.7.3.2 Execution of Set ARSN

The Execution of Set ARSN step shall

a) be executed by the Initiator;

b) have the following inputs:

   1) SPI of an existing Security Association,

   2) requested new value for the managed Anti-Replay Sequence Number;

c) have the following output: Security Association with new Anti-Replay Sequence Number; and

d) execute the following: replace the current value of the managed Anti-Replay Sequence Number with the requested value.

3.3.3.7.3.3 Signaling of Set ARSN Request

The Signaling of Set ARSN Request step shall

a) be executed by the Initiator;

b) have the following inputs:

   1) the SPI of the Security Association,

   2) requested new value for the managed Anti-Replay Sequence Number;

c) have the following output: SPI and new ARSN value transmitted to the Recipient;

d) execute the following: a Set ARSN Command PDU, as defined in 5.5.1.8, shall be created and transmitted to the Recipient using the interface specified in section 4.

3.3.3.7.3.4 Execution of Set ARSN

The Execution of Set ARSN step shall

a) be executed by the Recipient;

b) have the following inputs:

   1) SPI received from the Initiator,

   2) requested new value for the managed Anti-Replay Sequence Number received from the Initiator;

c) have the following output: Security Association with new Anti-Replay Sequence Number;

d) execute the following:

   1) verify that the SA exists and that its service type is Authentication or Authenticated Encryption,

   2) replace the current value of the managed Anti-Replay Sequence Number with the requested value.

3.3.3.8 Set Anti-Replay Sequence Number Window

3.3.3.8.1 Overview

The Set Anti-Replay Sequence Number Window directive is used to initialize the managed Anti-Replay Sequence Number window for a Security Association to the value supplied by the service user.

3.3.3.8.2 Preconditions for the Procedure

The Security Association service type must be Authentication or Authenticated Encryption.

3.3.3.8.3 Procedural Steps

3.3.3.8.3.1 General

The Set Anti-Replay Sequence Number Window (ARSNW) procedure shall include the following mandatory execution steps:

a) Execution of Set ARSNW; Role: Initiator;

b) Signaling of Set ARSNW Request; Role: Initiator;

c) Execution of Set ARSNW; Role: Recipient.

3.3.3.8.3.2 Execution of Set ARSNW

The Execution of Set ARSNW step shall

a) be executed by the Initiator;

b) have the following inputs:

   1) SPI of an existing Security Association,

   2) requested new value for the managed Anti-Replay Sequence Number Window;

c) have the following output: none; and

d) execute the following: replace the current value of the managed Anti-Replay Sequence Number window with the requested value.

3.3.3.8.3.3 Signaling of Set ARSNW Request

The Signaling of Set ARSNW Request step shall

a) be executed by the Initiator;

b) have the following inputs:

   1) the SPI of the Security Association,

   2) requested new value for the managed Anti-Replay Sequence Number window;

c) have the following outputs: SPI and new ARSNW value transmitted to the Recipient; and

d) execute the following: a Set ARSNW Command PDU, as defined in 5.5.1.9, shall be created and transmitted to the Recipient using the interface specified in section 4.

3.3.3.8.3.4 Execution of Set ARSNW

The Execution of Set ARSNW step shall

a) be executed by the Recipient;

b) have the following inputs:

   1) SPI received from the Initiator,

2) requested new value for the managed Anti-Replay Sequence Number window received from the Initiator;

c) have the following outputs: none; and

d) execute the following:

1) verify that the SA exists and that its service type is Authentication or Authenticated Encryption,

2) replace the current value of the managed Anti-Replay Sequence Number window with the requested value.

3.3.3.9 SA Status Request

3.3.3.9.1 Overview

The SA Status Request directive is used to request a summary of the current status of a Security Association.

3.3.3.9.2 Preconditions for the Procedure

There are no preconditions for the procedure.

3.3.3.9.3 Procedural Steps

3.3.3.9.3.1 General

The SA Status Request procedure shall include the following mandatory execution steps:

a) Signaling of SA Status Request; Role: Initiator;

b) Execution of SA Status Verification; Role: Recipient;

c) Signaling of SA Status Response; Role: Recipient.

3.3.3.9.3.2 Signaling of SA Status Request

The Signaling of SA Status Request step shall

a) be executed by the Initiator;

b) have the following input: SPI of an existing Security Association;

c) have the following output: SA Status Request transmitted to the Recipient; and

d) execute the following: an SA Status Request Command PDU, as defined in 5.5.1.10, shall be created and transmitted to the Recipient using the interface specified in section 4.

3.3.3.9.3.3 Execution of SA Status Verification

The Execution of SA Status Verification step shall

a) be executed by the Recipient;

b) have the following input: SA Status Request Command PDU received from the Initiator;

c) have the following output: state of an existing Security Association; and

d) execute the following: retrieve the most recent state transition for the SA indicated, or the current state of the SA if no previous state transition is known.

3.3.3.9.3.4 Signaling of SA Status Response

The Signaling of SA Status Response step shall

a) be executed by the Recipient;

b) have the following input: state of an existing Security Association created in Step 3.3.3.9.3.3;

c) have the following output: state of an existing Security Association transmitted to the Initiator;

d) execute the following: an SA Status Request Reply PDU, as defined in 5.5.1.10, shall be created and transmitted to the Initiator using the interface specified in section 4.

3.3.3.10 Read Anti-Replay Sequence Number

3.3.3.10.1 Overview

The Read ARSN directive is used to read the current Anti-Replay Sequence Number value associated with a given SA.

3.3.3.10.2 Preconditions for the Procedure

There are no preconditions for the procedure.

3.3.3.10.3 Procedural Steps

3.3.3.10.3.1 General

The Read ARSN procedure shall include the following mandatory execution steps:

a) Signaling of Read Anti-Replay Sequence Number Request; Role: Initiator;

b) Computation of the Read Anti-Replay Sequence Number Response; Role: Recipient;

c) Signaling of the Read Anti-Replay Sequence Number Response; Role: Recipient;

3.3.3.10.3.2 Signaling of Read Anti-Replay Sequence Number Request

The Signaling of Read Anti-Replay Sequence Number Request step shall

a) be executed by the Initiator;

b) have the following input: SPI of the SA;

c) have the following output: the Read Anti-Replay Sequence Number Request transmitted to the Recipient; and

d) execute the following: a Read Anti-Replay Sequence Number Command PDU, as defined in 5.5.1.11.2, shall be created and transmitted to the Recipient using the interface specified in section 4.

3.3.3.10.3.3 Computation of the Read Anti-Replay Sequence Number Response

The Computation of the Read Anti-Replay Sequence Number Response step shall

a) be executed by the Recipient;

b) have the following input: reception of Read Anti-Replay Sequence Number from the Initiator, including the SPI of the SA;

c) have the following output: Read Anti-Replay Sequence Number Response; and

d) execute the following: the recipient shall read the ARSN Value corresponding to the SA identified by the SPI and create the Read Anti-Replay Sequence Number Response.

3.3.3.10.3.4 Signaling of the Read Anti-Replay Sequence Number Response

The Signaling of the Read Anti-Replay Sequence Number Response step shall

a) be executed by the Recipient;

b) have the following input: the Read Anti-Replay Sequence Number Response created at Step 3.3.3.10.3.3;

c) have the following output: Read Anti-Replay Sequence Number Response transmitted to the Initiator; and

d) execute the following: a Read Anti-Replay Sequence Number Reply PDU, as defined in 5.5.1.11.3, shall be created and transmitted to the Initiator using the interface specified in section 4.

3.4 MONITORING & CONTROL SERVICE

3.4.1 OVERVIEW

The following service procedures are specified:

– Ping;

– Log Status;

– Dump Log;

– Erase Log;

– Self-Test;

– Alarm Flag Reset.

3.4.2 SERVICE PARAMETERS

3.4.2.1 Ping

The Ping procedure shall have no service parameter.

3.4.2.2 Log Status

The Log Status procedure shall have the following service parameters:

a) number of security messages stored in the Security Log (Integer);

b) available space in the Security Log (Integer).

3.4.2.3 Dump Log

The Dump Log procedure shall have the following service parameter:

set of security messages stored in the Security Log.

NOTE – The content of each security message is implementation specific and not specified by this Recommended Standard.

3.4.2.4 Erase Log

The Erase Log shall have the following service parameters:

a) number of messages stored in the Security Log after erasing (Integer);

b) space available in the Security Log (Integer).

3.4.2.5 Self-Test

The Self-Test procedure shall have the following service parameter:

Self-Test Result: OK/NOK (8 bit).

3.4.2.6 Alarm Flag Reset

The Alarm Flag Reset procedure shall have no service parameter.

3.4.3 SERVICE PROCEDURES

3.4.3.1 Ping

3.4.3.1.1 Overview

The ping procedure is used to test the status of a SDLS Recipient Security Function protecting a TC, TM, AOS, or USLP link. This directive generates a report. The intention behind the Ping procedure is to check that the Recipient Security Function is alive.

3.4.3.1.2 Preconditions for the procedure

There are no preconditions for the procedure.

3.4.3.1.3 Procedural steps

3.4.3.1.3.1 General

The Ping procedure shall include the following mandatory execution steps:

a) signaling of the Ping Request; Role: Initiator;

b) signaling of the Ping Response; Role: Recipient.

3.4.3.1.3.2 Signaling of the Ping Request

The Signaling of the Ping Request step shall

a) be executed by the Initiator;

b) have the following input: none;

c) have the following output: the Ping Request transmitted to the Recipient; and

d) execute the following: a Ping Request Command PDU, as defined in 5.6.1.1.2, shall be created and transmitted to the Recipient using the interface specified in section 4.

3.4.3.1.3.3 Signaling of the Ping Response

The Signaling of the Ping Response step shall

a) be executed by the Recipient;

b) have the following input: Ping Request received from the Initiator;

c) have the following output: Ping Response transmitted to the Initiator; and

d) execute the following: a Ping Reply PDU, as defined in 5.6.1.1.3, shall be created and transmitted to the Initiator using the interface specified in section 4.

3.4.3.2 Log Status

3.4.3.2.1 Overview

The Log Status directive is used to read the status of the Security Log, by asking for the number of event messages stored in the Security Log.

3.4.3.2.2 Preconditions for the Procedure

There are no preconditions for the procedure.

3.4.3.2.3 Procedural Steps

3.4.3.2.3.1 General

The Log Status procedure shall include the following mandatory execution steps:

a) Signaling of the Log Status Request; Role: Initiator;

b) Generation of the Log Status Response; Role: Recipient;

c) Signaling of the Log Status Response; Role: Recipient.

3.4.3.2.3.2 Signaling of the Log Status Request

The Signaling of the Log Status Request step shall

a) be executed by the Initiator;

b) have the following input: none;

c) have the following output: the Log Status Request transmitted to the Recipient; and

d) execute the following: a Log Status Request Command PDU, as defined in 5.6.1.2, shall be created and transmitted to the Recipient using the interface specified in section 4.

3.4.3.2.3.3 Generation of the Log Status Response

The Generation of the Log Status Response step shall

a) be executed by the Recipient;

b) have the following input: reception of Log Status Request from the Initiator;

c) have the following output: Log Status Response; and

d) execute the following:

   1) the Recipient shall assess the status of the Security Log and derive

      i) number of entries in the log,

      ii) remaining capacity in the log; and

   2) generate Log Status Response.

3.4.3.2.3.4 Signaling of the Log Status Response

The Signaling of the Log Status Response step shall

a) be executed by the Recipient;

b) have the following input: Log Status Response created at Step 3.4.3.2.3.3;

c) have the following output: the Log status reply transmitted to the Initiator; and

d) execute the following: a Log Status Reply PDU, as defined in 5.6.1.2, shall be created and transmitted to the Initiator using the interface specified in section 4.

3.4.3.3 Dump Log

3.4.3.3.1 Overview

The Dump Log directive is used to send to the ground the content of the Security Log. This directive does not erase the Security Log.

3.4.3.3.2 Preconditions for the Procedure

There are no preconditions for the procedure.

3.4.3.3.3 Procedural Steps

3.4.3.3.3.1 General

The Dump Log procedure shall include the following mandatory execution steps:

a) Signaling of the Dump Log request; Role: Initiator;

b) Computation of the Dump Log Response, comprising the entire set of messages stored in the Security Log; Role: Recipient;

c) Signaling of the Dump Log Response; Role: Recipient.

3.4.3.3.3.2 Signaling of the Dump Log Request

The Signaling of the Dump Log Request step shall

a) be executed by the Initiator;

b) have the following input: none;

c) have the following output: Dump Log Request transmitted to the Recipient;

d) execute the following: a Dump Log Request Command PDU, as defined in 5.6.1.3, shall be created and transmitted to the Recipient using the interface specified in section 4.

3.4.3.3.3.3 Computation of the Dump Log Response

The Computation of the Dump Log Response step shall

a) be executed by the Recipient;

b) have the following input: reception of Dump Log Request from the Initiator;

c) have the following output: Dump Log Response; and

d) execute the following:

   1) the Recipient shall derive the List of Log Security Messages from the Security Log,

   2) generate the Dump Log Response comprising all the Security Messages.

3.4.3.3.3.4 Signaling of the Dump Log Response

The Signaling of the Dump Log Response step shall

a) be executed by the Recipient;

b) have the following input: the Dump Log Response created at Step 3.4.3.3.3.3;

c) have the following output: Dump Log Response transmitted to the Initiator; and

d) execute the following: a Dump Log Reply PDU, as defined in 5.6.1.3, shall be created and transmitted to the Initiator using the interface specified in section 4.

3.4.3.4 Erase Log

3.4.3.4.1 Overview

The Erase Log directive is used to erase the Security Log.

3.4.3.4.2 Preconditions for the Procedure

There are no preconditions for the procedure.

3.4.3.4.3 Procedural Steps

3.4.3.4.3.1 General

The Erase Log procedure shall include the following mandatory execution steps:

a) Signaling of Erase Log Request; Role: Initiator;

b) Erasing of the entire set of messages stored in the Security Log and generation of the Erase Log Response; Role: Recipient;

c) Signaling of the Erase Log Response; Role: Recipient.

3.4.3.4.3.2 Signaling of the Erase Log Request

The Signaling of the Erase Log Request step shall

a) be executed by the Initiator;

b) have the following input: none;

c) have the following output: Erase Log Request transmitted to the Recipient; and

d) execute the following: an Erase Log Command PDU, as defined in 5.6.1.4, shall be created and transmitted to the Recipient using the interface specified in section 4.

3.4.3.4.3.3 Erasing of the Entire Set of Messages Stored in the Security Log and Computation of the Erase Log Response

The Erasing of the Entire Set of Messages Stored in the Security Log and Computation of the Erase Log Response step shall

a) be executed by the Recipient;

b) have the following input: reception of Erase Log Request from the Initiator;

c) have the following outputs:

1) number of entries in the Security Log after Erasure,

2) remaining space in the Security Log after Erasure; and

d) execute the following:

1) erase all Security Messages from the Security Log,

2) assess the status of the Security Log and derive

i) the number of entries in the Security Log after Erasure,

ii) the remaining space in the Security Log after Erasure;

3) generate the Erase Log Response comprising the number of entries and the remaining space in the Security Log.

3.4.3.4.3.4 Signaling of the Erase Log Response

The Signaling of the Erase Log Response step shall

a) be executed by the Recipient;

b) have the following input: Erase Log Response created at Step 3.4.3.4.3.3;

c) have the following output: Erase Log Response transmitted to the Initiator;

d) execute the following: an Erase Log Reply PDU, as defined in 5.6.1.4, shall be created and transmitted to the Initiator using the interface specified in section 4.

3.4.3.5 Self-Test

3.4.3.5.1 Overview

The Self-Test directive is used to trigger a Recipient Security Function self-test.

3.4.3.5.2 Preconditions for the Procedure

There are no preconditions for the procedure.

3.4.3.5.3 Procedurals Steps

3.4.3.5.3.1 General

The Self-Test procedure shall include the following mandatory execution steps:

a) Signaling of Self-Test Request; Role: Initiator;

b) Computation of the Self-Test Response; Role: Recipient;

c) Signaling of Self-Test Response; Role: Recipient.

3.4.3.5.3.2 Signaling of Self-Test Request

The Signaling of Self-Test Request step shall

a) be executed by the Initiator;

b) have the following input: none;

c) have the following output: Self-Test Request transmitted to the Recipient; and

d) execute the following: a Self-Test Command PDU, as defined in 5.6.1.5, shall be created and transmitted to the Recipient using the interface specified in section 4.

3.4.3.5.3.3 Computation of the Self-Test Response

The Computation of the Self-Test Response step shall

a) be executed by the Recipient;

b) have the following input: Reception of Self-Test Request from the Initiator;

c) have the following output: Self-Test Response; and

d) execute the following: upon reception of the Self-Test Request, the Recipient shall run a self-test and create the Self-Test Response.

NOTE – The self-test is implementation specific and not specified by this Recommended Standard.

3.4.3.5.3.4 Signaling of Self-Test Response

The Signaling of Self-Test Response step shall

a) be executed by the Recipient;

b) have the following input: the Self-Test Response created in Step 3.4.3.5.3.3;

c) have the following output: Self-Test Response transmitted to the Initiator; and

d) execute the following: a self-Test Reply PDU, as defined in 5.6.1.5, shall be created and transmitted to the Initiator using the interface specified in section 4.

3.4.3.6 Alarm Flag Reset

3.4.3.6.1 Preconditions for the Procedure

There are no preconditions for the procedure.

