

Draft Recommendation for  
Space Data System Standards

|  |
| --- |
| SDLS Extended Procedures |

PROPOSED Draft Recommended Standard

CCSDS 355.1-R-1

Red Book

October 2016

AUTHORITY

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  | Issue: | Red Book, Issue 0 |  |
|  | Date: | October 2016 |  |
|  | Location: | Not Applicable |  |
|  | | | |

**(WHEN THIS RECOMMENDED STANDARD IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF AUTHORITY:)**

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 address below.

This document is published and maintained by:

CCSDS Secretariat

Space Communications and Navigation Office, 7L70

Space Operations Mission Directorate

NASA Headquarters

Washington, DC 20546-0001, USA

FOREWORD

[Foreword text specific to this document goes here. The text below is boilerplate.]

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. 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 addressed to the CCSDS Secretariat at the 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.
* Chinese 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.
* 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.
* Indian Space Research Organization (ISRO)/India.
* Institute of Space Research (IKI)/Russian Federation.
* KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
* Korea Aerospace Research Institute (KARI)/Korea.
* Ministry of Communications (MOC)/Israel.
* 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.
* 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.
* United States Geological Survey (USGS)/USA.

PREFACE

This document is a draft CCSDS Recommended Standard. Its ‘Red Book’ status indicates that its contents are not stable, and several iterations resulting in substantial technical changes are likely to occur before it is considered to be sufficiently mature to be released for review by the CCSDS Agencies.

Implementers are cautioned **not** to fabricate any final equipment in accordance with this document’s technical content.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

DOCUMENT CONTROL

|  |  |  |  |
| --- | --- | --- | --- |
| **Document** | **Title and Issue** | **Date** | **Status** |
| CCSDS 000.0-R-1 | SDLS Extended Procedures, Proposed Draft Recommended Standard, Issue 1 | October 2016 | Red 1 Draft 1 |
| CCSDS 000.0-R-1 | SDLS Extended Procedures, Proposed Draft Recommended Standard, Issue 1 | January 2017 | Red 1 Draft 2 |
| CCSDS 000.0-R-1 | SDLS Extended Procedures, Proposed Draft Recommended Standard, Issue 1 | April 2017 | Red 1 Draft 3 |
|  |  |  |  |

CONTENTS

Section Page

[1 Introduction 1-1](#_Toc479753871)

[1.1 Purpose 1-1](#_Toc479753872)

[1.2 Scope 1-1](#_Toc479753873)

[1.3 Applicability 1-1](#_Toc479753874)

[1.4 Rationale 1-2](#_Toc479753875)

[1.5 Document Structure 1-2](#_Toc479753876)

[1.6 Definitions 1-3](#_Toc479753877)

[1.7 Coventions 1-3](#_Toc479753878)

[1.8 References 1-4](#_Toc479753879)

[2 Overview 2-1](#_Toc479753880)

[2.1 Concept of SDLS Extended Procedures 2-1](#_Toc479753881)

[2.2 Features of SDLS Extended Procedures 2-1](#_Toc479753882)

[2.3 Services provided by SDLS Extended Procedures 2-1](#_Toc479753883)

[2.4 Frame Security Report (FSR) 2-4](#_Toc479753884)

[3 Service Definition 3-1](#_Toc479753885)

[3.1 OVERVIEW 3-1](#_Toc479753886)

[3.2 Key Management Service 3-1](#_Toc479753887)

[3.3 SECURITY ASSOCIATIONS MANAGEMENT SERVICE 3-12](#_Toc479753888)

[3.4 MONITORING & CONTROL SERVICE 3-28](#_Toc479753889)

[4 Interface with SLP & SDLS 4-41](#_Toc479753890)

[4.1 Overview 4-41](#_Toc479753891)

[4.2 Interface with SLP 4-41](#_Toc479753892)

[4.3 Interface with SDLS 4-45](#_Toc479753893)

[5 Procedures Specification 5-47](#_Toc479753894)

[5.1 Overview 5-47](#_Toc479753895)

[5.2 Procedure Identification 5-47](#_Toc479753896)

[5.3 Protocol Data Units 5-47](#_Toc479753897)

[5.4 Key Management 5-51](#_Toc479753898)

[5.5 Security Associations Management 5-57](#_Toc479753899)

[5.6 SDLS Monitoring and Control (M&C) 5-70](#_Toc479753900)

[6 MANAGED PARAMETERS 6-80](#_Toc479753901)

[6.1 Overview 6-80](#_Toc479753902)

[6.2 Requirements 6-80](#_Toc479753903)

[7 CONFORMANCE Requirements 7-82](#_Toc479753904)

[ANNEX A Implementation Conformance Statement (ICS) Proforma (normative) A-1](#_Toc479753905)

[ANNEX B Security, SANA, and Patent Considerations (Informative) B-1](#_Toc479753906)

[ANNEX C Informative References (Informative) C-3](#_Toc479753907)

[ANNEX D Baseline Implementation Mode (Informative) D-5](#_Toc479753908)

Figure

[Figure 2‑1: SDLS Extended Procedures Cryptographic Key Lifecycle 2-2](#_Toc479753909)

[Figure 2‑2: Variable State Model for Security Association Management 2-3](#_Toc479753910)

[Figure 4‑1: Frame Security Report (FSR) 4-42](#_Toc479753911)

[Figure 5‑1: TLV Format Specification 5-48](#_Toc479753912)

[Figure 5‑2: Extended Procedures PDU 5-48](#_Toc479753913)

[Figure 5‑3: OTAR Command PDU 5-52](#_Toc479753914)

[Figure 5‑4: Key Activation Command PDU 5-54](#_Toc479753915)

[Figure 5‑5: Key Deactivation Command PDU 5-54](#_Toc479753916)

[Figure 5‑6: Key Destruction Command PDU 5-55](#_Toc479753917)

[Figure 5‑7: Key Verification Command PDU 5-56](#_Toc479753918)

[Figure 5‑8: Key Verification Reply PDU 5-57](#_Toc479753919)

[Figure 5‑9: SA Management Procedures Overview 5-58](#_Toc479753920)

[Figure 5‑10: Start SA PDU 5-59](#_Toc479753921)

[Figure 5‑11: Stop SA PDU 5-59](#_Toc479753922)

[Figure 5‑12: Rekey SA PDU 5-60](#_Toc479753923)

[Figure 5‑13: Expire SA PDU 5-61](#_Toc479753924)

[Figure 5‑14: Create SA PDU 5-64](#_Toc479753925)

[Figure 5‑15: Delete SA PDU 5-67](#_Toc479753926)

[Figure 5‑16: Set ARC PDU 5-67](#_Toc479753927)

[Figure 5‑17: Set ARC Window PDU 5-68](#_Toc479753928)

[Figure 5‑18: SA Status Request PDU 5-69](#_Toc479753929)

[Figure 5‑19: SA Status Request Reply PDU 5-70](#_Toc479753930)

[Figure 5‑20 : Ping Command PDU 5-71](#_Toc479753931)

[Figure 5‑21 : Ping Reply PDU 5-71](#_Toc479753932)

[Figure 5‑22 : Log Status Command PDU 5-72](#_Toc479753933)

[Figure 5‑23 : Log Status Reply PDU 5-72](#_Toc479753934)

[Figure 5‑24 : Dump Log Command PDU 5-73](#_Toc479753935)

[Figure 5‑25 : Dump Log Reply PDU 5-74](#_Toc479753936)

[Figure 5‑26 : Erase Log Command PDU 5-74](#_Toc479753937)

[Figure 5‑27 : Erase Log Reply PDU 5-75](#_Toc479753938)