3.4.3.6.2 Procedurals Steps

3.4.3.6.2.1 General

The Alarm Flag Reset procedure shall include the following mandatory execution steps:

a) Signaling of Alarm Flag Reset Request; Role: Initiator;

b) Resetting the Alarm Flag of the Frame Security Report; Role: Recipient.

3.4.3.6.2.2 Signaling of Alarm Flag Reset Request

The Signaling of Alarm Flag Reset Request step shall

a) be executed by the Initiator;

b) have the following input: none;

c) have the following output: Alarm Flag Reset Request transmitted to the Recipient; and

d) execute the following: an Alarm Flag Reset Command PDU, as defined in 5.6.1.6, shall be created and transmitted to the Recipient using the interface specified in section 4.

3.4.3.6.2.3 Resetting the Alarm Flag of the Frame Security Report

The Resetting the Alarm Flag of the Frame Security Report step shall

a) be executed by the Recipient;

b) have the following input: reception of Alarm Flag Reset Request from the Initiator;

c) have the following outputs: Alarm Flag of the FSR reset; and

d) execute the following: the recipient shall reset the Alarm Flag of the FSR.

NOTE – The way the Alarm Flag is stored and reset is implementation dependent and not specified in this standard.

4 INTERFACE WITH SLP & SDLS

4.1 OVERVIEW

The SDLS Extended Procedures are interfacing with the Space Link Protocols (SLP) for transport of the procedures Protocol Data Units (PDUs). This Recommended Standard mandates the SLP services be used for transfer of SDLS Extended Procedures PDUs over the space link. A new OCF, the FSR, is also specified to provide real-time reporting of the Recipient security function to the Initiator. The SDLS Extended Procedures Concept of Operations (reference [**C**13]) describes various options to implement the interface.

Since the SDLS Extended Procedures are meant to provide additional capabilities to the core protocol, they do require interfacing with it. The interfaces are, however, generally on the Initiator and Recipient side, and not directly on SDLS protocol level. However, it is recommended that at least one Security Association be allocated for transporting Extended Procedures PDUs over the space link.

4.2 INTERFACE WITH SLP

4.2.1 TRANSFER OF EXTENDED PROCEDURES SERVICE PDU OVER THE SPACE LINK

4.2.1.1 For transport of SDLS Extended Procedures PDUs on the TC link (TC data link protocol), the MAP packet service with a dedicated MAP shall be used (see references [6] and [8]).

4.2.1.2 For transport of SDLS Extended Procedures PDUs on the TM downlink (data link protocols), the VC packet service shall be used (see references [4] and [8]).

4.2.1.3 For transport of SDLS Extended Procedures PDUs using AOS (link or downlink), the VC packet service shall be used (see references [5] and [8]).

4.2.1.4 For transport of SDLS Extended Procedures PDUs using USLP (link or downlink), the MAP packet service shall be used (see references [9] and [8]).

NOTE – Grouping EP PDUs in one single packet is a way of ensuring that the related PDUs are transferred together.

4.2.2 FRAME SECURITY REPORT

4.2.2.1 General

4.2.2.1.1 The FSR, which is the protocol data unit transmitted from the Recipient to the Initiator of an SDLS secured TC, AOS, or USLP link, shall provide the systematic, real-time mechanism by which the SDLS function at the receiving end reports the status of TC, AOS, or USLP frame acceptance to the sending end.

NOTE – The FSR is not the only reporting mechanism for this SDLS protocol. Several on-demand or on-event reporting mechanisms and corresponding messages are specified in this Recommended Standard. They provide non-real-time or non-systematic reporting of the frame acceptance status at the receiving end of the SDLS secured TC, AOS, or USLP link.

4.2.2.1.2 The FSR shall be carried in the Operational Control Field of TM, AOS, or USLP Transfer Frames (references [4], [5], [9]) using the Master Channel Operational Control Field (MC_OCF) Service or the Virtual Channel Operational Control Field (VC_OCF) Service.

4.2.2.1.3 The FSR shall be sampled by the Recipient for each received SDLS protected frame.

4.2.2.1.4 The 32-bit FSR shall consist of 6 fields, positioned contiguously, in the following sequence:

a) Control Word Type (1 bit, mandatory);

b) FSR Version Number (3 bits, mandatory);

c) Alarm field (1 bit, mandatory);

d) Security event flags (3 bits, mandatory);

e) Last SPI field (16 bits, mandatory);

f) ARSN Value field (8 bits, mandatory).

NOTE – The structural components of the FSR are shown in figure 4-1.

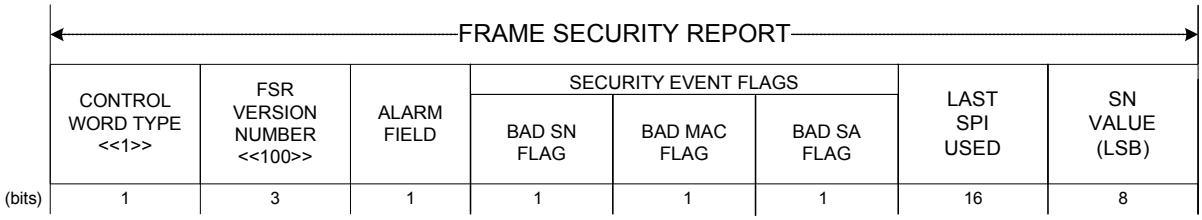

Figure 4-1: Frame Security Report

4.2.2.2 Control Word Type

4.2.2.2.1 Bit 0 of the FSR shall contain the Control Word Type.

NOTE – This field is used to distinguish Control Word Type ‘0’ (CLCW, specified in reference [6]) from other types of control words (Control Word Type ‘1’), such as the FSR, that may be alternatively carried in the OCF of TM, AOS, or USLP transfer frames.

4.2.2.2.2 This one-bit field shall be set to ‘1’.

4.2.2.3 FSR Version Number

4.2.2.3.1 Bits 1-3 of the FSR shall contain the FSR Version Number.

4.2.2.3.2 This 3-bit field shall be set to '100'.

NOTE – The FSR Version Number first bit ('1') identifies a CCSDS defined Type-2 OCF. The last 2 bits ('00') identify a 'Version-1' FSR, whose binary encoded Version Number is '00'. At present, a single version is defined in this Recommended Standard. The FSR Version Number is included to provide future growth flexibility.

4.2.2.4 Alarm Field

4.2.2.4.1 Bit 4 of the FSR shall contain the Alarm Flag.

4.2.2.4.2 The Alarm Flag shall indicate whether a TC, AOS, or USLP link Transfer Frame has been rejected by the Recipient Security Function.

4.2.2.4.3 A setting of '0' in the Alarm Flag shall indicate that all TC, AOS, or USLP link Transfer Frames have been accepted by Recipient Security Function since the last reset of the Alarm Flag.

4.2.2.4.4 A setting of '1' in the Alarm Flag shall indicate that at least one TC, AOS, or USLP link Transfer Frame has been rejected by the Recipient Security Function since the last reset of the Alarm Flag.

4.2.2.4.5 The Alarm Flag shall apply to all Virtual Channels and Security Associations of the TC, AOS, or USLP link.

4.2.2.4.6 The Alarm Flag shall be updated at each TC, AOS, or USLP link Transfer Frame processed by the Recipient Security Function.

4.2.2.4.7 Once the Alarm Flag is set to '1', it shall remain persistent until reset to '0' (NoAlarm state) by a dedicated command (see 5.6.1.6).

4.2.2.5 Security Event Flags

4.2.2.5.1 General

Bits 5-7 of the FSR shall contain the Flags specified in the following subsections.

4.2.2.5.2 Bad Sequence Number Flag

4.2.2.5.2.1 Bit 5 of the FSR shall contain the Bad Sequence Number Flag.

4.2.2.5.2.2 The Bad Sequence Number Flag shall indicate whether the ARSN of the last received TC, AOS, or USLP link Transfer Frame by the Recipient Security Function is valid.

4.2.2.5.2.3 A setting of '0' in the Bad Sequence Number Flag shall indicate that the ARSN carried by the last received TC, AOS, or USLP link Transfer Frame by the Recipient Security Function is valid (i.e., within the ARSN window).

4.2.2.5.2.4 A setting of '1' in the Bad Sequence Number Flag shall indicate that the ARSN carried by the last received TC, AOS, or USLP link Transfer Frame by the Recipient Security Function is invalid (i.e., outside the ARSN window).

4.2.2.5.2.5 The Bad Sequence Number Flag shall be updated at each TC, AOS, or USLP link Transfer Frame processed by the Recipient Security Function, its states not being persistent.

4.2.2.5.3 Bad MAC Flag

4.2.2.5.3.1 Bit 6 of the FSR shall contain the Bad MAC Flag.

4.2.2.5.3.2 The Bad MAC Flag shall indicate whether the last received TC, AOS, or USLP link Transfer Frame by the Recipient Security Function failed MAC verification.

4.2.2.5.3.3 A setting of '0' in the Bad MAC Flag shall indicate that the MAC carried by the last received TC, AOS, or USLP link Transfer Frame by the Recipient Security Function is valid (i.e., matches the MAC computed over the received Transfer Frame).

4.2.2.5.3.4 A setting of '1' in the Bad MAC Flag shall indicate that the MAC carried by the last received TC, AOS, or USLP link Transfer Frame by the Recipient Security Function is invalid (i.e., does not match the MAC computed over the received Transfer Frame).

4.2.2.5.3.5 The Bad MAC Flag shall be updated at each TC, AOS, or USLP link Transfer Frame processed by the Recipient Security Function, its states not being persistent.

4.2.2.5.4 Bad SA Flag

4.2.2.5.4.1 Bit 7 of the FSR shall contain the Bad SA Flag.

4.2.2.5.4.2 The Bad SA Flag shall indicate whether the last TC, AOS, or USLP link Transfer Frame received by the Recipient Security Function

- failed SA verification;
- carried an SPI pointing to an SA that is not in Operational State; or
- carried an SPI pointing to an Operational SA associated with a key that is not in Active State.

4.2.2.5.4.3 A setting of '0' in the Bad SA Flag shall indicate that the SPI carried by the last TC, AOS, or USLP link Transfer Frame received by the Recipient Security Function is valid.

4.2.2.5.4.4 A setting of '1' in the Bad SA Flag shall indicate that the SPI carried by the last TC, AOS, or USLP link Transfer Frame received by the Recipient Security Function is invalid.

4.2.2.5.4.5 The Bad SA Flag shall be updated at each TC, AOS, or USLP link Transfer Frame processed by the Recipient Security Function, its states not being persistent.

4.2.2.6 Last SPI used

4.2.2.6.1 Bits 8-23 of the FSR shall contain the SPI carried in the last received TC, AOS, or USLP link Transfer Frame by the Recipient Security Function.

NOTE – Bad MAC, Bad Sequence Number, and Bad SA flags are always associated with this SPI.

4.2.2.6.2 Sequence Number Value (LSB)

Bits 24-31 of the FSR shall contain the 8 Least Significant Bits (LSBs) of the ARSN carried in the last received TC, AOS, or USLP link Transfer Frame by the Recipient Security Function.

4.3 INTERFACE WITH SDLS

4.3.1 TRANSFER OF EP SERVICE PDUS

4.3.1.1 All EP Service Command PDUs shall only be transmitted over a SDLS channel protected by authentication or authenticated encryption.

NOTE – The choice between authentication-only and authenticated-encryption for the transmission of EP Service PDUs is driven by the mission risks analysis.

4.3.1.2 The two SDLS reserved SPIs (values of 'all zeros' [0] and 'all ones' [65535]) defined in reference [1] may be used for exchanging EP Service PDUs.

NOTE – To avoid unintentional loss of control of an SA, it is generally good practice to use a different SA from the one being affected by the EP Service Command PDU.

5 PROCEDURES SPECIFICATION

5.1 OVERVIEW

This section describes the procedures that are used to provide the Key Management, Security Association Management, and SDLS Monitoring & Control Services.

5.2 PROCEDURE IDENTIFICATION

The Extended Procedures PDU header shall identity the type of procedure with which the contents of the Extended Procedures PDU data field are associated.

NOTE – A mapping of field values to procedures is contained in table 5-1.

5.3 PROTOCOL DATA UNITS

5.3.1 OVERVIEW

5.3.1.1 General

Extended Procedures PDUs are the data structures that carry the information related to Extended Procedures commands and reports.

5.3.1.2 Tag, Length, Value Notation

SDLS Extended Procedures commands and reports share a common message format based on the ‘TLV’ concept. The Tag field uniquely identifies the command or the report. The Length field indicates the length of the Value field (may be zero). The (optional) Value field contains additional data pertaining to the message. As long as the Tag and Length fields are of fixed length, the TLV concept is very flexible, allowing for the defining of new commands and reports while maintaining full compatibility with previously defined messages. Figure 5-1 shows the TLV format.

For example, this flexibility may be used by an implementer who needs to define some proprietary messages while still retaining full CCSDS compatibility. (As long as CCSDS defined messages are correctly implemented, proprietary messages will simply be skipped if not recognized, thanks to the TLV format.) It should be noted that the TLV concept allows nesting: the Value field can itself be composed of one or more TLV messages.

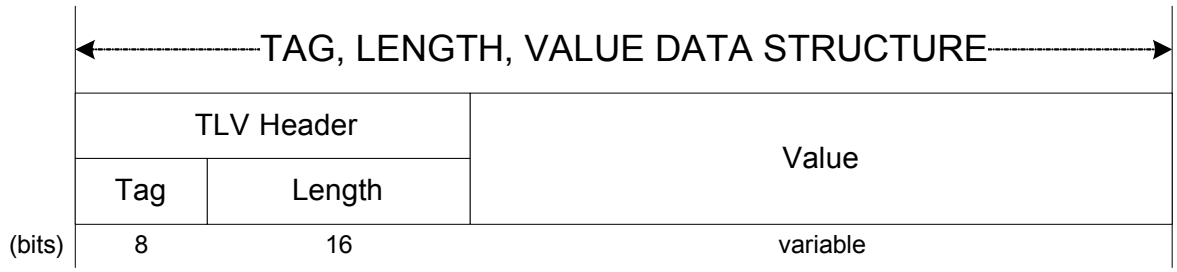

Figure 5-1: TLV Format Specification

5.3.2 EXTENDED PROCEDURES PDU

5.3.2.1 General

5.3.2.1.1 The Extended Procedures PDU shall be used for transport of SDLS Extended Procedures Commands and Replies.

5.3.2.1.2 The Extended Procedures PDU shall consist of two mandatory fields, positioned contiguously, in the following sequence:

a) Extended Procedures PDU Header (24 bits, mandatory);

b) Extended Procedures PDU Data Field (variable but octet-aligned, mandatory).

NOTES

1 Wherever an Extended Procedures PDU carries a field whose length in bits is greater than the length of the value occupying that field, it is assumed that the value is stored right-justified.

2 The format of the Extended Procedures PDU is shown in figure 5-2.

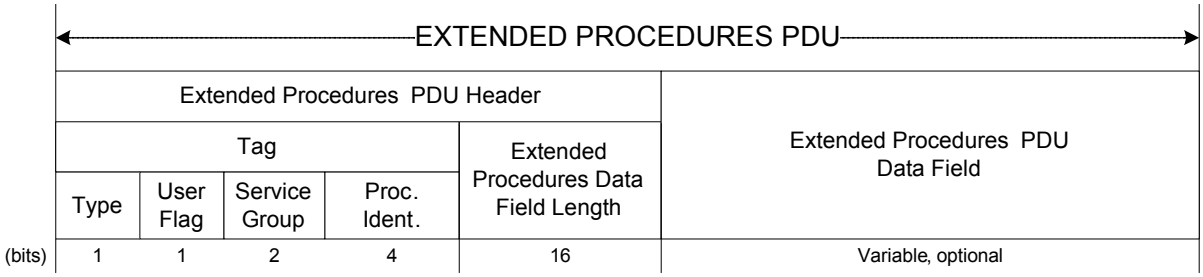

Figure 5-2: Extended Procedures PDU

5.3.2.2 Extended Procedures PDU Header

5.3.2.2.1 General

5.3.2.2.1.1 The Extended Procedures PDU Header shall consist of two mandatory fields, positioned contiguously, in the following sequence:

a) Extended Procedures Tag (8 bits, mandatory);

b) Extended Procedures Data Field Length (16 bits, mandatory).

5.3.2.2.1.2 The Extended Procedures Tag shall consist of four mandatory fields, positioned contiguously, in the following sequence:

a) Procedure Type (1 bit, mandatory);

b) User Flag (1 bit, mandatory);

c) Service Group Field (2 bits, mandatory);

d) Procedure Identification Field (4 bits, mandatory).

5.3.2.2.2 The Procedure Type Flag

5.3.2.2.2.1.1 The Procedure Type Flag shall identify if the Extended Procedures PDU is associated with a command from the Initiator to the Recipient, or a reply from the Recipient.

5.3.2.2.2.1.2 A setting of '0' shall identify a command.

5.3.2.2.2.1.3 A setting of '1' shall identify a reply.

5.3.2.2.2.2 User Flag

5.3.2.2.2.2.1 The User Flag shall identify if the Extended Procedures PDU is carrying a CCSDS defined procedure or a user defined procedure.

5.3.2.2.2.2.2 A setting of '0' shall identify a CCSDS defined procedure.

5.3.2.2.2.2.3 A setting of '1' shall identify a user defined procedure.

5.3.2.2.2.3 Service Group Field

5.3.2.2.2.3.1 The Service Group shall identify the Extended Procedures Service that the Extended Procedures PDU is associated with.

5.3.2.2.2.3.2 A setting of '00' shall identify a Key Management procedure.

5.3.2.2.2.3.3 A setting of '01' shall identify a Security Association Management procedure targeting SAs that handle communication from the Initiator to the Recipient.

5.3.2.2.2.3.4 A setting of '10' shall identify a Security Association Management procedure targeting SAs that handle communication from the Recipient to the Initiator.

5.3.2.2.2.3.5 A setting of '11' shall identify a Security Monitoring & Control procedure.

5.3.2.2.2.4 Procedure Identification Field

5.3.2.2.2.4.1 The Procedure Identification Field shall identify the procedure that is being communicated through the Extended Procedures PDU.

5.3.2.2.2.4.2 For CCSDS-defined procedures (see 5.3.2.2.2.2), the field shall have the settings as identified in table 5-1.

Table 5-1: Extended Procedures PDU Header Values

Procedure Identification	Assignment	Service Group
0001	OTAR	00 (Key Management)
0010	Key Activation	00 (Key Management)
0011	Key Deactivation	00 (Key Management)
0100	Key Verification	00 (Key Management)
0110	Key Destruction	00 (Key Management)
0111	Key Inventory	00 (Key Management)
0001	Create SA	01 or 10 (SA Management)
0110	Rekey SA	01 or 10 (SA Management)
1011	Start SA	01 or 10 (SA Management)
1110	Stop SA	01 or 10 (SA Management)
1001	Expire SA	01 or 10 (SA Management)
0100	Delete SA	01 or 10 (SA Management)
1010	Set Anti-Replay Sequence Number	01 or 10 (SA Management)
0101	Set Anti-Replay Sequence Number Window	01 or 10 (SA Management)
0000	Read Anti-Replay Sequence Number	01 or 10 (SA Management)
1111	SA Status Request	01 or 10 (SA Management)
0001	Ping	11 (Security Monitoring & Control)
0010	Log Status Request	11 (Security Monitoring & Control)
0011	Dump Log	11 (Security Monitoring & Control)
0100	Erase Log	11 (Security Monitoring & Control)
0101	Self-Test	11 (Security Monitoring & Control)
0111	Reset Alarm Flag	11 (Security Monitoring & Control)

5.3.2.3 Extended Procedures PDU Data Field Length

5.3.2.3.1 The Extended Procedures Data Field Length shall signal the length of the Extended Procedures PDU Data Field in bits.

5.3.2.3.2 The Extended Procedures Data Field Length value shall be octet-aligned.

5.3.2.4 Extended Procedures PDU Data Field

5.3.2.4.1 The presence of the Extended Procedures PDU Data Field is optional.

5.3.2.4.2 The size of the Extended Procedures PDU Data Field shall be as specified by the Extended Procedures Data Field Length.

5.3.2.4.3 If the Extended Procedures PDU Data Field Length is zero, the Extended Procedures PDU Data Field shall not be present.

5.4 KEY MANAGEMENT

5.4.1 KEY TYPES AND KEY LIFECYCLE

5.4.1.1 Key Types

Key Types shall be specified and used according to the recommendations provided in reference [2], subsection 3.1.

5.4.1.2 Key Lifecycle

Key Lifecycle shall be specified and used according to the recommendations provided in reference [2], subsection 3.2.

NOTE – The Suspended and Compromised States as specified in reference [2] are not used in the SDLS Extended Procedures Recommended Standard. The Compromised State as specified in reference [2] is applicable only to the Initiator and no procedure is associated with that State.

5.4.2 KEY MANAGEMENT PROCEDURES

5.4.2.1 Over-the-Air-Rekeying

5.4.2.1.1 General

The OTAR Rekeying Procedure shall support one Extended Procedures PDU data field structure:

OTAR command PDU.

5.4.2.1.2 OTAR command PDU

5.4.2.1.2.1 The OTAR command PDU shall be associated with Step 3.2.3.1.3.3 of the OTAR Procedure, as defined in 3.2.3.1.

5.4.2.1.2.2 The OTAR command PDU shall consist of a managed number of contiguously positioned mandatory fields:

a) Key ID of the master key used for encryption of session keys;

b) Initialization Vector for the authenticated encryption of the Upload Key Block (optional);

c) Upload Key Block consisting of *N* (Encrypted Key ID Field, Encrypted Upload Key Field) field pairs (managed length, mandatory);

d) MAC field for the authenticated encryption of the Upload Key Block.

NOTES

1 The PDU data field of the OTAR Command PDU may be repeated one to *N* times, *N* being a managed parameter.

2 The sizes of the fields of the OTAR Command PDU data field are managed parameters.

3 The format of the OTAR command PDU is shown in figure 5-3.

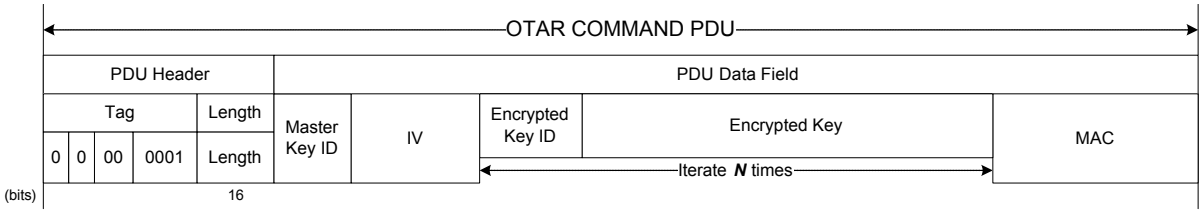

Figure 5-3: OTAR Command PDU

5.4.2.1.2.3 The Master Key ID field shall signal the Key ID of the master key used for encrypted authentication of the Upload Key Block.

5.4.2.1.2.4 The Initialization Vector fields shall signal, if applicable, the Initialization Vector required by the cryptographic algorithm used for authenticated encryption of the Upload Key Block.

5.4.2.1.2.5 The Encrypted Key ID field shall signal, in encrypted format, the identifiers of the session keys to be uploaded.

5.4.2.1.2.6 The Encrypted Upload Key field shall signal, in encrypted format, the cryptographic keys to be uploaded to the Recipient.

5.4.2.1.2.7 The MAC field shall signal the Message Authentication Code computed by the cryptographic algorithm used for authenticated encryption of the Upload Key Block.

5.4.2.2 Key Activation

5.4.2.2.1 General

The Key Activation Procedure shall support one Extended Procedures PDU data field structure:

Key Activation Command PDU.

5.4.2.2.2 Key Activation Command PDU

5.4.2.2.2.1 The Key Activation Command PDU shall be associated with Step 3.2.3.2.3.3 of the Key Activation Procedure, as defined in 3.2.3.2.

5.4.2.2.2.2 The Key Activation Command PDU shall consist of a managed number of contiguously positioned mandatory fields:

Key ID field (managed length, mandatory).

NOTES

1 The PDU data field of the Key Activation Command PDU may be repeated one to *N* times, *N* being a managed parameter.

2 The sizes of the fields of the Key Activation Command PDU data field are managed parameters.

3 The format of the Key Activation Command PDU is shown in figure 5-4.

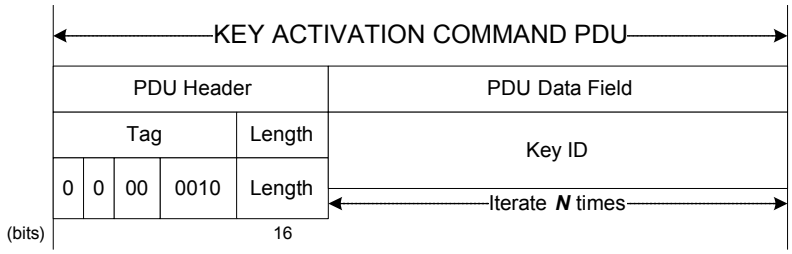

Figure 5-4: Key Activation Command PDU

5.4.2.2.2.3 The Key ID fields shall signal the identifiers of the cryptographic keys to be activated on the recipient.

5.4.2.3 Key Deactivation

5.4.2.3.1 General

The Key Deactivation Procedure shall support one Extended Procedures PDU data field structure:

Key Deactivation Command PDU.

5.4.2.3.2 Key Deactivation Command PDU

5.4.2.3.2.1 The Key Deactivation Command PDU shall be associated with Step 3.2.3.3.3.3 of the Key Deactivation Procedure, as defined in 3.2.3.3.

5.4.2.3.2.2 The Key Deactivation Command PDU shall consist of a managed number of contiguously positioned mandatory fields:

Key ID field (managed length, mandatory).

NOTES

1 The PDU data field of the Key Deactivation Command PDU may be repeated one to *N* times, *N* being a managed parameter.

2 The sizes of the fields of the Key Deactivation Command PDU data field are managed parameters.

3 The format of the Key Deactivation Command PDU is shown in figure 5-5.

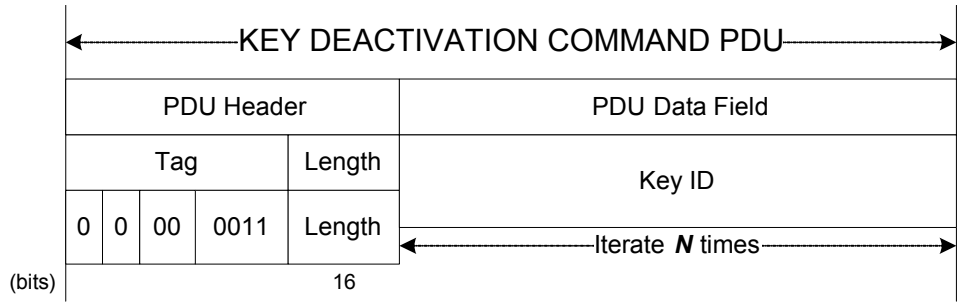

Figure 5-5: Key Deactivation Command PDU

5.4.2.3.2.3 The Key ID fields shall signal the identifiers of the cryptographic keys to be deactivated on the recipient.

5.4.2.4 Key Destruction

5.4.2.4.1 General

The Key Destruction Procedure shall support one Extended Procedures PDU data field structure:

Key Destruction Command PDU.

5.4.2.4.2 Key Destruction Command PDU

5.4.2.4.2.1 The Key Destruction Command PDU shall be associated with Step 3.2.3.4.3.3 of the Key Destruction Procedure, as defined in 3.2.3.4.

5.4.2.4.2.2 The Key Destruction Command PDU shall consist of a managed number of contiguously positioned mandatory fields:

Key ID field (managed length, mandatory).

NOTES

1 The PDU data field of the Key Destruction Command PDU may be repeated one to *N* times, *N* being a managed parameter.

2 The sizes of the fields of the Key Destruction Command PDU data field are managed parameters.

3 The format of the Key Destruction Command PDU is shown in figure 5-6.

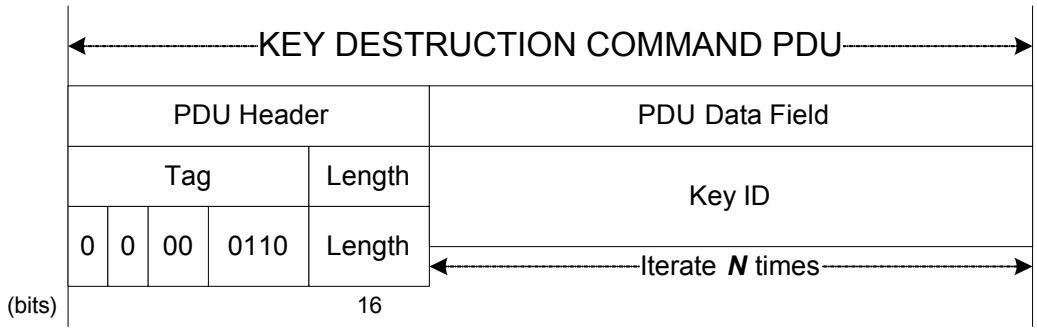

Figure 5-6: Key Destruction Command PDU

5.4.2.4.2.3 The Key ID fields shall signal the identifiers of the cryptographic keys to be destroyed on the recipient.

5.4.2.5 Key Verification

5.4.2.5.1 General

The Key Verification Procedure shall support two Extended Procedures PDU data field structures:

a) Key Verification Command PDU;

b) Key Verification Reply PDU.

5.4.2.5.2 Key Verification Command PDU

5.4.2.5.2.1 The Key Verification Command PDU shall be associated with Step 3.2.3.5.3.3 of the Key Verification Procedure, as defined in 3.2.3.5.

5.4.2.5.2.2 The Key Verification Command PDU shall consist of a managed number of contiguously positioned mandatory fields:

(Key ID Field, Challenge Field) pairs fields (managed length, mandatory).

NOTES

1 The PDU data field of the Key Verification Command PDU may be repeated one to *N* times, *N* being a managed parameter.

2 The sizes of the fields of the Key Verification Command PDU data field are managed parameters.

3 The format of the Key Verification Command PDU is shown in figure 5-7.

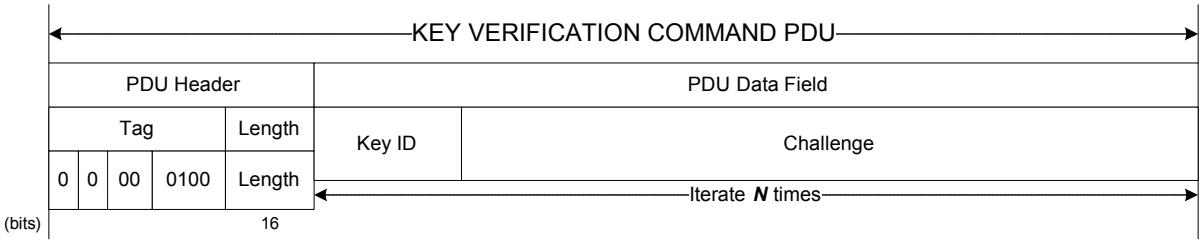

Figure 5-7: Key Verification Command PDU

5.4.2.5.2.3 The Key ID fields shall signal the identifiers of the keys to be verified.

5.4.2.5.2.4 The Challenge fields shall signal the challenges to be used for verification of keys at the Recipient.

5.4.2.5.3 Key Verification Reply PDU

5.4.2.5.3.1 The Key Verification Reply PDU shall be associated with Step 3.2.3.5.3.5 of the Key Verification Procedure, as defined in 3.2.3.5.

5.4.2.5.3.2 The Key Verification Reply PDU shall consist of a managed number of contiguously positioned mandatory fields:

> (Key ID Field, IV Field, Encrypted Challenge Field, Challenge MAC Field) quadruple fields (managed length, mandatory).

NOTES

1 The PDU data field of the Key Verification Reply PDU may be repeated one to *N* times, *N* being a managed parameter.

2 The sizes of the fields of the Key Verification Reply PDU data field are managed parameters.

3 The format of the Key Verification Reply PDU is shown in figure 5-8.

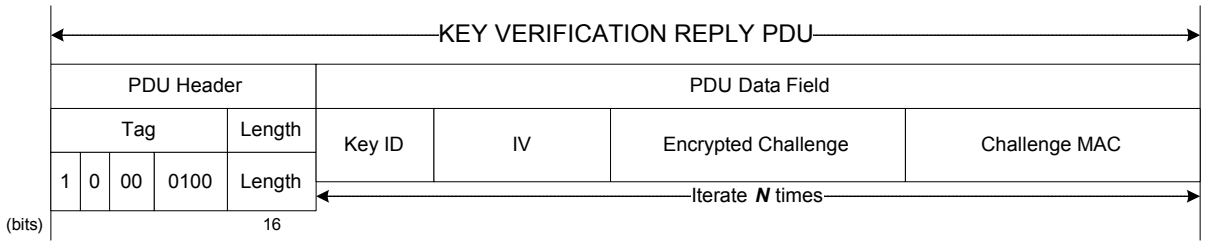

Figure 5-8: Key Verification Reply PDU

5.4.2.5.3.3 The Key ID fields shall signal the identifiers of the keys to be verified.

5.4.2.5.3.4 The Response fields shall signal, in encrypted format, the responses to the challenges.

5.4.2.6 Key Inventory

5.4.2.6.1 General

The Key Inventory Procedure shall support two Extended Procedures PDU data field structures:

a) Key Inventory Command PDU;

b) Key Inventory Reply PDU.

5.4.2.6.2 Key Inventory Command PDU

5.4.2.6.2.1 The Key Inventory Command PDU shall be associated with Step 3.2.3.6.3.2 of the Key Inventory Procedure, as defined in 3.2.3.6.

5.4.2.6.2.2 The Key Inventory Command PDU shall consist of the contiguously positioned mandatory fields:

a) First Key ID in range;

b) Last Key ID in range.

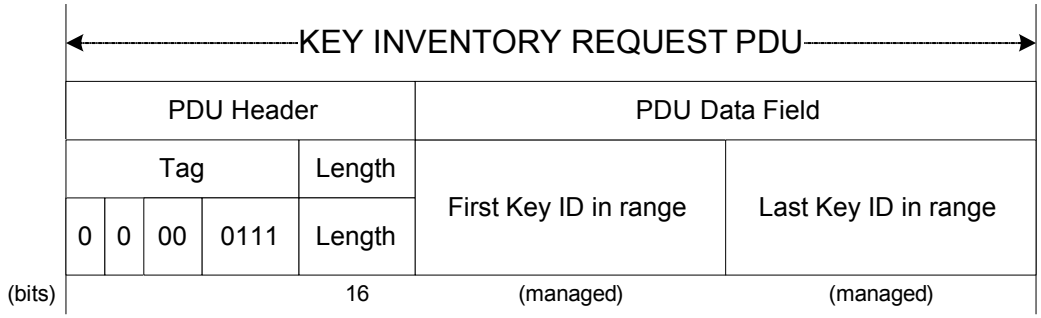

Figure 5-9: Key Inventory Command PDU

5.4.2.6.2.3 The First Key ID and Last Key ID fields shall signal the range of Key IDs to be scanned in the Recipient Key Data Base to extract available keys and their status.