[Figure 5‑28 : Self-Test Command PDU 5-76](#_Toc479753939)

[Figure 5‑29 : Self-Test Reply PDU 5-77](#_Toc479753940)

[Figure 5‑30: Read Sequence Number Command PDU 5-77](#_Toc479753941)

[Figure 5‑31 : Read Sequence Number Reply PDU 5-78](#_Toc479753942)

[Figure 5‑32 : Alarm Flag Reset Command PDU 5-79](#_Toc479753943)

[Figure 7‑1:Baseline Implementation Mode OTAR Command PDU D-7](#_Toc479753944)

[Figure 7‑2: Baseline Implementation Mode Key Activation Command PDU D-8](#_Toc479753945)

[Figure 7‑3: Baseline Implementation Mode Key Deactivation Command PDU D-8](#_Toc479753946)

[Figure 7‑4: Baseline Implementation Mode Key Verification Command PDU D-9](#_Toc479753947)

[Figure 7‑5: Baseline Implementation Mode OTAR Reply PDU D-9](#_Toc479753948)

[Figure 7‑6: Bit Numbering Convention D-10](#_Toc479753949)

[**Figure D‑7: A Possible Implementation of the CRC-32 Encoder** D-11](#_Toc479753950)

[**Figure D‑8: A Possible Implementation of the CRC-32 Decoder** D-12](#_Toc479753951)

[Figure 7‑9: Baseline Mode Start SA PDU D-12](#_Toc479753952)

[Figure 7‑10: Stop SA PDU D-13](#_Toc479753953)

[Figure 7‑11: Rekey SA PDU D-13](#_Toc479753954)

[Figure 7‑12: Expire SA PDU D-13](#_Toc479753955)

[Figure 7‑13: Set ARC PDU D-14](#_Toc479753956)

# Introduction

## Purpose

The purpose of this Recommended Standard is to specify the Space Data Link Security (SDLS) Protocol Extended Procedures. It defines the Key Management, Security Association Management and 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 unit status reporting mechanism.

## Scope

This Recommended Standard defines the SDLS Extended Procedures in terms of:

a) the protocol data units employed by the service provider; and

b) the procedures performed by the service provider; 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 [7] are to be considered for this purpose.

## 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.

## Rationale

The goals of this Recommended Standard are to:

1. provide standard specifications for the Extended Procedures required to operate the Space Data-Link Layer Security (SDLS) Protocol [1], in particular:

* Key Management;
* Security Associations Management; and
* SDLS Monitoring and Control.

1. 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 and AOS) [4,5];
2. 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 [C4].

## 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.

## Definitions

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

NOTE – 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 is managing 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.

NOTE – In most cases the Initiator is the ground control facility and the Recipient is the spacecraft.

## CoNventions

### Nomentclature

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.

### 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.

## 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 for Space Missions, Issue 1, Recommendation for Space Data System Standards (Magenta Book), CCSDS 353.0-M-1. Washington, D.C.: CCSDS, May 2017.

[3] Information Security Glossary of Terms, Issue 1, Report concerning Space Data System Standards (Green Book), CCSDS 350.8-G-1. Washington, D.C.: CCSDS, November 2012

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

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

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

[7] CCSDS Cryptographic Algorithms. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 352.0-B-1. Washington, D.C.: CCSDS, November 2012.

[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 Link Protocol (USLP)

NOTE – Informative references are listed in annex C.

# Overview

## Concept of SDLS Extended Procedures

The Space Data-Link Layer Security (SDLS) protocol [1] is a data processing method for space missions that need to apply authenticity and/or confidentiality to the contents of TM, AOS, 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.

## 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

In addition, the SDLS Extended Procedures specify a Frame Security Report (FSR) that is reported as an Operational Control Field within TM or AOS frames and contains a brief report on the status of the on-board security unit.

## Services provided by SDLS Extended Procedures

The SDLS Extended Procedures provide three different services:

1. Key Management Service;
2. Security Association (SA) Management Service; and
3. SDLS Management & Control Service.

### Key Management Service

The Key Management Service implements the abstract description of a subset of the Key Management Procedures that are specified in the Symmetric Key Management Recommended Practice [2]. Thus terminology from this 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.

A number of different key management infrastructure designs exist, however for reasons of 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, i.e. 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 many of the key management procedures enable the transitions between states. The CCSDS Symmetric Key Management Recommended Practice [2] specifies in detail the various states and the transition rules.



Figure ‑: SDLS Extended Procedures Cryptographic Key Lifecycle

### Security Association (SA) Management Service

The SDLS protocol provides encryption, authentication, or authenticated encryption for data link layer services of the TC, TM, and AOS protocols. The Security Association (SA) 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 over-the-air rekeying, or even instantiation on demand of Security Associations.



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



Figure ‑: Variable State Model for Security Association Management

### SDLS Monitoring & Control Service

The SDLS Management and Control Service is designed to support the SDLS Monitoring and control of the on-board security processor. This is done via a set of messages sent to the security processor (Commands) or received from it (Reports). These messages allow for complete control / command of the on-board security processor(s). A security processor may be a complete equipment or a piece of software providing security functions as defined in the SDLS standard, applied to one communication link : TC, TM or AOS.

## Frame Security Report (FSR)

The SDLS Extended Procedures also specify a new type of Operational Control Field for the Space Data Link Layer protocols. This Frame Security report contains information about the status of the security unit and about the security processing (e.g. indicating a recent authentication failure). Since a TM or AOS frame cannot contain two OCFs at the same time, it is expected that the FSR sampling alternates with the OCF Type 1 (Command Link Control Word (CLCW)) sampling.

# Service Definition

## 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 and specify the information associated with a particular procedure step and passed in either direction between the Initiator and the Recipient. The way in which a specific implementation makes this information available is not constrained by this specification.

This Section defines the Key Management Service, the Security Association Management Service, and the SLDS Monitoring & Control Service.

## Key Management Service

### Overview

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

The following service procedures are specified:

* Over-the-air-rekeying (OTAR);
* Key Activation;
* Key Deactivation;
* Key Destruction; and
* Key Verification.

### Service Parameters

#### Over-the-air-rekeying (OTAR)

The OTAR procedure shall have the following Service Parameters:

1. Key ID of the Master Key used for the OTAR command protection;
2. Initialization Vector for the Protected Set of Upload Session Keys MAC;
3. Set of Upload Session Keys;
4. Protected Set of Upload Session Keys, containing for each session key to be uploaded:
   * Session Key ID
   * Encrypted Key
   * Key CRC

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

1. Protected Set of Upload Session Keys MAC

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

NOTE – The actual number of session keys is indicated by the length field of the Tag, Length, Value (TLV) format (see Section 5.3.1.1).

#### Key Activation

The Key Activation procedure shall have the following Service Parameters:

1. Set of Key IDs

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

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

#### Key Deactivation

The Key Deactivation procedure shall have the following Service Parameters:

1. Set of Key IDs

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

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

#### Key Destruction

The Key Destruction procedure shall have the following Service Parameters:

1. Set of Key IDs

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

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

#### Key Verification

The Key Verification procedure shall have the following Service Parameters:

1. Set of Key IDs
2. Key Status Set Response, containing for each key

* Key ID
* Key Status (0000 = OK, 1111 = NOK)
* Key State (0000=Pre-Active, 0001=Active, 0010=Deactivated, 0011=Destroyed, 0100=Corrupted)

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

NOTE – The actual number of keys to be verified is defined through the length field of the TLV format.

* + 1. **Service Procedures**
       1. **Over-the-air-rekeying (OTAR)**

Over-the-air-rekeying (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 practice.