5.4.2.6.3 Key Inventory Reply PDU

5.4.2.6.3.1 The Key Inventory Reply PDU shall be associated with Step 3.2.3.6.3.4 of the Key Inventory Procedure, as defined in 3.2.3.6.

5.4.2.6.3.2 The Key Inventory Reply PDU shall consist of:

a) Total number of Key IDs returned field (16 bits, mandatory);

b) (Key ID, Key state) field pairs (managed length, mandatory).

NOTES

1 (Key ID, Key state) is repeated *N* times, *N* being the 'Total number of Key IDs returned'.

2 The sizes of the Key ID and Key state fields of the Key Inventory Reply PDU data field are managed parameters.

3 The format of the Key Inventory Reply PDU is shown in figure 5-10.

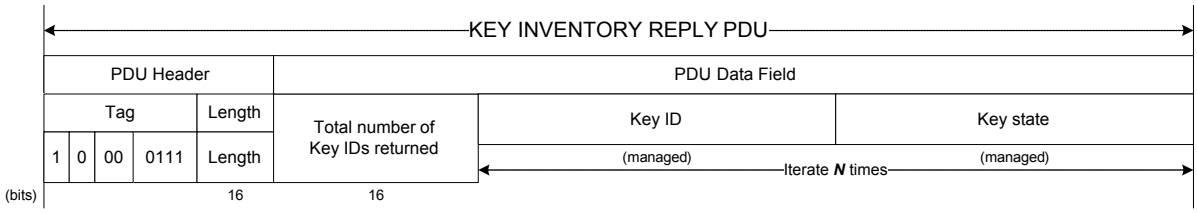

Figure 5-10: Key Inventory Reply PDU

5.4.2.6.3.3 The 'Total number of Key IDs returned' field shall signal the number of keys in the Recipient key data base for the range of Key IDs specified in the Key Inventory Command PDU.

5.4.2.6.3.4 The (Key ID, Key state) fields shall signal, for each key in the range, the Key ID and its state (pre-activation, active, deactivated, destroyed).

5.5 SECURITY ASSOCIATIONS MANAGEMENT

5.5.1 SA MANAGEMENT PROCEDURES

5.5.1.1 Overview

Security Association Management directives and state transitions are shown in figure 5-11.

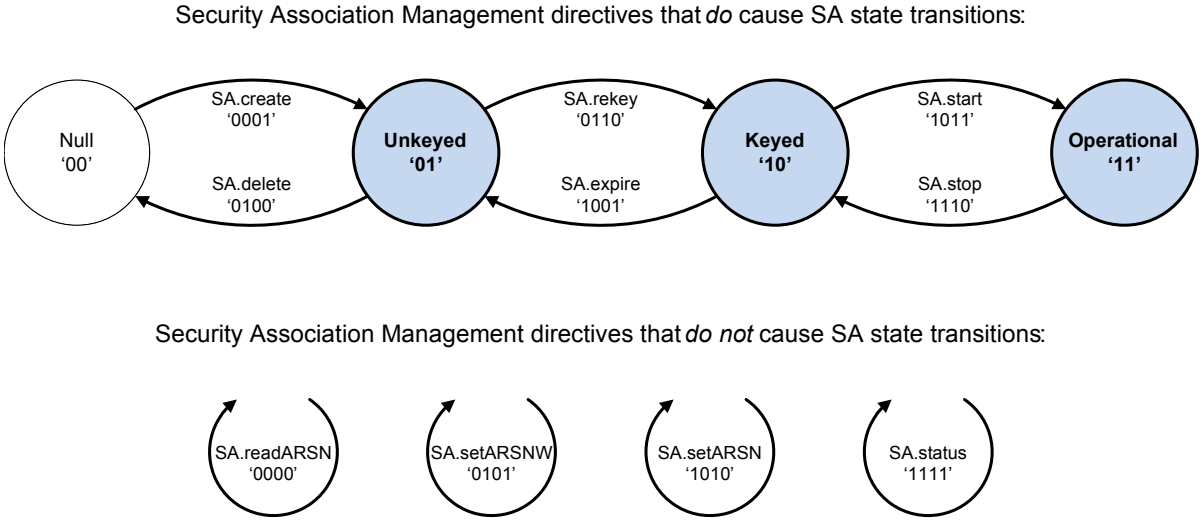

Figure 5-11: SA Management Procedures Overview

5.5.1.2 Start SA

5.5.1.2.1 General

The Start SA Procedure shall support one Extended Procedures PDU data field structure:

Start SA PDU.

5.5.1.2.2 Start SA PDU

5.5.1.2.2.1 The Start SA PDU shall be associated with the Start SA Procedure, as defined in 3.3.3.1.

5.5.1.2.2.2 The Start SA PDU shall consist of a managed number of contiguously positioned mandatory fields:

a) SPI of the applicable Security Association (16 bits, mandatory);

b) one or more Global Virtual Channels (GVC)/Global Multiplexer Access Points (GMAP) with which to use the SA (32 bits each, mandatory).

NOTES

1 The GVC/GMAP ID is a unique identifier comprising the values of the Master Channel ID, Virtual Channel ID, and (if applicable) Multiplexer Access Point ID fields. The precise method of encoding these into a unique 32-bit field is mission-specific.

2 The format of the Start SA PDU is shown in figure 5-12.

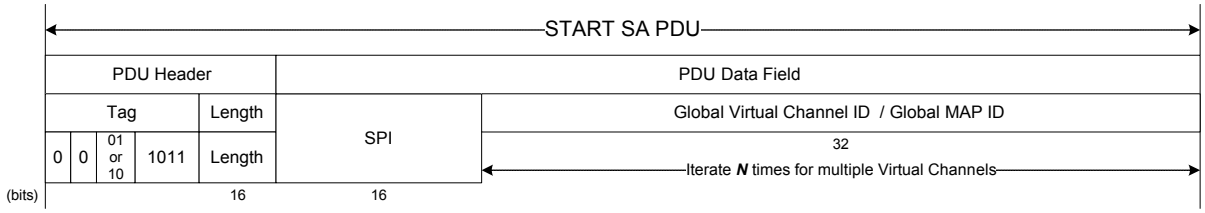

Figure 5-12: Start SA PDU

5.5.1.2.2.3 The SPI field shall signal the applicable Security Association.

5.5.1.2.2.4 The GVC ID/GMAP ID field shall signal the Global Virtual Channel(s)/Global Multiplexer Access Point(s) with which the SA is to be activated.

5.5.1.3 Stop SA

5.5.1.3.1 General

The Stop SA Procedure shall support one Extended Procedures PDU data field structure:

Stop SA PDU.

5.5.1.3.2 Stop SA PDU

5.5.1.3.2.1 The Stop SA PDU shall be associated with the Stop SA Procedure, as defined in 3.3.3.2.

5.5.1.3.2.2 The Stop SA PDU shall consist of a single field:

SPI of the applicable Security Association (16 bits, mandatory).

NOTE – The format of the Stop SA PDU is shown in figure 5-13.

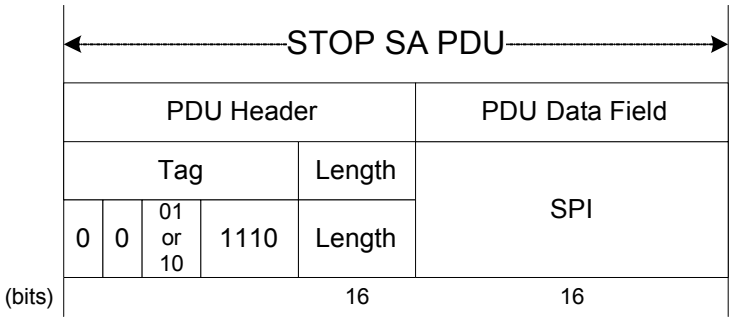

Figure 5-13: Stop SA PDU

5.5.1.3.2.3 The SPI field shall signal the applicable Security Association.

5.5.1.4 Rekey SA

5.5.1.4.1 General

The Rekey SA Procedure shall support one Extended Procedures PDU data field structure:

Rekey SA PDU.

5.5.1.4.2 Rekey SA PDU

5.5.1.4.2.1 The Rekey SA PDU shall be associated with the Rekey SA Procedure, as defined in 3.3.3.3.

5.5.1.4.2.2 The Rekey SA PDU shall consist of a managed number of contiguously positioned fields:

a) SPI of the applicable Security Association (16 bits, mandatory);

b) New encryption key ID for the SA (managed length, optional);

c) New authentication key ID for the SA (managed length, optional);

d) New ARSN of the applicable Security Association (managed length, mandatory);

e) New IV associated with the encryption key if needed (managed length, optional).

NOTE – The format of the Rekey SA PDU is shown in figure 5-14.

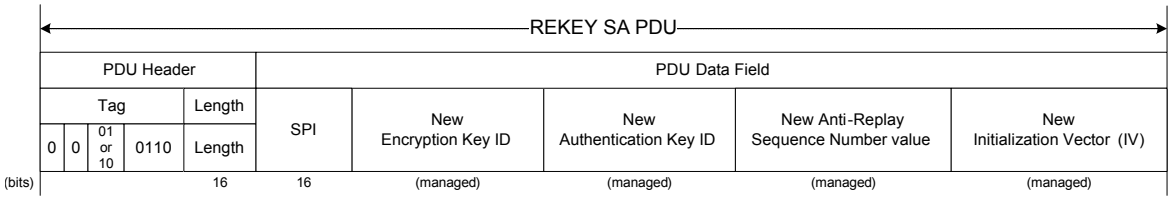

Figure 5-14: Rekey SA PDU

5.5.1.4.2.3 The SPI field shall signal the SPI of the Security Association to be rekeyed.

5.5.1.4.2.4 The New Encryption Key ID field shall signal the new encryption key.

5.5.1.4.2.5 The New Authentication Key ID field shall signal the new authentication key.

5.5.1.4.2.6 If the SA service type is Authenticated Encryption using a combined single-key algorithm, the New Authentication Key ID field shall signal the new key.

5.5.1.4.2.7 The Anti-Replay Sequence Number value field shall signal the new value of the ARSN to be used with the new key.

5.5.1.4.2.8 IV field shall signal the new value of the IV to be used with the encryption key.

5.5.1.5 Expire SA

5.5.1.5.1 General

The Expire SA Procedure shall support one Extended Procedures PDU data field structure:

Expire SA PDU.

5.5.1.5.2 Expire SA PDU

5.5.1.5.2.1 The Expire SA PDU shall be associated with the Expire SA Procedure, as defined in 3.3.3.4.

5.5.1.5.2.2 The Expire SA PDU shall consist of a single mandatory field:

SPI of the applicable Security Association (16 bits, mandatory).

NOTE – The format of the Expire SA PDU is shown in figure 5-15.

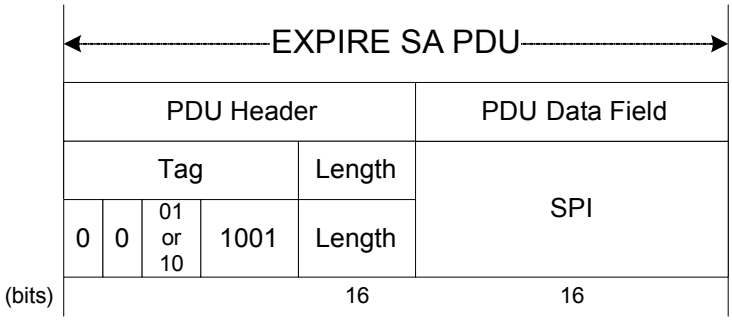

Figure 5-15: Expire SA PDU

5.5.1.5.2.3 The SPI field shall signal the SPI of the Security Association whose key is to be expired.

5.5.1.6 Create SA

5.5.1.6.1 General

The Create SA Procedure shall support one Extended Procedures PDU data field structure:

Create SA PDU.

5.5.1.6.2 Create SA PDU

5.5.1.6.2.1 The Create SA PDU shall be associated with the Create SA Procedure, as defined in 3.3.3.5.

5.5.1.6.2.2 The Create SA PDU shall consist of a managed number of contiguously positioned fields:

a) SPI of the applicable Security Association (16 bits, mandatory);

b) SA Service Type flag for Encryption (1 bit, mandatory);

c) SA Service Type flag for Authentication (1 bit, mandatory);

d) Security Header IV Field Length (6 bits, mandatory);

e) Security Header SN Field Length (6 bits, mandatory);

f) Security Header PL Field Length (2 bits, mandatory);

g) Security Trailer MAC Field Length (8 bits, mandatory);

h) Encryption cipher suite length (8 bits, mandatory);

i) Encryption cipher suite identifier (managed length, optional);

j) Initialization vector length (8 bits, mandatory);

k) IV value (managed length, optional);

l) Authentication cipher suite length (8 bits, mandatory);

m) Authentication cipher suite identifier (managed length, optional);

n) Authentication bit mask length (8 bits, mandatory);

o) Authentication bit mask value (managed length, optional);

p) ARSN length (8 bits, mandatory);

q) ARSN value (managed length, optional);

r) Anti-Replay Sequence Number Window length (8 bits, mandatory);

s) Anti-Replay Sequence Number Window value (managed length, optional).

NOTE – The format of the Create SA PDU is shown in figure 5-16.

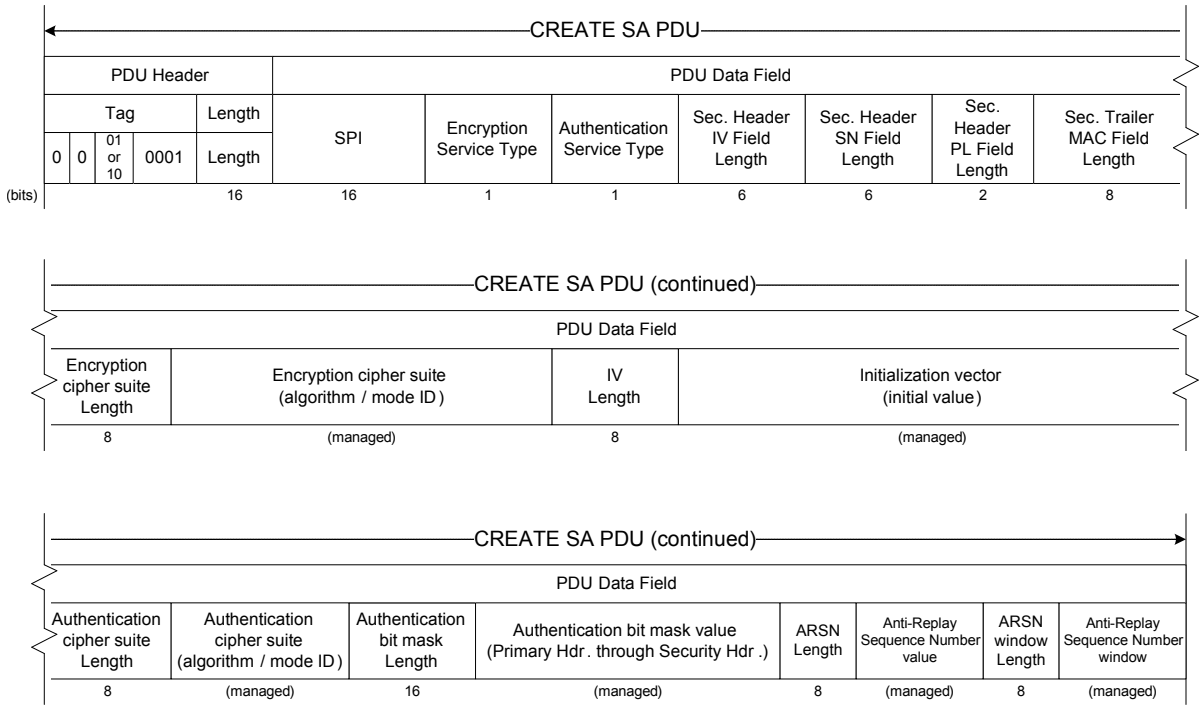

Figure 5-16: Create SA PDU

5.5.1.6.2.3 The SPI field shall signal the SPI of the Security Association to be created.

5.5.1.6.2.4 The Encryption Service Type flag shall signal that the SA to be created provides encryption service (1 = encryption; 0 = no encryption).

5.5.1.6.2.5 The Authentication Service Type flag shall signal that the SA to be created provides authentication service (1 = authentication; 0 = no authentication).

5.5.1.6.2.6 If the SA Service Type is Authenticated Encryption, both Encryption Service Type and Authentication Service Type flags shall be set.

5.5.1.6.2.7 The Security Header IV Field Length shall signal the length of the Initialization Vector field in the Security Header.

5.5.1.6.2.8 The Security Header SN Field Length shall signal the length of the Sequence Number field in the Security Header.

5.5.1.6.2.9 The Security Header PL Field Length shall signal the length of the Pad Length field in the Security Header.

5.5.1.6.2.10 The Security Trailer MAC Field Length shall signal the length of the MAC field in the Security Trailer.

5.5.1.6.2.11 The Encryption cipher suite length shall signal the length of the Encryption cipher suite field in the PDU.

5.5.1.6.2.12 The Encryption cipher suite identifier shall signal the encryption algorithm and mode of operation for the SA.

NOTE – The interpretation of the Encryption cipher suite identifier field is mission-specific. If more than one algorithm and mode are supported, the identifier should uniquely select which one is intended for use.

5.5.1.6.2.13 The Initialization vector length shall signal the length of the Initialization vector field in the PDU.

5.5.1.6.2.14 The IV value shall signal the initial managed value of the Initialization Vector for the SA.

5.5.1.6.2.15 The Authentication cipher suite length shall signal the length of the Authentication cipher suite field in the PDU.

5.5.1.6.2.16 The Authentication cipher suite identifier shall signal the authentication algorithm and mode of operation for the SA.

NOTE – The interpretation of the Authentication cipher suite identifier field is mission-specific. If more than one algorithm and mode are supported, the identifier should uniquely select which one is intended for use.

5.5.1.6.2.17 The Authentication bit mask length shall signal the length of the Authentication bit mask field in the PDU.

5.5.1.6.2.18 The Authentication bit mask shall signal the Authentication bit mask value for the SA.

5.5.1.6.2.19 The ARSN length shall signal the length of the Anti-Replay Sequence Number field in the PDU.

5.5.1.6.2.20 The ARSN value shall signal the initial value of the managed Anti-Replay Sequence Number for the SA.

5.5.1.6.2.21 The Anti-Replay Sequence Number window length shall signal the length of the Anti-Replay Sequence Number window field in the PDU.

5.5.1.6.2.22 The Anti-Replay Sequence Number window value shall signal the initial value of the managed Anti-Replay Sequence Number window for the SA.

5.5.1.7 Delete SA

5.5.1.7.1 General

The Delete SA Procedure shall support one Extended Procedures PDU data field structure:

Delete SA PDU.

5.5.1.7.2 Delete SA PDU

5.5.1.7.2.1 The Delete SA PDU shall be associated with the Delete SA Procedure, as defined in 3.3.3.6.

5.5.1.7.2.2 The Delete SA PDU shall consist of a single mandatory field:

SPI of the applicable Security Association (16 bits, mandatory).

NOTE – The format of the Delete SA PDU is shown in figure 5-17.

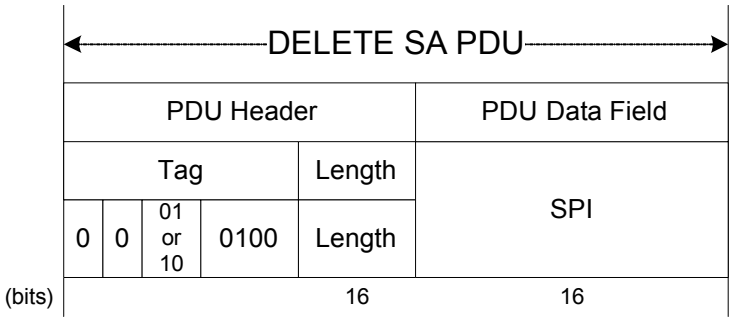

Figure 5-17: Delete SA PDU

5.5.1.7.2.3 The SPI field shall signal the SPI of the Security Association to be deleted.

5.5.1.8 Set Anti-Replay Sequence Number

5.5.1.8.1 General

The Set Anti-Replay Sequence Number Procedure shall support one Extended Procedures PDU data field structure:

Set Anti-Replay Sequence Number PDU.

5.5.1.8.2 Set Anti-Replay Sequence Number PDU

5.5.1.8.2.1 The Set Anti-Replay Sequence Number PDU shall be associated with the Set ARSN Procedure, as defined in 3.3.3.7.

5.5.1.8.2.2 The Set Anti-Replay Sequence Number PDU shall consist of a managed number of contiguously positioned mandatory fields:

a) SPI of the applicable Security Association (16 bits, mandatory);

b) new value of the Anti-Replay Sequence Number (managed length, mandatory).

NOTE – The format of the Set Anti-Replay Sequence Number PDU is shown in figure 5-18.

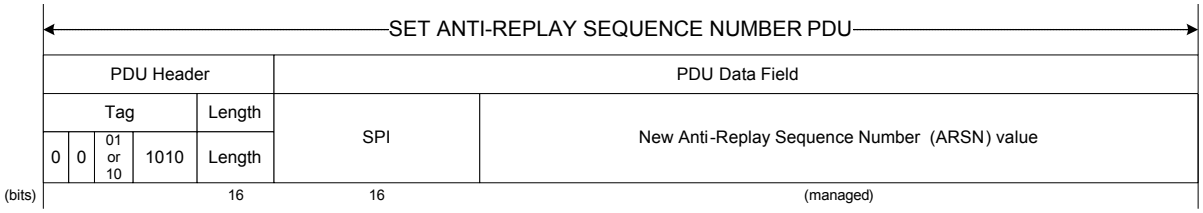

Figure 5-18: Set ARSN PDU

5.5.1.8.2.3 The SPI field shall signal the SPI of the Security Association whose Anti-Replay Sequence Number is to be modified.

5.5.1.8.2.4 The Anti-Replay Sequence Number field shall signal the new Anti-Replay Sequence Number value.

5.5.1.9 Set Anti-Replay Sequence Number Window

5.5.1.9.1 General

The Set Anti-Replay Sequence Number Procedure shall support one Extended Procedures PDU data field structure:

Set Anti-Replay Sequence Number Window PDU.

5.5.1.9.2 Set Anti-Replay Sequence Number Window PDU

5.5.1.9.2.1 The Set Anti-Replay Sequence Number Window PDU shall be associated with the Set Anti-Replay Sequence Number Window Procedure, as defined in 3.3.3.8.

5.5.1.9.2.2 The Set Anti-Replay Sequence Number Window PDU shall consist of a managed number of contiguously positioned mandatory fields:

a) SPI of the applicable Security Association (16 bits, mandatory);

b) new value of the Anti-Replay Sequence Number window (managed length, mandatory).

NOTE – The format of the Set Anti-Replay Sequence Number Window PDU is shown in figure 5-19.

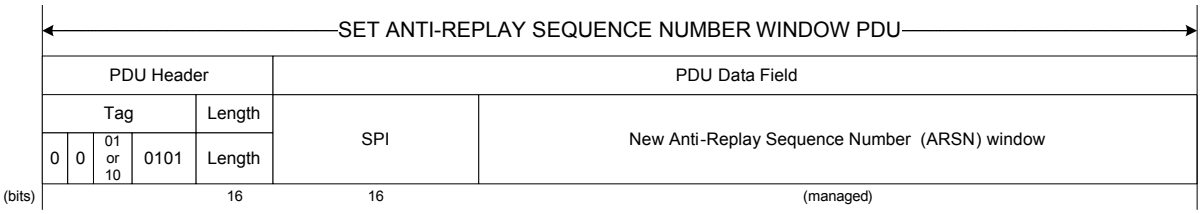

Figure 5-19: Set Anti-Replay Sequence Number Window PDU

5.5.1.9.2.3 The SPI field shall signal the SPI of the Security Association whose Anti-Replay Sequence Number window is to be modified.

5.5.1.9.2.4 The Anti-Replay Sequence Number Window field shall signal the new Anti-Replay Sequence Number window value.

5.5.1.10 SA Status Request

5.5.1.10.1 General

The SA Status Request Procedure shall support one Extended Procedures PDU data field structure:

SA Status Request PDU.

5.5.1.10.2 SA Status Request PDU

5.5.1.10.2.1 The SA Status Request PDU shall be associated with the SA Status Request Procedure, as defined in 3.3.3.9.

5.5.1.10.2.2 The SA Status Request PDU shall consist of a single mandatory field:

SPI of the applicable Security Association (16 bits, mandatory).

NOTE – The format of the SA Status Request PDU is shown in figure 5-20.

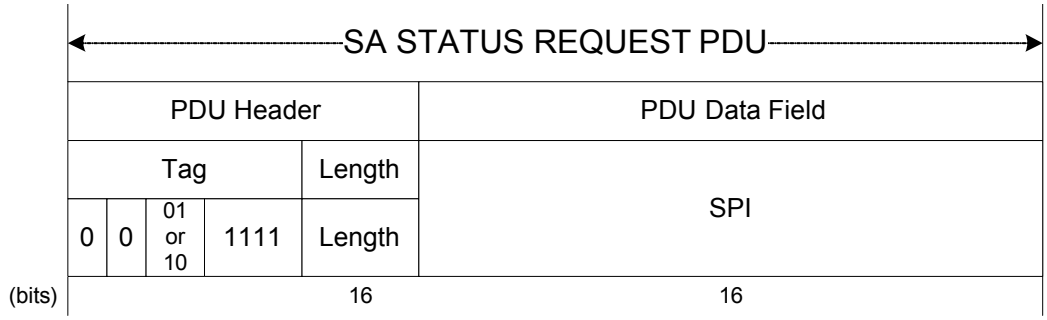

Figure 5-20: SA Status Request PDU

5.5.1.10.2.3 The SPI field shall signal the SPI of the Security Association to be queried.

5.5.1.10.3 SA Status Request Reply PDU

5.5.1.10.3.1 The SA Status Request Reply PDU shall be associated with the SA Status Request Procedure, as defined in 3.3.3.9.

5.5.1.10.3.2 The SA Status Request Reply PDU shall consist of two contiguously positioned mandatory fields:

a) SPI of the applicable Security Association (16 bits, mandatory);

b) procedure identification of the last executed state transition directive for the applicable Security Association (8 bits, mandatory).

NOTES

1 Within each SA state transition directive defined in this Recommended Standard, the previous (exited) state of the SA corresponds to the leftmost two bits of the Procedure Identification, and the current (entered) state of the SA corresponds to the rightmost two bits of the Procedure Identification.

2 The format of the SA Status Request Reply PDU is shown in figure 5-21.

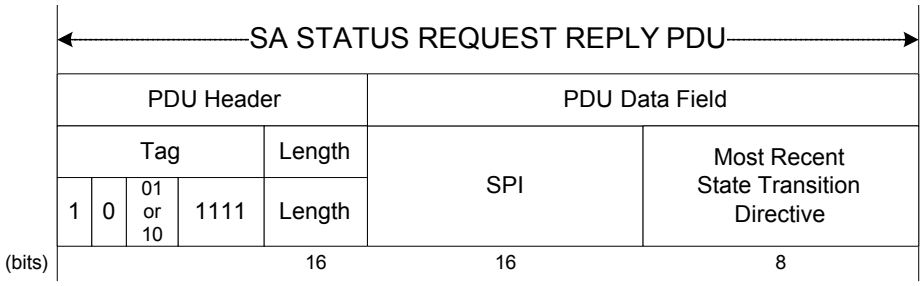

Figure 5-21: SA Status Request Reply PDU

5.5.1.10.3.3 The SPI field shall signal the SPI of the Security Association to be queried.

5.5.1.10.3.4 The State Transition Directive field shall signal the Procedure Identification of the last executed state transition directive for the applicable Security Association; or, if no previous state transition for the SA is known, the current state of the SA.

5.5.1.11 Read Anti-Replay Sequence Number

5.5.1.11.1 General

The Read Anti-Replay Sequence Number Procedure shall support two Extended Procedures PDU data field structures:

a) Read Anti-Replay Sequence Number Command PDU;

b) Read Anti-Replay Sequence Number Reply PDU.

5.5.1.11.2 Read Anti-Replay Sequence Number Command PDU

5.5.1.11.2.1 The Read Anti-Sequence Number Command PDU shall be associated with Step 3.3.3.10.2 of the Read Anti-Replay Sequence number Procedure, as defined in 3.3.3.10.

5.5.1.11.2.2 The Read Anti-Replay Sequence Number Command PDU shall consist of one mandatory data field:

SPI of the SA whose ARSN is to be read (16 bits, mandatory).

NOTE – The format of the Read Anti-Replay Sequence Number Command PDU is shown in figure 5-22.

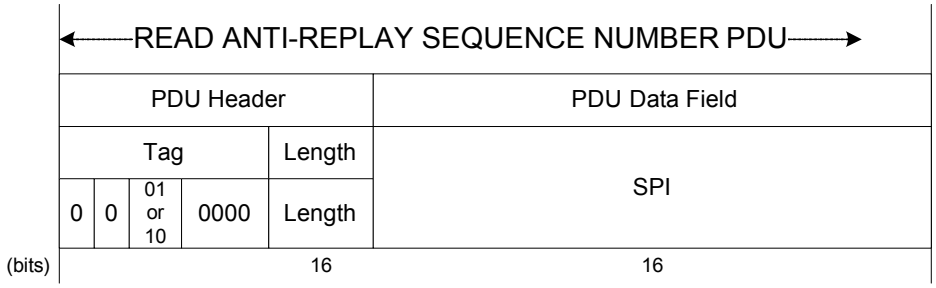

Figure 5-22: Read Anti-Replay Sequence Number Command PDU

5.5.1.11.3 Read Anti-Replay Sequence Number Reply PDU

5.5.1.11.3.1 The Read Anti-Replay Sequence Number Reply PDU shall be associated with Step 3.3.3.10.3.4 of the Read Anti-Replay Sequence Number Procedure, as defined in 3.3.3.10.

5.5.1.11.3.2 The Read Anti-Replay Sequence Number Reply PDU shall consist of one mandatory data field:

ARSN Value (managed length, mandatory).

NOTE – The format of the Read Anti-Replay Sequence Number Reply PDU is shown in figure 5-23.

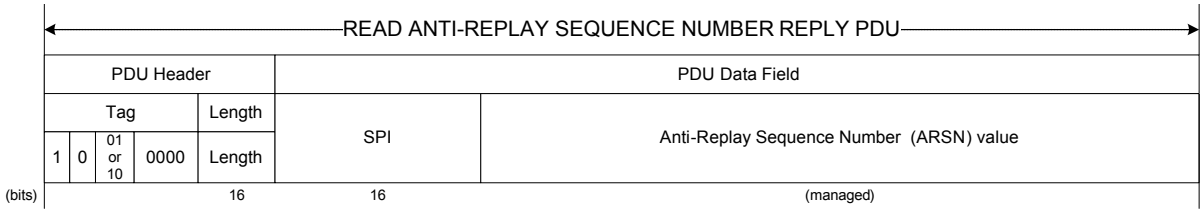

Figure 5-23: Read Anti-Replay Sequence Number Reply PDU

NOTE – The Anti-Replay Sequence Number Value field shall contain the full value of the Anti-Replay Sequence Number, without truncation.

5.6 SDLS MONITORING AND CONTROL

5.6.1 MONITORING AND CONTROL PROCEDURES

5.6.1.1 Ping

5.6.1.1.1 General

The Ping Command Procedure shall support two Extended Procedures PDU data field structures:

a) Ping Command PDU;

b) Ping Reply PDU.

5.6.1.1.2 Ping Command PDU

5.6.1.1.2.1 The Ping Command PDU shall be associated with Step 3.4.3.1.3.2 of the Ping Procedure, as defined in 3.4.3.1.

5.6.1.1.2.2 The Ping Command PDU shall have no data field.

NOTE – The format of the Ping Command PDU is shown in figure 5-24.

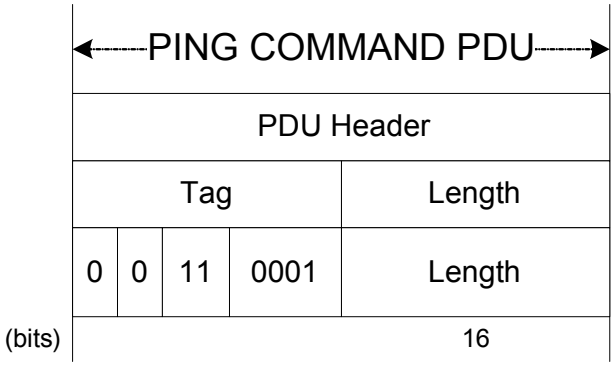

Figure 5-24: Ping Command PDU

5.6.1.1.3 Ping Reply PDU

5.6.1.1.3.1 The Ping Reply PDU shall be associated with Step 3.4.3.1.3.3 of the Ping Procedure defined in 3.4.3.1.

5.6.1.1.3.2 The Ping Reply PDU shall have no data field.

NOTE – The format of the Ping Reply PDU is shown in figure 5-25.

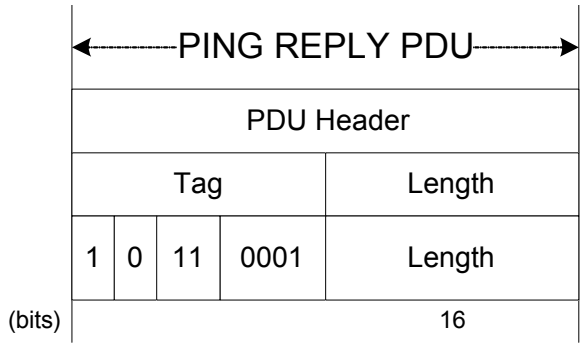

Figure 5-25: Ping Reply PDU

5.6.1.2 Log Status

5.6.1.2.1 General

The Log Status Procedure shall support two Extended Procedures PDU data field structures:

a) Log Status Command PDU;

b) Log Status Reply PDU.

5.6.1.2.2 Log Status Command PDU

5.6.1.2.2.1 The Log Status Command PDU shall be associated with Step 3.4.3.2.3.2 of the Log Status Procedure, as defined in 3.4.3.2.

5.6.1.2.2.2 The Log Status Command PDU shall have no data field.

NOTE – The format of the Log Status Command PDU is shown in figure 5-26.