* + - * 1. **Preconditions for the Procedure**

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

* + - * 1. **Procedural Steps**

**The OTAR procedure shall include the following mandatory execution steps:**

1. Protection of set of Upload Session Keys; Role: Initiator
2. Signaling of set of Protected Set of Upload Session Keys; Role: Initiator
3. Processing of Protected Set of Upload Session Keys; Role: Recipient

**Protection of set of Upload Session Keys**

This step shall be executed by the Initiator.

This step shall have the following inputs:

* Set of Upload Session Keys
* Key ID of the Master Key

This step shall have the following outputs:

* Protected Set of Upload Session Keys ready for upload
* Master Key in Active State

This step shall execute the following:

* 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.
* For each key in the set of upload session keys, the Initiator shall first generate a CRC over the (Key ID, Key) for each session key.
* Authenticated encryption under the selected master key shall be applied to the complete set of triplets (Key ID, Key, CRC) to create the Protected Set of Upload Session Keys. This shall be done using the agreed cryptographic algorithm under the master key identified by the master key ID. The initialization vector (if applicable) and MAC parameters shall be populated accordingly.

**Signaling of Protected Set of Upload Session Keys**

This step shall be executed by the Initiator.

This step shall have the following inputs:

* Protected Set of Upload Session Keys
* Key ID of the Master Key

This step shall have the following outputs:

* The Protected Set of Upload Session Keys and the Key ID of the Master Key transmitted to the Recipient

This step shall execute the following:

* An OTAR Command PDU as defined in Section 5.4.2.1 shall be created and transmitted to the Recipient using the SLP interface specified in Section 4.

**Processing of Protected Set of Upload Session Keys**

This step shall be executed by the Recipient.

This step shall have the following inputs:

* The Protected Set of Upload Session Keys and the Key ID of the Master Key received from the Initiator.

This step shall have the following outputs:

* None

This step shall execute the following:

* 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.
* For each decrypted Upload Session Key, the Recipient shall verify its integrity using the CRC provided.
* For each decrypted Upload Key, the Recipient shall store it in Pre-Active state using the indicated Key ID.

NOTE – This may or may not imply that other keys that are stored at memory slot associated with the indicated Upload Key ID are overridden. Proper management of the key memory is not the subject of this recommended standard and is mission specific.

* + - 1. **Key Activation**

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.

* + - * 1. **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.

* + - * 1. **Procedural Steps**

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

1. Activation of Initiator Keys; Role: Initiator
2. Signaling of Key IDs for Keys to be activated; Role: Initiator
3. Activation of Recipient Keys; Role: Recipient

**Activation of Initiator Session Keys**

This step shall be executed by the Initiator.

This step shall have the following inputs:

* Set of Key IDs

This step shall have the following outputs:

* All keys identified by the Set of Key IDs in State Activated.

NOTE – See Section 5.4.1.2 and reference [2] for more information on key states.

This step shall execute the following:

* The Keys identified by set of Key IDs shall be transitioned from Pre-Active State to Active State

**Signaling of Keys to be Activated**

This step shall be executed by the Initiator.

This step shall have the following inputs:

* Set of Key IDs of keys activated in Step 1.

This step shall have the following outputs:

* The Set of Key IDs of keys activated in Step 1 transmitted to the Recipient

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

This step shall execute the following:

* A Key Activation Command PDU as defined in Section 5.4.2.2 shall be created and transmitted to the Recipient using the SLP interface specified in Section 4.

**Activation of Recipient Session Keys**

This step shall be executed by the Recipient.

This step shall have the following inputs:

* The Set of Key IDs of keys activated in Step 1 received from the Initiator

This step shall have the following outputs:

* All session keys identified by the set of key IDs in State Active.

This step shall 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
  + - 1. **Key Deactivation**

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.

* + - * 1. **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.

* + - * 1. **Procedural Steps**

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

1. Deactivation of Initiator keys; Role: Initiator
2. Signaling of Key IDs of the keys to be deactivated; Role: Initiator
3. Deactivation of Recipient keys; Role: Recipient

**Deactivation of Initiator Keys**

This step shall be executed by the Initiator.

This step shall have the following inputs:

* The set of key IDs of keys to be deactivated

This step shall have the following outputs:

* All keys identified by the set of key IDs in State Deactivated.

NOTE – See Section 5.4.1.2 and reference [2] for more information on key states.

This step shall 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

**Signaling of Keys to be deactivated**

This step shall be executed by the Initiator.

This step shall have the following inputs:

* The set of Key IDs of keys deactivated in Step 1.

This step shall have the following outputs:

* The set of Key IDs of keys deactivated in Step 1 transmitted to the Recipient

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

This step shall execute the following:

* A Key Deactivation Command PDU as defined in Section 5.4.2.3 shall be created and transmitted to the Recipient using the SLP interface specified in Section 4.

**Deactivation of Recipient Keys**

This step shall be executed by the Recipient.

This step shall have the following inputs:

* The set of Key IDs of keys deactivated in Step 1 received from the Initiator.

This step shall have the following outputs:

* All keys identified by the set of key IDs in State Deactivated.

This step shall 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.
  + - 1. **Key Destruction**

The Key Destruction deletes a number of keys from both, Initiator and Recipient key databases.

* + - * 1. **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.

* + - * 1. **Procedural Steps**

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

1. Destruction of Initiator session keys; Role: Initiator
2. Signaling of session Key IDs to be destroyed; Role: Initiator
3. Destruction of Recipient session keys; Role: Recipient

**Destruction of Initiator Session Keys**

This step shall be executed by the Initiator.

This step shall have the following inputs:

* The set of key IDs of keys to be destroyed

This step shall have the following outputs:

* All keys identified by the set of key IDs in State Destroyed.

NOTE – See Section 5.4.1.2 and reference [2] for more information on key states.

This step shall 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

**Signaling of Keys to be destroyed**

This step shall be executed by the Initiator.

This step shall have the following inputs:

* The set of Key IDs of keys destroyed in Step 1.

This step shall have the following outputs:

* The set of Key IDs of keys destroyed in Step 1 transmitted to the Recipient

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

This step shall execute the following:

* A Key Destruction Command PDU as defined in Section 5.4.2.4 shall be created and transmitted to the Recipient using the SLP interface specified in Section 4.

**Destruction of Recipient Session Keys**

This step shall be executed by the Recipient.

This step shall have the following inputs:

* The set of Key IDs of keys destroyed in Step 1 received from the Initiator

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

This step shall have the following outputs:

* All keys identified by the set of key IDs in State Destroyed.

This step shall 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
  + - 1. **Key Verification**

The procedure allows the verification of a set of keys at the Recipient. This gives confirmation to the Initiator that the keys are not corrupted or modified and fully operational. The procedure will also report on the state of the keys.

* + - * 1. **Preconditions for the Procedure**

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

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

* + - * 1. **Procedural Steps**

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

1. Signaling of Key Verification Request; Role: Initiator
2. Execution of Key Verification; Role: Recipient
3. Signaling of Key Verification Responses; Role: Recipient

**Signaling of Key Verification Request**

This step shall be executed by the Initiator.

This step shall have the following inputs:

* The set of Key IDs to be verified.

This step shall have the following outputs:

* Key IDs to be verified transmitted to the Recipient

This step shall execute the following:

* A Key Verification Command PDU as defined in Section 5.4.2.5 shall be created and transmitted to the Recipient using the SLP interface specified in Section 4.

**Execution of Key Verification**

This step shall be executed by the Recipient.

This step shall have the following inputs:

* Key IDs to be verified received from the Initiator.

This step shall have the following outputs:

* The Key Verification Response Set

This step shall execute the following:

* For each key ID in the set of Key IDs verify the associated key using the stored CRC. Result is either OK or NOK
* For each key ID in the set of Key IDs identify the Key State.