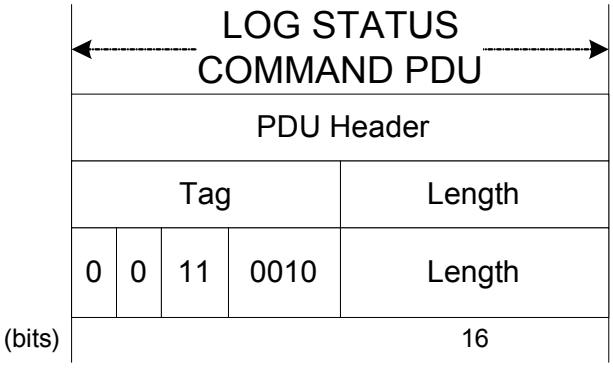

Figure 5-26: Log Status Command PDU

5.6.1.2.3 Log Status Reply PDU

5.6.1.2.3.1 The Log Status Reply PDU shall be associated with Step 3.4.3.2.3.4 of the Log Status procedure, as defined in 3.4.3.2.

5.6.1.2.3.2 The Log Status Reply PDU shall consist of two contiguously positioned mandatory fields:

a) number of events in the Security Log (managed length, mandatory);

b) remaining space in the Security Log (managed length, mandatory).

NOTE – The format of the Log Status Reply PDU is shown in figure 5-27.

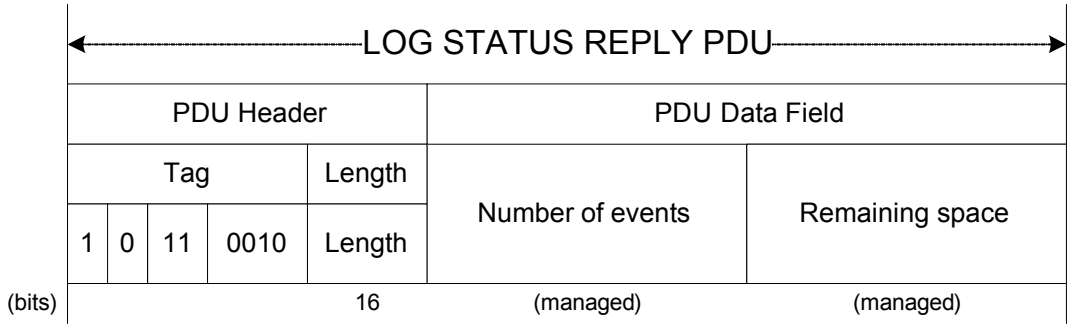

Figure 5-27: Log Status Reply PDU

5.6.1.3 Dump Log

5.6.1.3.1 General

The Dump Log Procedure shall support two Extended Procedures PDU data field structures:

a) Dump Log Command PDU;

b) Dump Log Reply PDU.

5.6.1.3.2 Dump Log Command PDU

5.6.1.3.2.1 The Dump Log Command PDU shall be associated with Step 3.4.3.3.3.2 of the Dump Log Procedure, as defined in 3.4.3.3.

5.6.1.3.2.2 The Dump Log Command PDU shall have no data field.

NOTE – The format of the Dump Log Command PDU is shown in figure 5-28.

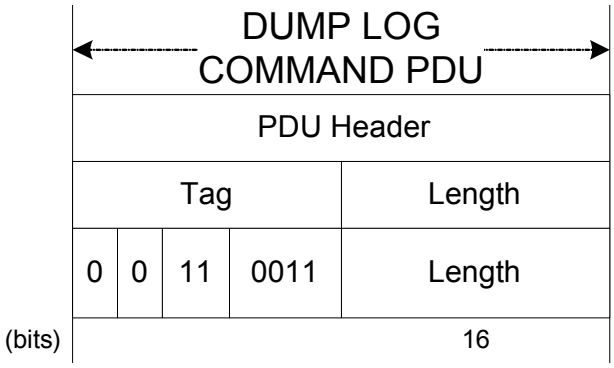

Figure 5-28: Dump Log Command PDU

5.6.1.3.3 Dump Log Reply PDU

5.6.1.3.3.1 The Dump Log Reply PDU shall be associated with Step 3.4.3.3.3.4 of the Dump Log Procedure, as defined in 3.4.3.3.

5.6.1.3.3.2 The Dump Log Reply PDU shall consist of a variable number of contiguously positioned fields:

Security Event Message (TLV formatted, T and L fields lengths managed).

NOTE – The format of the Dump Log Reply PDU is shown in figure 5-29.

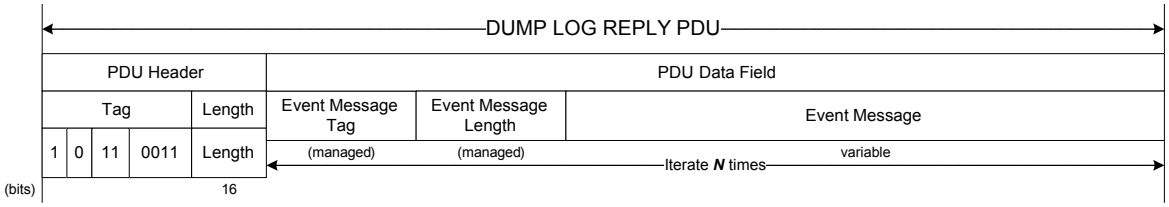

Figure 5-29: Dump Log Reply PDU

5.6.1.4 Erase Log

5.6.1.4.1 General

The Erase Log Procedure shall support two Extended Procedures PDU data field structures:

a) Erase Log Command PDU;

b) Erase Log Reply PDU.

5.6.1.4.2 Erase Log Command PDU

5.6.1.4.2.1 The Erase Log Command PDU shall be associated with Step 3.4.3.4.3.2 of the Erase Log Procedure, as defined in 3.4.3.4.

5.6.1.4.2.2 The Erase Log Command PDU shall have no data field.

NOTE – The format of the Erase Log Command PDU is shown in figure 5-30.

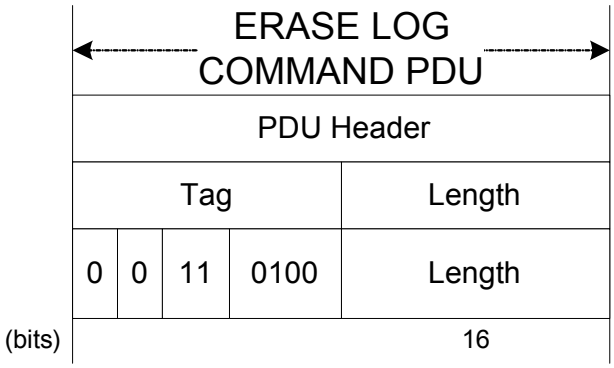

Figure 5-30: Erase Log Command PDU

5.6.1.4.3 Erase Log Reply PDU

5.6.1.4.3.1 The Erase Log Reply PDU shall be associated with Step 3.4.3.4.3.4 of the Erase Log Procedure, as defined in 3.4.3.4.

5.6.1.4.3.2 The Erase Log Reply PDU shall consist of two contiguously positioned mandatory fields:

a) number of events in the Security Log after erasure of the Security Log (managed length, mandatory);

b) remaining space in the Security Log after erasure of the Security Log (managed length, mandatory).

NOTE – The format of the Erase Log Reply PDU is shown in figure 5-31.

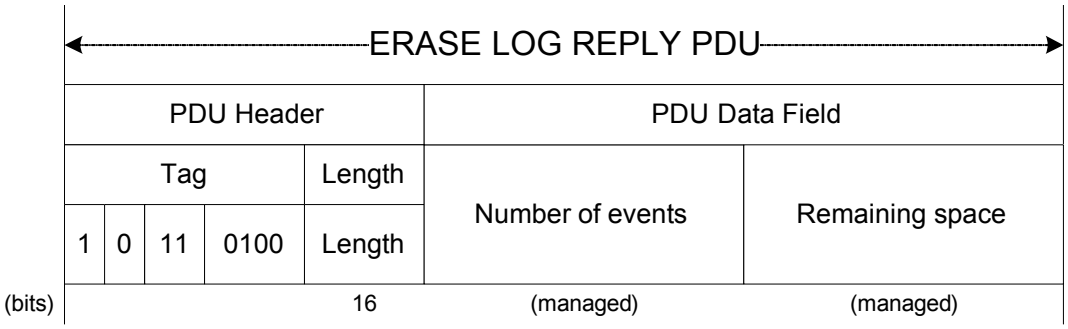

Figure 5-31: Erase Log Reply PDU

5.6.1.5 Self-Test

5.6.1.5.1 General

The Self-Test Procedure shall support two Extended Procedures PDU data field structures:

a) Self-Test Command PDU;

b) Self-Test Reply PDU.

5.6.1.5.2 Self-Test Command PDU

5.6.1.5.2.1 The Self-Test Command PDU shall be associated with Step 3.4.3.5.3.2 of the Self-Test Procedure, as defined in 3.4.3.5.

5.6.1.5.2.2 The Self-Test Command PDU shall have no data field.

NOTE – The format of the Self-Test Command PDU is shown in figure 5-32.

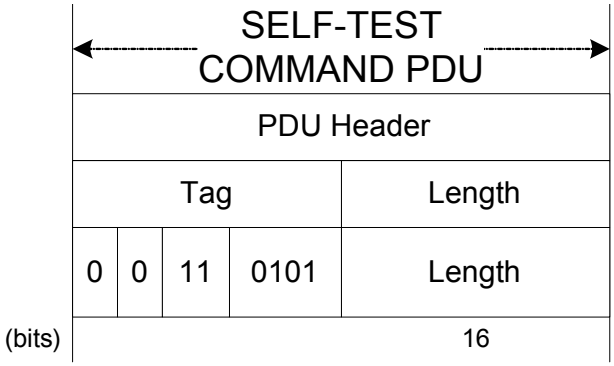

Figure 5-32: Self-Test Command PDU

5.6.1.5.3 Self-Test Reply PDU

5.6.1.5.3.1 The Self-Test Reply PDU shall be associated with Step 3.4.3.5.3.4 of the Self-Test Procedure, as defined in 3.4.3.5.

5.6.1.5.3.2 The Self-Test Reply PDU shall consist of one mandatory data field:

Self-Test result:

1) 0XXXXXXXb: Self-Test OK (8-bit length);

2) 1XXXXXXXb: Self-Test not OK (8-bit length).

NOTES

1 The bits having the value 'X' in the above definitions are not used by CCSDS. Their values are left to the implementer's choice.

2 The format of the Self-Test Reply PDU is shown in figure 5-33.

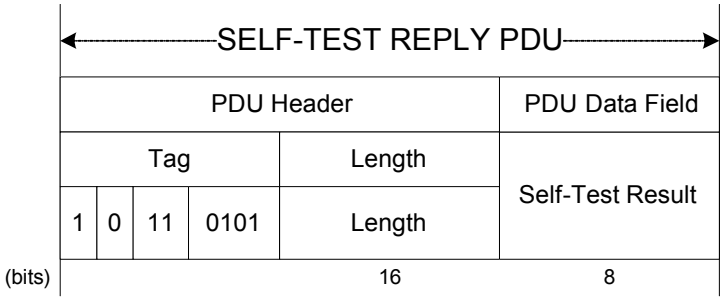

Figure 5-33: Self-Test Reply PDU

5.6.1.6 Alarm Flag Reset

5.6.1.6.1 General

The Alarm Flag Reset Procedure shall support one Extended Procedures PDU data field structures:

Alarm Flag Reset Command PDU.

5.6.1.6.2 Alarm Flag Reset Command PDU

5.6.1.6.2.1 The Alarm Flag Reset Command PDU shall be associated with Step 3.4.3.6.2.2 of the Alarm Flag Reset Procedure, as defined in 3.4.3.6.

5.6.1.6.2.2 The Alarm Flag Reset Command PDU shall have no data field.

NOTE – The format of the Alarm Flag Reset Command PDU is shown in figure 5-34.

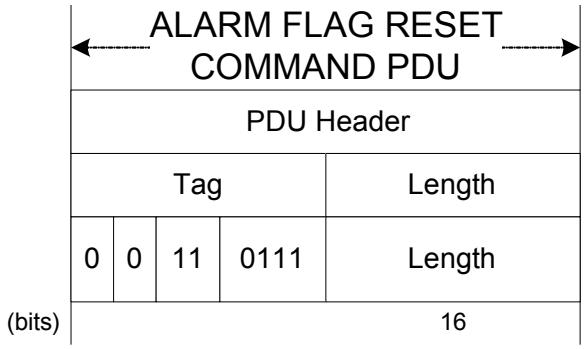

Figure 5-34: Alarm Flag Reset Command PDU

6 MANAGED PARAMETERS

6.1 OVERVIEW

In order to conserve bandwidth on the space link, certain parameters associated with the Security Protocol are handled by management rather than by inline communications protocol. The managed parameters are generally those which tend to be static for long periods of time, and whose change signifies a major reconfiguration of the service provider associated with a particular mission. These managed parameters are intended to be included in any service-provider system that manages Security Associations, but no specification for such a management system is provided or implied.

6.2 REQUIREMENTS

6.2.1 The managed parameters used for the SDLS Extended Procedures shall be those listed in table 6-1.

NOTES

1 These parameters are defined in an abstract sense, and are not intended to imply any particular implementation of a management system.

2 The majority of managed parameters are the parameters of the SA data base managed by both the sending and receiving ends, which must match one another in order to operate correctly.

6.2.2 All managed parameters of the Space Data Link Protocol (see references [4], [5], [9], and [6]) used on the physical channel shall be treated as also applicable to the SDLS Extended Procedures.

Table 6-1: Managed Parameters for SDLS Extended Procedures

Managed Parameter	Allowed Values	Defined In Reference
Key Management managed parameters:		
Key State	Pre-active, Active, Deactivated, Destroyed	[2]
OTAR Number of Keys to be uploaded	Integer greater than zero (> 0)	
OTAR IV length	Algorithm-specific	
OTAR Upload Session Key Size	Algorithm-specific	
OTAR Cryptographic Algorithm	AES/Counter Mode, GCM, Mission-specific	[7]
Key Verification IV length	Algorithm-specific	

Managed Parameter	Allowed Values	Defined In Reference
Key ID Field Size	Mission-specific	
Key State Field Size	Mission-specific	
OTAR MAC Size	Algorithm-specific	
Key Verification Challenge Length	Algorithm-specific	
Key Verification MAC Size	Algorithm-specific	
SA Management managed parameters:		
SPI	1-65534	[1]
SA Service Type	Authentication Encryption, Authenticated Encryption	[1]
Length for Security Header IV field	1-32 octets	[1]
Length for Security Header SN field	2-8 octets	[1]
Length for Security Header PL field	1-2 octets	[1]
Length for Security Trailer MAC field	8-64 octets	[1]
Encryption cipher suite identifier	Mission-specific	
IV length	Algorithm-specific	[1]
IV value	Algorithm-specific	[1]
Authentication cipher suite identifier	Mission-specific	
Authentication bit mask length	Size of SLP transfer frame	[1]
Authentication bit mask value	Binary mask	[1]
ARSN length	Algorithm-specific	[1]
ARSN value	Integer	[1]
ARSN Window length	Algorithm-specific	[1]
ARSN Window value	Integer greater than zero (> 0)	[1]
Encryption Key ID	Mission-specific	
Authentication Key ID	Mission-specific	
Global Virtual Channel ID(s) Global MAP ID(s)		[4], [5], [6], [9]
Monitoring & Control managed parameters:		
Number of events (in Security Log) field length	Mission-specific	
Remaining Space (in Security Log) field length	Mission-specific	

7 CONFORMANCE REQUIREMENTS

An implementer of the Extended Procedures shall verify conformance with this Recommended Standard by completing a Protocol Implementation Conformance Statement (PICS) for their implementation based on the CCSDS-defined PICS proforma for the protocol.

NOTE – A compliant PICS proforma is provided in annex A of this document.

ANNEX A

PROTOCOL IMPLEMENTATION CONFORMANCE STATEMENT PROFORMA

(NORMATIVE)

A1 INTRODUCTION

A1.1 OVERVIEW

To evaluate conformance of a particular implementation, it is necessary to have a statement of which capabilities and options have been implemented for a given protocol specification. Such a statement is called a Protocol Implementation Conformance Statement. This annex provides the PICS proforma for the Space Data Link Security Extended Procedures in compliance with the relevant requirements, and in accordance with the relevant guidance given in ISO/IEC 9646-7.

A1.2 CONFORMANCE TO THIS PICS PROFORMA

If it is claimed to conform to this Recommended Standard, the actual PICS proforma to be filled in by a supplier shall be technically equivalent to the text of the PICS proforma in this annex and shall preserve the numbering/naming and ordering of the PICS proforma items. A PICS that conforms to this Recommended Standard shall be a conforming PICS proforma completed in accordance with the instructions for completion given in A2.

A1.3 COPYRIGHT

Users of this Recommended Standard may freely reproduce this PICS proforma so that it can be used for its intended purpose and may further publish the completed PICS.

A2 INSTRUCTIONS FOR COMPLETING THE PICS PROFORMA

A2.1 OVERVIEW

In order to reduce the size of tables in the PICS proforma, notations have been introduced that have allowed the use of a multi-column layout, in which the columns are headed ‘Status’ and ‘Support’. The definition of each of these follows.

A2.2 STATUS COLUMN

The 'Status' column indicates the level of support required for conformance to the standard. The values are as follows:

M Mandatory support is required.