**Signaling of Key Verification Responses**

This step shall be executed by the Recipient.

This step shall have the following inputs:

* The Key Verification Response Set created in Step b).

This step shall have the following outputs:

* Key Verification Response Set transmitted to the Initiator

This step shall execute the following:

* A Key Verification Response Set as defined in Section 5.4.2.5 shall be created and transmitted to the Initiator using the SLP interface specified in Section 4.

NOTE: This set includes that status of the key (OK or NOK) and the state.

## SECURITY ASSOCIATIONS MANAGEMENT SERVICE

The Security Association Management Service establishes the context of a Security Association for a particular Global Virtual Channel and/or Global MAP ID. 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;
* Create SA;
* Delete SA;
* Set Anti-Replay Counter;
* Set Anti-Replay Window; and
* SA Status Request.

### SERVICE PARAMETERS

#### Start SA

The Start SA directive shall have the following Service Parameters:

a) Security parameter index (SPI);

b) Global Virtual Channel ID(s) with which the SA is to be used;

#### Stop SA

The Stop SA directive shall have the following Service Parameters:

a) Security parameter index (SPI);

#### Rekey SA

The Rekey SA directive shall have the following Service Parameters:

1. Security parameter index (SPI);
2. Encryption Key ID; and
3. Authentication Key ID.

#### Expire SA

The Expire SA directive shall have the following Service Parameters:

a) Security parameter index (SPI).

#### Create SA

The Create SA directive shall have the following Service Parameters:

a) Security parameter index (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) Initialization vector (IV) length and initial value;

g) Authentication cipher suite length and identifier;

h) Authentication bit mask length and value;

i) Anti-replay counter (ARC) length and initial value; and

j) Anti-replay counter window length and value.

#### Delete SA

The Delete SA directive shall have the following Service Parameters:

a) Security parameter index (SPI).

#### Set Anti-Replay Counter

The Set Anti-Replay Counter directive shall have the following Service Parameters:

1. Security parameter index (SPI); and
2. Anti-replay counter value.

#### Set Anti-Replay Window

The Set Anti-Replay Window directive shall have the following Service Parameters:

1. Security parameter index (SPI); and
2. Anti-replay window value.

#### SA Status Request

The SA Status Request directive shall have the following Service Parameters:

1. Security parameter index (SPI); and
2. Procedure Identification of most recent SA state transition directive.

### SERVICE PROCEDURES

#### Start SA

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

##### Preconditions for the Procedure

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

##### Procedural Steps

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

1. Execution of Start SA; Role: Initiator
2. Signaling of Start SA Request; Role: Initiator
3. Execution of Start SA; Role: Recipient

###### Execution of Start SA

This step shall be executed by the Initiator.

This step shall have the following inputs:

* SPI of an existing Security Association which is in the ‘Keyed’ state.
* The specified GVC/GMAP ID(s) with which to use the SA.

This step shall have the following outputs:

* The SA transitions from ‘Keyed’ to ‘Operational’ state.

This step shall 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.

###### Signaling of Start SA Request

This step shall be executed by the Initiator.

This step shall have the following inputs:

* The SPI of the Security Association to activate
* The GVC/GMAP ID(s) upon which to activate the SA

This step shall have the following outputs:

* SPI and specified GVC/GMAP ID(s) transmitted to the Recipient

This step shall execute the following:

* A Start SA PDU as defined in Section 5.5.1.1 shall be created and transmitted to the Recipient using the SLP interface specified in Section 4.

###### Execution of Start SA

This step shall be executed by the Recipient.

This step shall have the following inputs:

* SPI and specified GVC/GMAP ID(s) received from the Initiator.

This step shall have the following outputs:

* The SA transitions from ‘Keyed’ to ‘Operational’ state.

This step shall 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.

#### Stop SA

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

##### Preconditions for the Procedure

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

##### Procedural Steps

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

1. Execution of Stop SA; Role: Initiator
2. Signaling of Stop SA Request; Role: Initiator
3. Execution of Stop SA; Role: Recipient

###### Execution of Stop SA

This step shall be executed by the Initiator.

This step shall have the following inputs:

* SPI of an existing Security Association which is in the ‘Operational’ state.

This step shall have the following outputs:

* The SA transitions from ‘Operational’ to ‘Keyed’ state.

This step shall 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.

###### Signaling of Stop SA Request

This step shall be executed by the Initiator.

This step shall have the following inputs:

* The SPI of the Security Association to stop

This step shall have the following outputs:

* SPI transmitted to the Recipient

This step shall execute the following:

* A Stop SA PDU as defined in Section 5.5.1.2 shall be created and transmitted to the Recipient using the SLP interface specified in Section 4.

###### Execution of Stop SA

This step shall be executed by the Recipient.

This step shall have the following inputs:

* SPI received from the Initiator.

This step shall have the following outputs:

* The SA transitions from ‘Operational’ to ‘Keyed’ state.

This step shall 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.

#### Rekey SA

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. This directive may be implicit.

##### Preconditions for the Procedure

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

##### Procedural Steps

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

1. Execution of Rekey SA; Role: Initiator
2. Signaling of Rekey SA Request; Role: Initiator
3. Execution of Rekey SA; Role: Recipient

###### Execution of Rekey SA

This step shall be executed by the Initiator.

This step shall have the following inputs:

* SPI of an existing Security Association which is in the ‘Unkeyed’ state.
* The specified key ID(s) to use.

This step shall have the following outputs:

* The SA transitions from ‘Unkeyed’ to ‘Keyed’ state.

This step shall 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.

###### Signaling of Rekey SA Request

This step shall be executed by the Initiator.

This step shall have the following inputs:

* The SPI of the Security Association to rekey
* The specified key ID(s) to use.

This step shall have the following outputs:

* SPI transmitted to the Recipient

This step shall execute the following:

* A Rekey SA PDU as defined in Section 5.5.1.2 shall be created and transmitted to the Recipient using the SLP interface specified in Section 4.

###### Execution of Rekey SA

This step shall be executed by the Recipient.

This step shall have the following inputs:

* SPI received from the Initiator.

This step shall have the following outputs:

* The SA transitions from ‘Unkeyed’ to ‘Keyed’ state.

This step shall 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.

#### Expire SA

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. This directive may be implicit.

##### Preconditions for the Procedure

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

##### Procedural Steps

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

1. Execution of Expire SA; Role: Initiator
2. Signaling of Expire SA Request; Role: Initiator
3. Execution of Expire SA; Role: Recipient

###### Execution of Expire SA

This step shall be executed by the Initiator.

This step shall have the following inputs:

* SPI of an existing Security Association which is in the ‘Keyed’ state.

This step shall have the following outputs:

* The SA transitions from ‘Keyed’ to ‘Unkeyed’ state.

This step shall execute the following:

* Remove all key(s) from the SA.

###### Signaling of Expire SA Request

This step shall be executed by the Initiator.

This step shall have the following inputs:

* The SPI of the Security Association to expire

This step shall have the following outputs:

* SPI transmitted to the Recipient

This step shall execute the following:

* A Expire SA PDU as defined in Section 5.5.1.2 shall be created and transmitted to the Recipient using the SLP interface specified in Section 4.

###### Execution of Expire SA

This step shall be executed by the Recipient.

This step shall have the following inputs:

* SPI received from the Initiator.

This step shall have the following outputs:

* The SA transitions from ‘Keyed’ to ‘Unkeyed’ state.

This step shall execute the following:

1. Verify that the specified SA exists and is in the ‘Keyed’ state.
2. Remove all key(s) from the SA.

#### Create SA

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

##### Preconditions for the Procedure

The Security Association must not already exist.

##### Procedural Steps

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

1. Execution of Create SA; Role: Initiator
2. Signaling of Create SA Request; Role: Initiator
3. Execution of Create SA; Role: Recipient

###### Execution of Create SA

This step shall be executed by the Initiator.

This step shall have the following inputs:

* SPI of a nonexistent Security Association.

This step shall have the following outputs:

* SA in the ‘Unkeyed’ state.

This step shall execute the following:

1. Verify that the specified SA does not exist.
2. Initialize a Security Association (SA) having the specified Security Parameter Index (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 Initialization vector (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 Anti-replay counter (ARC) length and initial value into the SA;
11. Add the Anti-replay counter window length and value into the SA.

###### Signaling of Create SA Request

This step shall be executed by the Initiator.

This step shall have the following inputs:

* The SPI of the Security Association to create
* The GVC/GMAP ID(s) upon which to activate the SA

This step shall have the following outputs:

* SPI and specified GVC/GMAP ID(s) transmitted to the Recipient

This step shall execute the following:

* A Create SA PDU as defined in Section 5.5.1.1 shall be created and transmitted to the Recipient using the SLP interface specified in Section 4.

###### Execution of Create SA

This step shall be executed by the Recipient.

This step shall have the following inputs:

* Fields in the Create SA PDU received from the Initiator.

This step shall have the following outputs:

* SA in the ‘Unkeyed’ state.

This step shall execute the following:

1. Verify that the specified SA does not exist.
2. Initialize a Security Association (SA) having the specified Security Parameter Index (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 Initialization vector (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 Anti-replay counter (ARC) length and initial value into the SA;
11. Add the Anti-replay counter window length and value into the SA.

#### Delete SA

The Delete SA directive is used to remove a Security Association entirely. This directive may be implicit.

##### Preconditions for the procedure

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

##### Procedural steps

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

1. Execution of Delete SA Request; Role: Initiator
2. Signaling of Delete SA Request; Role: Initiator
3. Execution of Delete SA Request; Role: Recipient

###### Execution of Delete SA Request (Initiator)

This step shall be executed by the Initiator.

This step shall have the following inputs:

* SPI of an existing Security Association which is in the ‘Unkeyed’ state.

This step shall have the following outputs:

* The SA transitions to a null state (‘No SA’).

This step shall execute the following:

* Erase all managed parameters of the SA.

###### Signaling of Delete SA Request

This step shall be executed by the Initiator.

This step shall have the following inputs:

* The SPI of the Security Association to delete

This step shall have the following outputs:

* SPI transmitted to the Recipient

This step shall execute the following:

* A Delete SA PDU as defined in Section 5.5.1.1 shall be created and transmitted to the Recipient using the SLP interface specified in Section 4.

###### Execution of Delete SA Request (Recipient)

This step shall be executed by the Recipient.

This step shall have the following inputs:

* SPI received from the Initiator.

This step shall have the following outputs:

* The SA transitions to a null state (‘No SA’).

This step shall execute the following:

1. Verify that the specified SA exists and is in the ‘Unkeyed’ state.
2. Erase all managed parameters of the SA.

#### Set Anti-Replay Counter

The Set Anti-Replay Counter directive is used to initialize the managed anti-replay sequence number for a Security Association to the value supplied by the service user.

##### Preconditions for the Procedure

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

##### Procedural Steps

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

1. Execution of Set ARC; Role: Initiator
2. Signaling of Set ARC Request; Role: Initiator
3. Execution of Set ARC; Role: Recipient

###### Execution of Set ARC

This step shall be executed by the Initiator.

This step shall have the following inputs:

* SPI of an existing Security Association.
* Requested new value for the managed anti-replay sequence number.

This step shall have the following outputs:

* None.

This step shall execute the following:

* Replace the current value of the managed anti-replay sequence number with the requested value.

###### Signaling of Set ARC Request

This step shall be executed by the Initiator.

This step shall have the following inputs:

* The SPI of the Security Association
* Requested new value for the managed anti-replay sequence number.

This step shall have the following outputs:

* SPI and new ARC value transmitted to the Recipient

This step shall execute the following:

* A Set ARC Command PDU as defined in Section 5.5.1.2 shall be created and transmitted to the Recipient using the SLP interface specified in Section 4.

###### Execution of Set ARC

This step shall be executed by the Recipient.

This step shall have the following inputs:

* SPI received from the Initiator.
* Requested new value for the managed anti-replay sequence number received from the Initiator

This step shall have the following outputs:

* None.

This step shall 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.

#### Set Anti-Replay Window

The Set Anti-Replay 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.

##### Preconditions for the Procedure

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

##### Procedural Steps

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

1. Execution of Set ARCW; Role: Initiator
2. Signaling of Set ARCW Request; Role: Initiator
3. Execution of Set ARCW; Role: Recipient

###### Execution of Set ARCW

This step shall be executed by the Initiator.

This step shall have the following inputs:

* SPI of an existing Security Association.
* Requested new value for the managed anti-replay sequence number window.

This step shall have the following outputs:

* None.

This step shall execute the following:

* Replace the current value of the managed anti-replay sequence number window with the requested value.

###### Signaling of Set ARCW Request

This step shall be executed by the Initiator.

This step shall have the following inputs:

* The SPI of the Security Association
* Requested new value for the managed anti-replay sequence number window.

This step shall have the following outputs:

* SPI and new ARCW value transmitted to the Recipient

This step shall execute the following:

* A Set ARCW Command PDU as defined in Section 5.5.1.2 shall be created and transmitted to the Recipient using the SLP interface specified in Section 4.

###### Execution of Set ARCW

This step shall be executed by the Recipient.

This step shall have the following inputs:

* SPI received from the Initiator.
* Requested new value for the managed anti-replay sequence number window received from the Initiator.

This step shall have the following outputs:

* None.

This step shall 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.

#### SA Status Request

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

##### Preconditions for the Procedure

None.

* + - * 1. **Procedural Steps**

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

1. Signaling of SA Status Request; Role: Initiator
2. Execution of SA Status Verification; Role: Recipient
3. Signaling of SA Status Response; Role: Recipient

**Signaling of SA Status** **Request**

This step shall be executed by the Initiator.

This step shall have the following inputs:

* SPI of an existing Security Association

This step shall have the following outputs:

* SA Status Request Command PDU

This step shall execute the following:

* A SA Status Request Command PDU as defined in Section 5.4.2.5 shall be created and transmitted to the Recipient using the SLP interface specified in Section 4.

**Execution of SA Status** **Verification**

This step shall be executed by the Recipient.

This step shall have the following inputs:

* SA Status Request Command PDU received from the Initiator.

This step shall have the following outputs:

* State of an existing Security Association.

This step shall 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.

**Signaling of SA Status** **Response**

This step shall be executed by the Recipient.

This step shall have the following inputs:

* State of an existing Security Association created in Step b).

This step shall have the following outputs:

State of an existing Security Association transmitted to the Initiator.This step shall execute the following:

* A SA Status Request Reply PDU as defined in Section 5.4.2.5 shall be created and transmitted to the Initiator using the SLP interface specified in Section 4.

## MONITORING & CONTROL SERVICE

### OVERVIEW

The following service procedures are specified:

* Ping;
* Log Status;
* Dump Log;
* Erase Log;
* Self-Test;
* Read Sequence Number; and
* Alarm Flag Reset.

### SERVICE PARAMETERS

#### Ping

The Ping procedure shall have no Service Parameter.

#### Log Status

The Log Status procedure shall have the following Service Parameters:

1. Number of Security Messages stored in the Security Log (Integer).
2. Available space in the Security Log (Integer).

#### Dump Log

The Dump Log procedure shall have the following Service Parameter :

1. 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. However, each security message has to comply with the TLV format.