O Optional support is permitted for conformance to the standard. If implemented, it must conform to the specifications and restrictions contained in the standard. These restrictions may affect the optionality of other items.

O.*n* The item is optional, but support of at least one of the options labeled with the same number *n* is mandatory. The definitions for the qualification statements used in this annex are written under the tables in which they appear.

C.*n* The item is conditional (where *n* is the number which identifies the applicable condition). The definitions for the conditional statements used in this annex are written under the tables in which they appear.

n/a The item is not applicable.

A2.3 SUPPORT COLUMN

The 'Support' column shall be completed by the supplier or implementer to indicate the level of implementation of each feature. The proforma has been designed such that the only entries required in the 'Support' column are the following:

Y Yes, the feature has been implemented.

N No, the feature has not been implemented.

– The item is not applicable.

A2.4 ITEM REFERENCE NUMBERS

Each line within the PICS proforma that requires implementation detail to be entered is numbered at the left-hand edge of the line. This numbering is included as a means of uniquely identifying all possible implementation details within the PICS proforma. The need for such unique referencing has been identified by the testing bodies.

The means of referencing individual responses should be to specify the following sequence:

a) a reference to the smallest subsection enclosing the relevant item;

b) a solidus character, '/';

c) the reference number of the row in which the response appears;

d) if, and only if, more than one response occurs in the row identified by the reference number, then each possible entry is implicitly labeled a, b, c, etc., from left to right, and this letter is appended to the sequence.

An example of the use of this notation would be A4/1, which refers to the SDLS implementation's Key Management Security Service.

A2.5 COMPLETION OF THE PICS

The implementer shall complete all entries in the column marked 'Support'. In certain clauses of the PICS proforma, further guidance for completion may be necessary. Such guidance shall supplement the guidance given in this clause and shall have a scope restricted to the clause in which it appears. In addition, other specifically identified information shall be provided by the implementer when requested. No changes shall be made to the proforma except the completion as required. Recognizing that the level of detail required may, in some instances, exceed the space available for responses, a number of responses specifically allow for the addition of appendices to the PICS.

A3 GENERAL INFORMATION

A3.1 REFERENCED BASE STANDARDS

The SDLS Extended Procedures (this Recommended Standard) is the only base standard referenced in this PICS proforma. In the tables below, numbers in the Reference column refer to applicable subsections within this document.

A3.2 IDENTIFICATION OF THE PICS

Date of statement (yyyy-mm-dd)	
PICS version	
System Conformance Statement cross-reference	
Other information	

NOTE – The System Conformance Statement is identified in ISO/IEC 9646-7 (reference [**C**11]). It contains a declaration of the layers of the Reference Model covered by the implementation to be tested.

A3.3 IDENTIFICATION OF THE SYSTEM SUPPLIER AND/OR TEST LABORATORY CLIENT

Organization name	
Contact name	
Address	
Telephone	
Email	
Other information	

A3.4 IDENTIFICATION OF THE IMPLEMENTATION UNDER TEST

Implementation name	
Implementation version	
Machine name	
Machine version	
Operating system name	
Operating system version	
Special configuration	
Other information	

A3.5 IDENTIFICATION OF THE PROTOCOL

Protocol specification/version	
Technical corrigenda implemented	
Other amendments implemented (explain)	

A3.6 GLOBAL STATEMENT OF CONFORMANCE

Are all mandatory features implemented? (Yes or No)	

NOTE – If a 'No' answer is given to this question, then the implementation does not conform to the SDLS Extended Procedures standard. Non-supported mandatory capabilities are to be identified in the PICS, with an explanation of why the implementation is non-conforming.

Non-conforming capabilities (explain)	

A4 SUPPORTED SECURITY SERVICES

Item	Protocol Feature	Reference	Status	Support
1	Key Management		O.1	
2	SA Management		O.1	
3	Monitoring & Control		O.1	
	O.1: Support for at least one of [A4/1 | A4/2 | A4/3] is M			

A5 SERVICE PRIMITIVES

A5.1 KEY MANAGEMENT SERVICE PRIMITIVES

Item	Protocol Feature	Reference	Status	Support
1	OTAR		M	
2	Key Activation		M	
3	Key Deactivation		M	
4	Key Destruction		O	
5	Key Verification		M	
6	Key Inventory		M	

A5.2 SA MANAGEMENT SERVICE PRIMITIVES

Item	Protocol Feature	Reference	Status	Support
1	Start SA		M	
2	Stop SA		M	
3	Rekey SA		O	
4	Expire SA		C.2	
5	Create SA		O	
6	Delete SA		C.3	
7	Set ARSN		O	
8	Set ARSNW		O	
9	SA Status Request		O	
10	Read Sequence Number		M	
	C.2: if [A5.2/3] is supported then M, else O C.3: if [A5.2/5] is supported then M, else n/a			

A5.3 MONITORING & CONTROL SERVICE PRIMITIVES

Item	Protocol Feature	Reference	Status	Support
1	Ping		M	
2	Log Status		C.1	
3	Dump Log		C.1	
4	Erase Log		C.1	
5	Self-test		M	
6	Alarm Flag Reset		M	
	C.1: if Security Log implemented then Items 2, 3, 4 mandatory			

A6 PROTOCOL DATA UNITS

A6.1 PDU HEADER

Item	Protocol Feature	Reference	Status	Support
1	Procedure Type		M	
2	User Flag		M	
3	Service Group		M	
4	Procedure Identification		M	
5	Length		M	
6	PDU Data Field		O	

A6.2 OTAR COMMAND PDU DATA FIELD

Item	Protocol Feature	Reference	Status	Support
1	Master Key ID		M	
2	Initialization Vector		O	
3	Encrypted Key ID		M	
4	Encrypted Key		M	

A6.3 KEY ACTIVATION COMMAND PDU DATA FIELD

Item	Protocol Feature	Reference	Status	Support
1	Key ID		M	

A6.4 KEY DEACTIVATION COMMAND PDU DATA FIELD

Item	Protocol Feature	Reference	Status	Support
1	Key ID		M	

A6.5 KEY DESTRUCTION COMMAND PDU DATA FIELD

Item	Protocol Feature	Reference	Status	Support
1	Key ID		M	

A6.6 KEY VERIFICATION PDU DATA FIELD

Item	Protocol Feature	Reference	Status	Support
1	Set of Key IDs		M	
2	Set of Challenges		M	
3	Set of Challenge Responses		M	

A6.7 KEY INVENTORY PDU DATA FIELD

Item	Protocol Feature	Reference	Status	Support
1	Range of Key IDs		M	
2	Set of (Key ID, Key States)		M	

A6.8 START SA PDU DATA FIELD

Item	Protocol Feature	Reference	Status	Support
1	SPI		M	
2	GVCID/GMAP ID		M	

A6.9 STOP SA PDU DATA FIELD

Item	Protocol Feature	Reference	Status	Support
1	SPI		M	

A6.10 REKEY SA PDU DATA FIELD

Item	Protocol Feature	Reference	Status	Support
1	SPI		M	
2	ARSN		M	
3	Encryption Key ID		O.1	
4	Authentication Key ID		O.1	
	O.1: Support for at least one of [A4/3 | A4/4] is M			

A6.11 EXPIRE SA PDU DATA FIELD

Item	Protocol Feature	Reference	Status	Support
1	SPI		M	

A6.12 CREATE SA PDU DATA FIELD

Item	Protocol Feature	Reference	Status	Support
1	SPI		M	
2	Encryption Key ID		M	
3	Authentication Key ID		M	
4	SH IV Length		M	
5	SH SN Length		M	
6	SH PL Length		M	
7	ST MAC Length		M	
8	Encr. Cipher Suite Length		M	
9	Encryption Cipher Suite		M	
10	IV Length		M	
11	IV		M	
12	Auth. Cipher Suite Length		M	
13	Authentication Cipher Suite		M	
14	Auth. Bit Mask Length		M	
15	Authentication Bit Mask		M	
16	ARSN Length		M	
17	ARSN		M	
18	ARSNW Length		M	
19	ARSNW		M	

A6.13 DELETE SA PDU DATA FIELD

Item	Protocol Feature	Reference	Status	Support
1	SPI		M	

A6.14 SET ARSN PDU DATA FIELD

Item	Protocol Feature	Reference	Status	Support
1	SPI		M	
2	ARSN		M	

A6.15 SET ARSNW PDU DATA FIELD

Item	Protocol Feature	Reference	Status	Support
1	SPI		M	
2	ARSNW		M	

A6.16 SA STATUS REQUEST PDU DATA FIELD

Item	Protocol Feature	Reference	Status	Support
1	SPI		M	

A6.17 READ SEQUENCE NUMBER REPLY PDU DATA FIELD

Item	Protocol Feature	Reference	Status	Support
1	Anti-Replay Sequence Number Value		M	

A6.18 SA STATUS REQUEST REPLY PDU DATA FIELD

Item	Protocol Feature	Reference	Status	Support
1	SPI		M	
2	Last State Transition		M	

A6.19 PING COMMAND PDU DATA FIELD

None.

A6.20 PING REPLY PDU DATA FIELD

None.

A6.21 LOG STATUS COMMAND PDU DATA FIELD

None.

A6.22 LOG STATUS REPLY PDU DATA FIELD

Item	Protocol Feature	Reference	Status	Support
1	Event Message Tag		M	
2	Event Message Length		M	
3	Event Message Value		M	

A6.23 ERASE LOG COMMAND PDU DATA FIELD

None.

A6.24 LOG STATUS REPLY PDU DATA FIELD

Item	Protocol Feature	Reference	Status	Support
1	Number of events		M	
2	Remaining Space		M	

A6.25 SELF-TEST COMMAND PDU DATA FIELD

None.

A6.26 SELF-TEST REPLY PDU DATA FIELD

Item	Protocol Feature	Reference	Status	Support
1	Self-Test Result		M	

A6.27 ALARM-FLAG RESET COMMAND PDU DATA FIELD

None.

ANNEX B

SECURITY, SANA, AND PATENT CONSIDERATIONS

(INFORMATIVE)

B1 SECURITY CONSIDERATIONS

B1.1 SECURITY CONCERNS WITH RESPECT TO THE CCSDS DOCUMENT

B1.1.1 General

Communications security attempts to ensure the confidentiality, integrity, and/or authenticity of transmitted data, as required depending on the threat, the mission security policy(s), and the desire of the mission planners. It is possible for a single data unit to require all three of these security attributes to ensure that the transmitted data is not disclosed, not altered, and not spoofed. This Recommended Standard supports the management of communication security as established by the SDLS protocol.

B1.1.2 Data Privacy

This Recommended Standard does not address specific protection related to potential sensitivity of certain SDLS Extended Procedures PDUs. This is up to the implementer and is discussed in the SDLS EP Green Book (reference [**C**13]).

B1.1.3 Data Integrity

This Recommended Standard specifies in 4.3.1 that SDLS Extended Procedures PDUs be protected at least by authentication during their transfer over the space link, therefore providing data integrity.

B1.1.4 Authentication of Communicating Entities

This Recommended Standard relies on the SDLS protocol to properly authenticate the communicating entities.

B1.2 POTENTIAL THREATS AND ATTACK SCENARIOS

The same considerations as for the SDLS protocol (reference [1]) apply here. Specific potential threats and attack scenarios are addressed in more detail in reference [**C**12].

B1.3 CONSEQUENCES OF NOT APPLYING SECURITY TO THE TECHNOLOGY

Without authentication, unauthorized extended procedures or software might be uploaded to a spacecraft. Without data integrity, corrupted extended procedures might be uploaded to a spacecraft potentially resulting in the loss of the security capabilities or, in the worst case, the mission. Without confidentiality, session keys and the data field contents of sensitive extended procedures PDUs may be disclosed to an attacker.

B2 SANA CONSIDERATIONS

This Recommended Standard defines no new information registries. The recommendations of this document do not require any action from SANA.

B3 PATENT CONSIDERATIONS

At the time of publication, CCSDS was not aware of any claimed patent rights applicable to implementing the provisions of this Recommended Standard.

ANNEX C

INFORMATIVE REFERENCES

(INFORMATIVE)

[**C**1] *Information Technology—Open Systems Interconnection—Basic Reference Model: The Basic Model*. 2nd ed. International Standard, ISO/IEC 7498-1:1994. Geneva: ISO, 1994.

[**C**2] *Information Processing Systems—Open Systems Interconnection—Basic Reference Model—Part 2: Security Architecture*. International Standard, ISO 7498-2:1989. Geneva: ISO, 1989.

[**C**3] *Space Data Link Security Protocol—Summary of Concept and Rationale*. Report Concerning Space Data System Standards (Green Book), CCSDS 350.5-G-1. Washington, D.C.: CCSDS, June 2018.

[**C**4] *The Application of Security to CCSDS Protocols*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 350.0-G-3. Washington, D.C.: CCSDS, March 2019.

[**C**5] *Security Architecture for Space Data Systems*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 351.0-M-1. Washington, D.C.: CCSDS, November 2012.

[**C**6] *Space Missions Key Management Concept*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 350.6-G-1. Washington, D.C.: CCSDS, November 2011.

[**C**7] *Overview of Space Communications Protocols*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 130.0-G-3. Washington, D.C.: CCSDS, July 2014.

[**C**8] *National Information Assurance (IA) Glossary*. Revised. CNSSI No. 4009. Fort Meade, Maryland: CNSS, April 6, 2015.

[**C**9] *Glossary of Key Information Security Terms*. Rev. 2. Edited by Richard Kissel. NIST IR 7298. Gaithersburg, Maryland: NIST, May 2013.

[**C**10] Elaine Barker. *Recommendation for Key Management—Part 1: General*. Revision 4. National Institute of Standards and Technology Special Publication 800-57. Gaithersburg, Maryland: NIST, January 2016.

[**C**11] *Information Technology—Open Systems Interconnection—Conformance Testing Methodology and Framework—Part 7: Implementation Conformance Statements*. International Standard, ISO/IEC 9646-7:1995. Geneva: ISO, 1995.

[**C**12] *Security Threats against Space Missions*. Issue 2. Report Concerning Space Data System Standards (Green Book), CCSDS 350.1-G-2. Washington, D.C.: CCSDS, December 2015.

[**C**13] *Space Data Link Security Protocol—Extended Procedures—Summary of Concept of Rationale*. Report Concerning Space Data System Standards. Forthcoming.

ANNEX D

BASELINE IMPLEMENTATION MODE

(INFORMATIVE)

D1 FRAME SECURITY REPORT

D1.1 OVERVIEW

This annex specifies the baseline implementation mode for the FSR and its integration into the transfer service protocol.

D1.2 TRANSFER SERVICE INTERFACE

D1.2.1 FSR Use with TM, AOS, or USLP

The baseline implementation mode to be used for integrating the FSR into the TM, AOS, or USLP transfer service is as follows:

a) The FSR is reported as Operational Control Field (OCF Type 2) (see also 4.2.2.3.2).

b) In case COP-1 is reporting on the same virtual channel, the FSR reporting alternates with the Command Link Control Word (OCF Type 1) reporting.

D1.2.2 Interface with SDLS

The SDLS implementation used for the SDLS Extended Procedures is the SDLS baseline implementation mode (see reference [1], annex E).

D2 PROTOCOL DATA UNITS

The baseline implementation of the TLV format does not make use of nesting.

D3 RESERVED SPIs/SAs

The baseline implementation mode uses the two reserved SPIs for exchanging EP Service PDUs as specified in 4.3.

These two reserved SPIs are ‘all zeros’ (0) and ‘all ones’ (65535) (see also reference [1]).

D4 KEY MANAGEMENT SERVICE

D4.1 GENERAL

NOTE – This subsection specifies the baseline implementation mode for the Key Management Service Extended Procedures.

The baseline implementation mode does not include the Key Destruction and Key Inventory Procedures.

D4.2 SECURITY ALGORITHM AND KEY CONFIGURATION

D4.2.1 General

The following security algorithm configuration is used to support the Key Management Service Extended Procedures in the baseline implementation mode.

D4.2.2 Algorithm for OTAR

The baseline implementation to be used for OTAR operation is as follows:

a) for authenticated encryption of the key block: AES-GCM, as defined in reference [7];

b) the keys are 256 bits in total length;

c) the IV is 96 bits in total length;

d) the output MAC for the authentication is 128 bits in length.

D4.2.3 Algorithm for Key Verification

The baseline implementation to be used for Key Verification is as follows:

a) for the authenticated encryption of the challenge: AES-GCM, as defined in reference [7];

b) the keys are 256 bits in total length;

c) the IV is 96 bits in total length;

d) the output MAC for the authentication is 128 bits in length.

D4.3 KEY MANAGEMENT SERVICES PARAMETERS

D4.3.1 OTAR

The baseline implementation configuration to be used for OTAR procedure (3.2.3.1 and 5.4.2.1) interoperability testing and operation is as follows:

a) The Master Key ID field of the OTAR Command PDU has a size of 16 bit.

NOTE – It is up to the implementer to decide if master keys are assigned a special range from the total key ID range.

b) The Initialization Vector field of the OTAR Command PDU has a size of 96 bits.

c) Each Encrypted Key Block of the OTAR Command PDU has a size of 272 bits, consisting of

1) the Key ID fields (16 bits), and

2) the Session Key fields (256 bits).

d) The MAC field of the OTAR Command PDU has a size of 128 bits.

e) The total length of the PDU (Header + Data Field) does not exceed 995 octets (=upload of 28 session keys). This is to ensure that the complete Command PDU fits into one frame of 1024 octets.

f) The Length field indicates a number of $N \times 272 + 240$, where N is the number of session keys to be uploaded, $N \leq 28$.