#### Erase Log

The Erase Log shall have the following service parameters :

1. Number of messages stored in the Security Log after erasing (Integer)
2. Space available in the Security Log (Integer)

#### Self-Test

The Self-Test procedure shall have the following Service Parameter :

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

#### Read Sequence Number

The Read Sequence Number (SN) procedure shall have the following Service Parameters :

1. Security Parameter Index (16 bit)

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

1. Sequence Number Value (bit field length managed by the SA)

NOTE – The length of the SN is a managed parameter within the SA. It can vary. However, the full value of the Sequence Number must be reported without truncation.

#### Alarm Flag Reset

The Alarm Flag Reset procedure shall have the following Service Parameter :

1. Alarm Flag (1 bit)

### SERVICE PROCEDURES

#### Ping

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

##### Preconditions for the procedure

None.

##### Procedural steps

###### The Ping procedure shall include the following mandatory execution steps:

1. Signaling of the Ping Request; Role: Initiator
2. Signaling of the Ping Response; Role: Recipient

###### Signaling of the Ping Request

This step shall be executed by the Initiator.

This step shall have the following input:

None

This step shall have the following output:

* The Ping Request transmitted to the Recipient.

This step shall execute the following:

* A Ping Request Command PDU as defined in Section 5.6.1.1.2 shall be created and transmitted to the Recipient using the SLP interface specified in Section 4.

###### Signaling of the Ping Response

This step shall be executed by the Recipient.

This step shall have the following input:

• Ping Request received from the Initiator

This step shall have the following output:

• Ping Response transmitted to the Initiator.

This step shall execute the following:

A Ping Reply PDU as defined in Section 5.6.1.1.3 shall be created and transmitted to the Initiator using the SLP interface specified in Section 4.

#### Log Status

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

##### Preconditions for the procedure

None.

##### Procedural steps

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

1. Signaling of the Log Status Request; Role: Initiator
2. Generation of the Log Status Response; Role: Recipient
3. Signaling of the Log Status Response; Role : Recipient.

###### Signaling of the Log Status Request

This step shall be executed by the Initiator.

This step shall have the following input:

• None

This step shall have the following output:

• The Log Status Request transmitted to the Recipient.

This step shall execute the following:

A Log Status Request Command PDU as defined in Section 5.6.1.2 shall be created and transmitted to the Recipient using the SLP interface specified in Section 4.

###### Generation of the Log Status Response

This step shall be executed by the Recipient.

This step shall have the following input:

• Reception of Log Status Request from the Initiator.

This step shall have the following output:

• Log Status Response.

This step shall execute the following:

* The Recipient shall assess the status of the Security Log and derive

o Number of entries in the log

o Remaining capacity in the log

* Generate Log Status Response.

###### Signaling of the Log Status Response

This step shall be executed by the Recipient

This step shall have the following input :

• Log Status Response created at step b)

This step shall execute the following :

* A Log Status Reply PDU as defined in Section 5.6.1.2 shall be created and transmitted to the Initiator using the SLP interface specified in Section 4.

#### Dump Log

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.

##### Preconditions for the procedure

None.

##### Procedural steps

###### 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

###### Signaling of the Dump Log Request

This step shall be executed by the Initiator.

This step shall have the following input:

* None

This step shall have the following output:

* The Dump Log Request transmitted to the Recipient.

This step shall execute the following:

* A Dump Log Request Command PDU as defined in Section 5.6.1.3 shall be created and transmitted to the Recipient using the SLP interface specified in Section 4.

###### Computation of the Dump Log Response

This step shall be executed by the Recipient.

This step shall have the following input :

* Reception of Dump Log Request from the Initiator.

This step shall have the following output :

* Dump Log Response

This step shall execute the following:

* The Recipient shall derive the List of Log Security Messages from the Security Log.
* Generate the Dump Log Response comprising all the Security Messages.

###### Signaling of the Dump Log Response

This step shall be executed by the Recipient.

This step shall have the following input :

* The Dump Log Response created at Step b)

This step shall have the following output :

* Dump Log Response transmitted to the Initiator

This step shall execute the following :

* A Dump Log Reply PDU as defined in Section 5.6.1.3 shall be created and transmitted to the Initiator using the SLP interface specified in Section 4.

#### Erase Log

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

##### Preconditions for the procedure

None.

##### Procedural steps

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

1. Signaling of Erase Log Request; Role : Initiator
2. Erasing of the entire set of messages stored in the Security Log and generation of the Erase Log Response; Role : Recipient
3. Signaling of the Erase Log Response; Role : Recipient

###### Signaling of the Erase Log Request

This step shall be executed by the Initiator.

This step shall have the following input:

* None

This step shall have the following output:

* The Erase Log Request transmitted to the Recipient.

This step shall execute the following:

* An Erase Log Command PDU as defined in Section 5.6.1.4 shall be created and transmitted to the Recipient using the SLP interface specified in Section 4.

###### Erasing of the entire set of messages stored in the Security Log and computation of the Erase Log Response

This step shall be executed by the Recipient.

This step shall have the following input:

* Reception of Erase Log Request from the Initiator.

This step shall have the following outputs:

* Number of entries in the Security Log
* Remaining space in the Security Log

This step shall execute the following:

The Recipient shall :

* Erase all Security Messages from the Security Log
* Assess the status of the Security Log and derive:
  + Number of entries in the Security Log
  + Remaining space in the Security Log
* Generate the Erase Log Response comprising the number of entries and the remaining space in the Security Log

###### Signaling of the Erase Log Response

This step shall be executed by the Recipient.

This step shall have the following input :

* Erase Log Response created at Step b)

This step shall have the following output :

* Erase Log Response transmitted to the Initiator

This step shall execute the following :

* An Erase Log Reply PDU as defined in Section 5.6.1.4 shall be created and transmitted to the Initiator using the SLP interface specified in Section 4.

#### Self-Test

The Self-Test directive is used to trigger a security processor self-test.

##### Preconditions for the procedure

None.

##### Procedurals steps

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

1. Signaling of Self-Test Request; Role : Initiator
2. Computation of the Self-Test Response; Role : Recipient
3. Signaling of Self-Test Response; Role : Recipient

###### Signaling of Self-Test Request

This step shall be executed by the Initiator.

This step shall have the following input:

* None

This step shall have the following output:

* The Self-Test Request transmitted to the Recipient

This step shall execute the following:

* A Self-Test Command PDU as defined in Section 5.6.1.5 shall be created and transmitted to the Recipient using the SLP interface specified in Section 4.

###### Computation of the Self-Test Response

This step shall be executed by the Recipient.

This step shall have the following input:

* Reception of Self-Test Request from the Initiator

This step shall have the following output:

* Self-Test Response

This step shall 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.

###### Signaling of Self-Test Response

This step shall be executed by the Recipient.

This step shall have the following input :

* The Self-Test Response created in Step b)

This step shall have the following output :

* Self-Test Response transmitted to the Initiator.

This step shall execute the following :

* A self-Test Reply PDU as defined in Section 5.6.1.5 shall be created and transmitted to the Initiator using the SLP interface specified in Section 4.

#### Read Sequence Number (SN)

The Read SN directive is used to read the current Sequence Number value associated to a given SA.

##### Preconditions for the procedure

None.

##### Procedural steps

###### The Read Sequence Number procedure shall include the following mandatory execution steps:

1. Signaling of Read Sequence Number Request; Role : Initiator
2. Computation of the Read Sequence Number Response; Role : Recipient
3. Signaling of the Read Sequence Number Response; Role : Recipient

###### Signaling of Read Sequence Number Request

This step shall be executed by the Initiator.

This step shall have the following input:

* Security Parameter Index (SPI) of the SA

This step shall have the following output:

* The Read Sequence Number Request transmitted to the Recipient

This step shall execute the following:

* A Read Sequence Number Command PDU as defined in Section 5.6.1.6 shall be created and transmitted to the Recipient using the SLP interface specified in Section 4.

###### Computation of the Read Sequence Number Response

This step shall be executed by the Recipient.

This step shall have the following input:

* Reception of Read Sequence Number Request from the Initiator, including the SPI of the SA

This step shall have the following output:

* Read Sequence Number Response

This step shall execute the following:

* The recipient shall read the Sequence Number Value corresponding to the SA identified by the SPI and create the Read Sequence Number Response.

###### Signaling of the Read Sequence Number Response

This step shall be executed by the recipient

This step shall have the following input :

* The Read Sequence Number response created at Step b)

This step shall have the following output :

* Read Sequence Number Response transmitted to the Initiator

This step shall execute the following :

* A Read Sequence Number Reply PDU as defined in Section 5.6.1.6 shall be created and transmitted to the Initiator using the SLP interface specified in Section 4.

#### Alarm Flag Reset

##### Preconditions for the procedure

None.

##### Procedurals steps

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

1. Signaling of Alarm Flag Reset Request; Role : Initiator
2. Resetting the Alarm Flag of the Frame Security Report; Role : Recipient

###### Signaling of Alarm Flag Reset Request

This step shall be executed by the Initiator.

This step shall have the following input:

* None

This step shall have the following output:

* The Alarm Flag Reset Request transmitted to the Recipient.

This step shall execute the following:

* An Alarm Flag Reset Command PDU as defined in Section 5.6.1.7 shall be created and transmitted to the Recipient using the SLP interface specified in Section 4.

###### Resetting the Alarm Flag of the Frame Security Report

This step shall be executed by the Recipient.

This step shall have the following input:

* Reception of Alarm Flag Reset Request from the Initiator.

This step shall have the following outputs:

* Alarm Flag of the Frame Security Report (FSR) reset.

This step shall 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.

# Interface with SLP & SDLS

## Overview

The SDLS Extended Procedures are interfacing with the Space Link Protocols (SLP) for transport of the procedures protocol data units. This recommended practice does not mandate or recommend the transport data structures except for a new Operational Control Field (OCF), the Frame Security Report (FSR). The SDLS Extended Procedures Concept of Operations 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 however are generally on the Initiator and Recipient side, and not directly on protocol level. However, a Security Association should be allocated to the management channel.

## Interface with SLP

### Transfer of EP Service PDU Over the Space Link

#### For transport of SDLS Extended Procedures PDUs on the uplink (TC or USLP data link protocols, the MAP packet service with a dedicated MAP shall be used (see references [6], [8], [9]).

#### For transport of SDLS Extended Procedures PDUs on the downlink (TM, AOS or USLP data link protocols), the VC packet service shall be used (see references [4], [5], [8], [9]).

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

### Frame Security Report (FSR)

#### General

##### The Frame Security Report (FSR), which is the protocol data unit transmitted from the Recipient to the Initiator of an SDLS secured TC uplink, shall provide the systematic, real-time mechanism by which the SDLS function at the receiving end reports the status of TC 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 recommendation. They provide non real-time or non-systematic reporting of the frame acceptance status at the receiving end of the SDLS secured TC uplink.

##### The FSR shall be carried in the Operational Control Field of TM or AOS Transfer Frames (references [4], [5]) using the MC\_OCF or the VC\_OCF Service.

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

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

1. Control Word Type (1 bit, mandatory)
2. FSR Version Number (3 bits, mandatory)
3. Alarm field (1 bit, mandatory)
4. Security event flags (3 bits, mandatory)
5. Last SPI field (16 bits, mandatory)
6. SN Value field (8 bits, mandatory)

NOTE – The structural components of the FSR are shown in Figure 4‑1.



Figure ‑: Frame Security Report (FSR)

#### Control Word Type

##### Bit 0 of the FSR shall contain the Control Word Type

NOTE – This field is used to distinguish Command Link Control Word (CLCW) – control word type “0” specified in [6] from other type of Control Word (type “1”) like the FSR, that may be alternatively carried in the Operational Control Field (OCF) of TM and AOS transfer frames.

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

#### FSR Version Number

##### Bits 1-3 of the FSR shall contain the FSR Version Number

##### 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’) identifies a ‘Version-1’ FSR, whose binary encoded Version Number is ‘00’. At present, a single version is defined in this recommendation. The FSR Version Number is included to provide future growth flexibility.

#### Alarm Field

##### Bit 4 of the FSR shall contain the Alarm Flag

##### The Alarm Flag shall indicate whether a TC Transfer Frame has been rejected by the on-board SDLS function.

##### A setting of ‘0’ in the Alarm Flag shall indicate that all TC Transfer Frames have been accepted by the on-board SDLS function since the last reset of the Alarm Flag.

##### A setting of ‘1’ in the Alarm Flag shall indicate that at least one TC Transfer Frame has been rejected by the on-board SDLS function since the last reset of the Alarm Flag.

##### The Alarm Flag shall apply to all Virtual Channels and Security Associations of the TC uplink.

##### The Alarm Flag shall be updated at each TC Transfer Frame processed by the SDLS on-board function.

##### Once the Alarm Flag is set to ‘1‘, it shall remain persistent until reset to ‘0’ (NoAlarm state) by a dedicated command (See Section5.6.1.7).

#### Security Event Flags

##### General

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

##### Bad Sequence Number Flag

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

###### The Bad Sequence Number Flag shall indicate whether the Sequence Number (SN) of the last received TC Transfer Frame by the SDLS on-board function is valid.

###### A setting of ‘0’ in the Bad Sequence Number Flag shall indicate that the SN carried by the last received TC Transfer Frame by the on-board SDLS function is valid (i.e. within the SN window).

###### A setting of ‘1’ in the Bad Sequence Number Flag shall indicate that the SN carried by the last received TC Transfer Frame by the on-board SDLS function is invalid (i.e. outside the SN window).

###### The Bad Sequence Number Flag shall be updated at each TC Transfer Frame processed by the SDLS on-board function, its states not being persistent.

##### Bad MAC Flag

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

###### The Bad MAC Flag shall indicate whether the last received TC Transfer Frame by the SDLS on-board function failed MAC verification.

###### A setting of ‘0’ in the Bad MAC Flag shall indicate that the MAC carried by the last received TC Transfer Frame by the on-board SDLS function is valid (i.e. matches the MAC computed over the received Transfer Frame).

###### A setting of ‘1’ in the Bad MAC Flag shall indicate that the MAC carried by the last received TC Transfer Frame by the on-board SDLS function is invalid (i.e. does not match the MAC computed over the received Transfer Frame).

###### The Bad MAC Flag shall be updated at each TC Transfer Frame processed by the SDLS on-board function, its states not being persistent.

##### Invalid SPI Flag

###### Bit 7 of the FSR shall contain the Invalid SPI Flag.

###### The Invalid SPI Flag shall indicate whether the last received TC Transfer Frame by the SDLS on-board function failed SA verification.

###### A setting of ‘0’ in the Bad SPI Flag shall indicate that the SPI carried by the last received TC Transfer Frame by the on-board SDLS function is valid (i.e. SPI points to an SA that is associated with the GVCID and/or GMAP of the received Transfer Frame)

###### A setting of ‘1’ in the Bad SPI Flag shall indicate that the SPI carried by the last received TC Transfer Frame by the on-board SDLS function is invalid (i.e. SPI points to an SA that is not associated with the GVCID and/or GMAP of the received Transfer Frame)

###### The Bad SPI Flag shall be updated at each TC Transfer Frame processed by the SDLS on-board function, its states not being persistent.