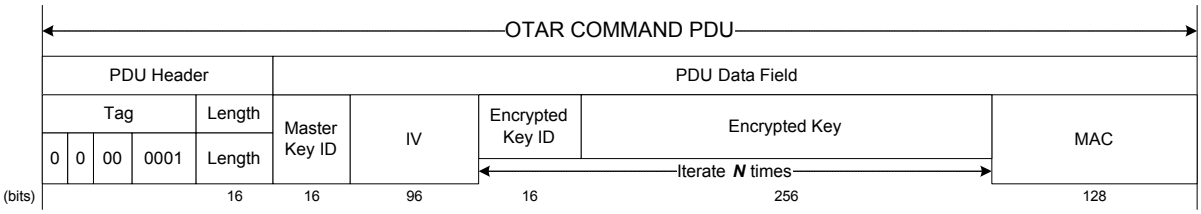

Figure D-1: Baseline Implementation Mode OTAR Command PDU

D4.3.2 Key Activation

The baseline implementation configuration to be used for Key Activation procedure (3.2.3.2 and 5.4.2.2) interoperability testing and operation is as follows:

a) The Key ID fields of the Key Activation PDU data field structure has a size of 16 bits.

b) The total length of the PDU (Header + Data Field) does not exceed 35 octets (=activation of 16 session keys).

c) The Length field indicates a number of $N \times 16$, where N is the number of session keys to be activated, $N \leq 16$.

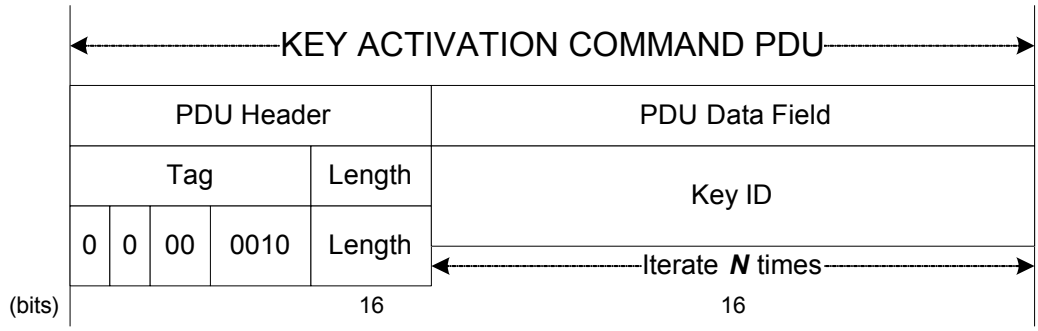

Figure D-2: Baseline Implementation Mode Key Activation Command PDU

D4.3.3 Key Deactivation

The baseline implementation configuration to be used for Key Deactivation procedure (3.2.3.3 and 5.4.2.3) interoperability testing and operation is as follows:

a) The Key ID fields of the Key Deactivation PDU data field structure has a size of 16 bits.

b) The total length of the PDU (Header + Data Field) does not exceed 35 octets (=deactivation of 16 session keys).

c) The Length field indicates a number of $N \times 16$, where N is the number of session keys to be deactivated, $N \leq 16$.

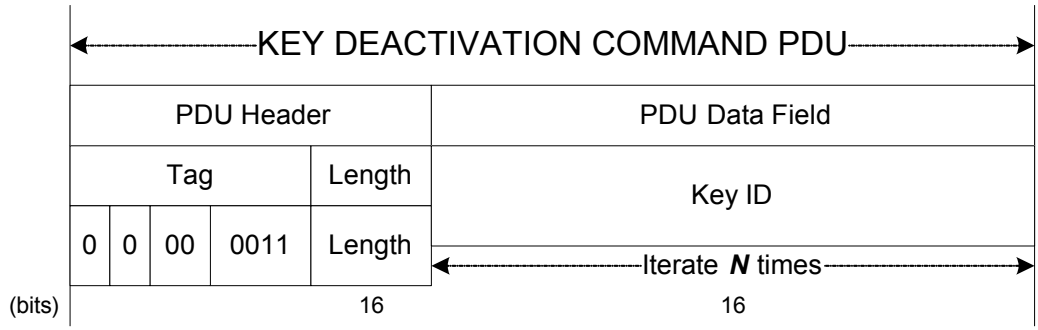

Figure D-3: Baseline Implementation Mode Key Deactivation Command PDU

D4.3.4 Key Verification

The baseline implementation configuration to be used for Key Verification procedure (3.2.3.5 and 5.4.2.5) interoperability testing and operation is as follows:

a) The Key Verification command does not require a Master Key ID.

b) The Key ID fields of the Key Verification Command PDU has a size of 16 bits. Values 0–127 are not used to reference session keys.

c) The Challenge fields of the Key Verification Command PDU has a size of 128 bits.

d) The total length of the Command and Replay PDUs (Header + Data Field) does not exceed 995 octets (=verification of 21 session keys). This is to ensure that the complete Reply and Command PDUs fit into one frame of 1024 octets.

e) The Length field of the Command PDU indicates a number of $N \times 18$, where N is the number of session keys to be verified, $N \leq 21$.

f) The Key Verification reply PDU does not require a Master Key ID.

g) The Key ID fields of the Key Verification Reply PDU has a size of 16 bits. Values 0–127 are not used to reference session keys.

h) The Encrypted Challenge fields of the Key Verification Reply PDU has a size of 128 bits.

i) The Challenge MAC fields of the Key Verification Reply PDU has a size of 128 bits.

j) The IV fields of the Key Verification Reply PDU has a size of 96 bits.

k) The Length field of the Reply PDU indicates a number of $N \times 46$, where N is the number of session keys to be verified, $N \leq 21$.

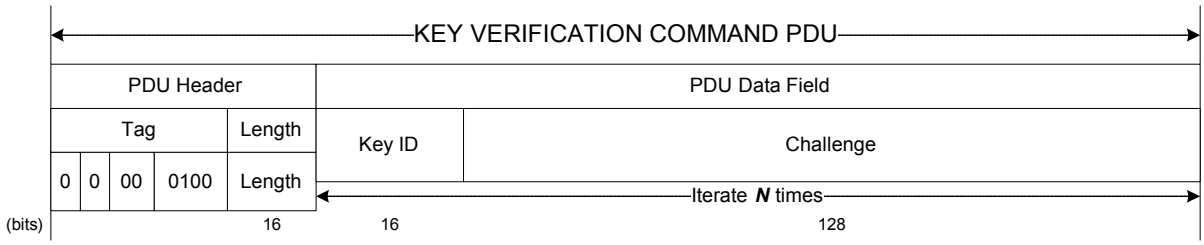

Figure D-4: Baseline Implementation Mode Key Verification Command PDU

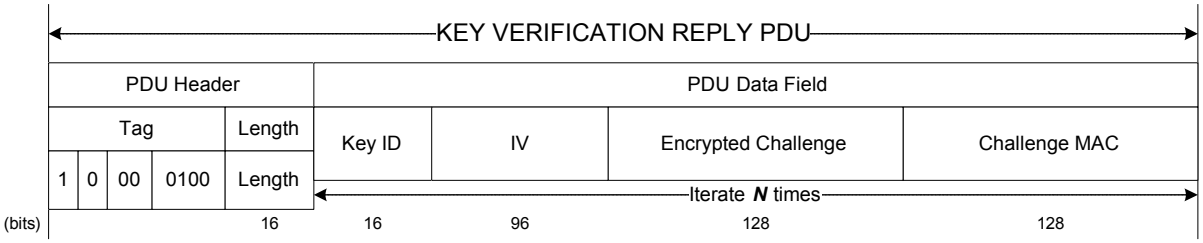

Figure D-5: Baseline Implementation Mode Key Verification Reply PDU

D5 SECURITY ASSOCIATIONS MANAGEMENT SERVICE

D5.1 OVERVIEW

This subsection specifies the baseline implementation mode for the Security Associations Management Service Extended Procedures. The configuration to be used for SA Management interoperability testing and operation is as follows.

D5.2 START SA

D5.2.1 The SPI field is 16 bits in length.

D5.2.2 The GVCID/GMAP ID field is a multiple of 32 bits in length. Each GVCID/GMAP ID entered consists of a concatenation of the following values from the underlying Space Link Protocol:

a) Transfer Frame Version Number (4 bits, right-justified);

b) Spacecraft ID (16 bits, right-justified);

c) Virtual Channel ID (6 bits, right-justified);

d) Multiplexer Access Point ID (6 bits).

D5.2.3 The total length of the PDU (Header + Data Field) does not exceed 995 octets. This is to ensure that the complete Command PDU fits into one frame of 1024 octets.

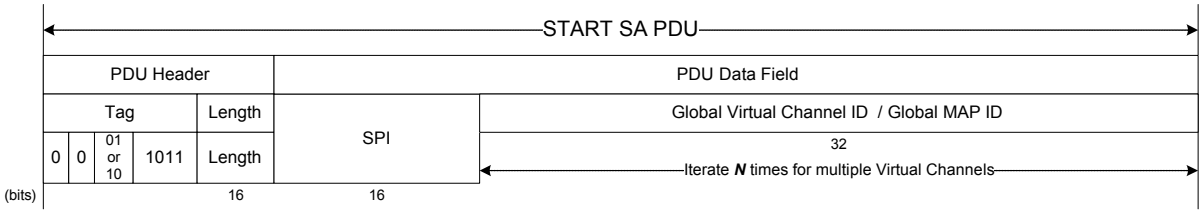

Figure D-6: Baseline Implementation Start SA Command PDU

D5.3 STOP SA

The SPI field is 16 bits in length.

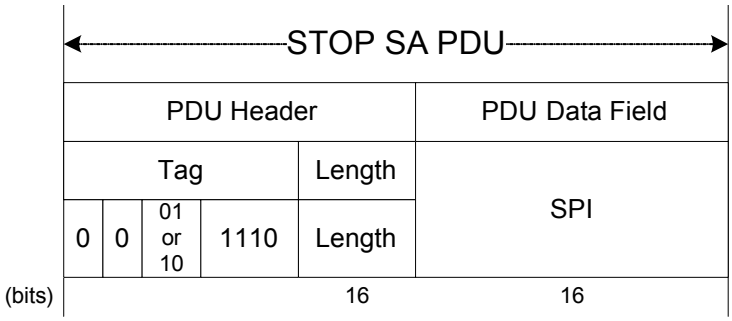

Figure D-7: Stop SA Command PDU

D5.4 REKEY SA

D5.4.1 The New Encryption Key ID field is 16 bits in length.

D5.4.2 The New Authentication Key ID field is 16 bits in length, right-justified.

D5.4.3 The ARSN is 32 bits in length for TC SAs. It is carried in the New ARSN value field.

D5.4.4 The IV is 96 bits in length for TM, AOS, and USLP SAs. It is carried in the New ARSN value field.

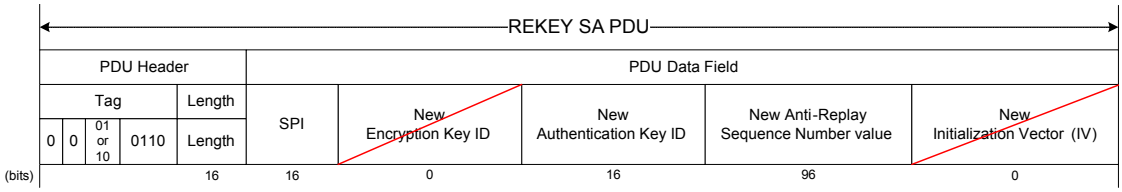

Figure D-8: Rekey SA Command PDU for TC

NOTES

1 The ARSN field length needs to be 96 bits to be able to carry either ARSN (TC SAs) or IV (TM, AOS, or USLP SAs). If this ARSN field carries an ARSN (TC SAs) and not an IV, the left-most 64 bits are zeroed.

2 Since the ARSN is identical to the IV for the SDLS baseline mode AES-GCM algorithm, executing this procedure will set the IV.

D5.5 EXPIRE SA

The SPI field is 16 bits in length.

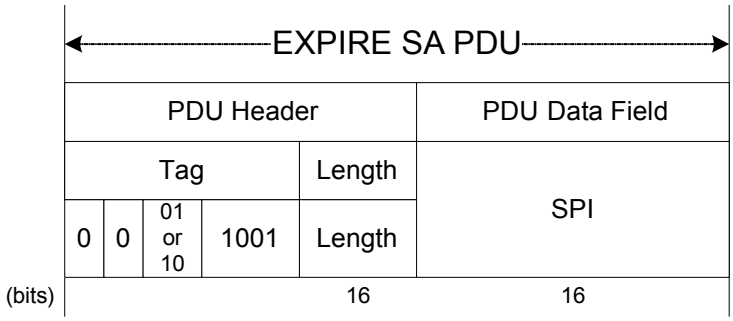

Figure D-9: Expire SA Command PDU

D5.6 SET ANTI-REPLAY SEQUENCE NUMBER

D5.6.1 The SPI field is 16 bits in length.

D5.6.2 The New ARSN Value field is 96 bits in length, right-justified.

NOTE – The ARSN length needs to be 96 bits. If this affects an ARSN and not an IV, the left-most 64 bits are zeroed.

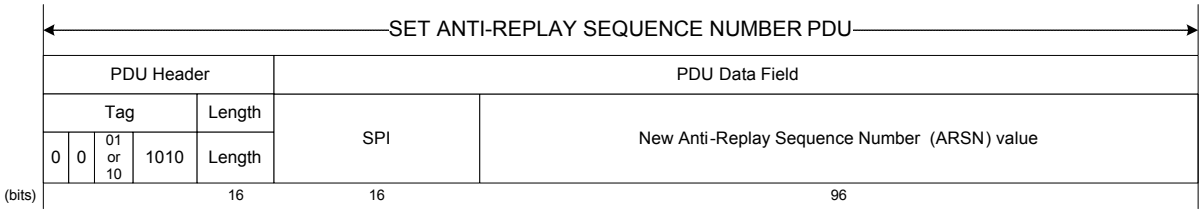

Figure D-10: Set ARSN Command PDU

NOTE – Since the ARSN is identical to the IV for the SDLS baseline mode AES-GCM algorithm, executing this procedure will set the IV.

D5.7 READ ANTI-REPLAY SEQUENCE NUMBER

The baseline implementation configuration to be used for Read ARSN procedure interoperability testing and operation is as follows:

a) For TC: The length of the ARSN is 4 octets (32 bits);

b) For TM/AOS: The length of the ARSN/Initialization Vector is 12 octets (96 bits).

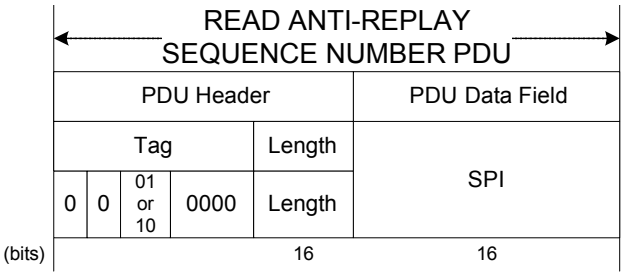

Figure D-11: Read ARSN Command PDU

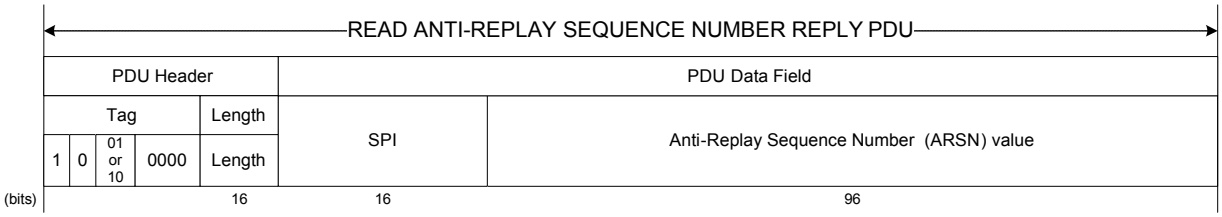

Figure D-12: Read ARSN Reply PDU

D6 SDLS MONITORING AND CONTROL SERVICE

D6.1 OVERVIEW

This subsection specifies the baseline implementation mode for the Monitoring and Control Service Extended Procedures.

D6.2 SDLS MONITORING AND CONTROL SERVICES PARAMETERS

D6.2.1 Ping

No specific configuration needed.

D6.2.2 Alarm Flag Reset

The baseline implementation configuration to be used for Alarm Flag Reset procedure interoperability testing and operation is

no specific configuration needed.

ANNEX E

ACRONYMS

(INFORMATIVE)

AOS	Advanced Orbiting Systems
ARSN	Anti-Replay Sequence Number
ARSNW	Anti-Replay Sequence Number Window
CLCW	command link control word
EP	(SDLS) Extended Procedures
FSR	Frame Security Report
GVCID	global virtual channel ID
GMAP ID	global MAP ID
IV	initialization vector
LSB	least significant bit
MAC	Message Authentication Code
MAP	multiplexer access point
MC_OCF	master channel operational control field
MOC	mission operations center
OCF	operational control field
OTAR	over the air rekeying
PDU	protocol data unit
PL	pad length
SA	Security Association
SN	Sequence Number
SDLS	Space Data Link Security
SH	Security Header
SLP	space link protocols
SPI	Security Parameter Index
TC	telecommand
TLV	tag length value
TM	telemetry
VC	virtual channel
VC_OCF	virtual channel operational control field