#### Last SPI used

##### Bits 8-23 of the FSR shall contain the SPI carried in the last received TC Transfer Frame by the on-board SDLS function.

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

##### Sequence Number Value (LSB)

##### Bits 24-31 of the FSR shall contain the 8 Least Significant Bits (LSB) of the Sequence Number (SN) carried in the last received TC Transfer Frame by the on-board SDLS function.

## Interface with SDLS

### Transfer of EP Service PDU

#### Two Security Associations associated with master keys shall be reserved for use with the SDLS Extended Procedures and shall be active upon start of a mission phase.

#### A Security Association cannot be used to execute any SA Management Procedure that would affect itself. Another security association must be used in this case.

NOTE – SDLS has two reserved SPIs (values of ‘all zeros’ (0) and ‘all ones’ (65535)). Those could be used by the SDLS Extended Procedures to manage Security Associations.

#### All EP Service Command PDUs shall only be transmitted over an authenticated SDLS channel.

#### Sensitive EP Service PDUs shall only be communicated over a SDLS channel protected by authenticated encryption. Table 4‑1 shows the sensitivity attributes for each procedure.

| **Procedure** | **Status** |
| --- | --- |
| Over-the-air-rekeying (OTAR)  (OTAR PDUs provide their own protection) | Not Sensitive |
| Key Activation | Sensitive |
| Key Deactivation | Sensitive |
| Key Verification | Sensitive |
| Key Destruction | Sensitive |
| Delete SA | Not Sensitive |
| Expire SA | Not Sensitive |
| Stop SA | Not Sensitive |
| Create SA | Sensitive |
| Rekey SA | Not Sensitive |
| Start SA | Not Sensitive |
| SA Status Request | Not Sensitive |
| Set Anti-Replay Counter | Not Sensitive |
| Set Anti-Replay Counter Window | Sensitive |
| Ping | Not Sensitive |
| Log Status Request | Not Sensitive |
| Dump Log | Sensitive |
| Erase Log | Sensitive |
| Self-Test | Not Sensitive |
| Read Sequence Number | Not Sensitive |
| Reset Alarm Flag | Not Sensitive |

Table ‑: Extended Procedures Sensitivity

# Procedures Specification

## Overview

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

## Procedure Identification

#### The Extended Procedures PDU header shall identity the type of procedure to which the contents of the Extended Procedures PDU data field is associated.

NOTE – For a mapping of field values to procedures, see Table 5‑1.

## Protocol Data Units

### Overview

Extended Procedures Protocol Data Units (PDUs) are the data structures that carry the information related to Extended Procedures commands and reports.

#### Tag, Length, Value Notation

SLDS Extended Procedures commands and reports share a common message format, based on the “Tag, Length, Value” (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 to define 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 TLV concept allows nesting: the Value field can itself be composed of one or more TLV messages.



Figure ‑: TLV Format Specification

### Extended Procedures PDU

#### General

##### The Extended Procedures PDU shall be used for transport of SDLS Extended Procedures.

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

1. Extended Procedures PDU Header (24 bits; mandatory)
2. Extended Procedures PDU Data Field (variable but octet-aligned; mandatory)

NOTE – 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.

NOTE – The format of the Extended Procedures PDU is shown in Figure 5‑2.



Figure ‑: Extended Procedures PDU

#### Extended Procedures PDU Header

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

1. Extended Procedures Tag (8 bits; mandatory)
2. Extended Procedures Data Field Length (16 bits; mandatory)

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

1. Procedure Type (1 bit; mandatory)
2. User Flag (1 bit; mandatory)
3. Service Group Field (2 bits; mandatory)
4. Procedure Identification Field (4 bits; mandatory)

###### The Procedure Type Flag

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.

A setting of “0” shall identify a command.

A setting of “1” shall identify a reply.

###### User Flag

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

A setting of “0” shall identify a CCSDS defined procedure.

A setting of “1” shall identify a user defined procedure.

###### Service Group Field

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

A setting of “00” shall identify a Key Management procedure.

A setting of “01” shall identify a Security Association Management procedure.

A setting of “10” shall identify a Security Monitoring & Control procedure.

A setting of “11” may identify a user defined service group.

###### Procedure Identification Field

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

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

| **Procedure Identification** | **Assignment** |
| --- | --- |
| 0001 | Over-the-air-rekeying (OTAR) |
| 0010 | Key Activation |
| 0011 | Key Deactivation |
| 0100 | Key Verification |
| 0110 | Key Destruction |
| 0001 | Create SA |
| 0110 | Rekey SA |
| 1011 | Start SA |
| 1110 | Stop SA |
| 1001 | Expire SA |
| 0100 | Delete SA |
| 1010 | Set Anti-Replay Counter |
| 0101 | Set Anti-Replay Counter Window |
| 1111 | SA Status Request |
| 0001 | Ping |
| 0010 | Log Status Request |
| 0011 | Dump Log |
| 0100 | Erase Log |
| 0101 | Self-Test |
| 0110 | Read Sequence Number |
| 0111 | Reset Alarm Flag |

Table ‑: Extended Procedures PDU Header Values

#### Extended Procedures PDU Data Field Length

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

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

#### Extended Procedures PDU Data Field

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

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

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

## Key Management

### Key Types and Key Lifecycle

#### Key Types

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

#### Key Lifecycle

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

### Key Management Procedures

#### Over-the-air-rekeying (OTAR)

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

1. OTAR command PDU

##### OTAR command PDU

###### The OTAR command PDU shall be associated with Step 2 of the OTAR Procedure as defined in Section 3.2.3.1.

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

1. Key ID of the master key used for encryption of session keys;
2. Initialization Vector for the authenticated encryption of the Upload Key Block (optional);
3. Upload Key Block consisting of N (Encrypted Key ID Field, Encrypted Upload Key Field, Encrypted CRC) field triplets (managed length; mandatory)
4. MAC field for the authenticated encryption of the Upload Key Block

NOTE – The number of (Encrypted Key ID Field, Encrypted Upload Key Field, Encrypted CRC) field triplets fields in the PDU data field and the size of the Master Key ID, initialization vector, encrypted key ID fields, encrypted upload key, and encrypted CRC, and MAC are managed parameters.

NOTE – The format of the OTAR command PDU is shown in Figure 5‑3.



Figure ‑: OTAR Command PDU

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

###### 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.

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

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

###### The Encrypted CRC field shall signal, in encrypted format, the CRC for keys to be uploaded to the Recipient.

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

#### Key Activation

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

1. Key Activation Command PDU.

##### Key Activation Command PDU

###### The Key Activation Command PDU shall be associated with Step 2 of the Key Activation Procedure as defined in Section 3.2.3.2.

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

1. Key ID field (managed length; mandatory)

NOTE – The number of Key ID fields in the PDU data field and the size of the key ID fields are managed parameters.

NOTE – The format of the Key Activation Command PDU is shown in Figure 5‑4.



Figure ‑: Key Activation Command PDU

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

#### Key Deactivation

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

1. Key Deactivation Command PDU

##### Key Deactivation Command PDU

###### The Key Deactivation Command PDU shall be associated with Step 2 of the Key Deactivation Procedure as defined in Section 3.2.3.3.

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

1. Key ID field (managed length; mandatory)

NOTE – The number of Key ID fields in the PDU data field and the size of the key ID fields are managed parameters.

NOTE – The format of the Key Deactivation Command PDU is shown in Figure 5‑5.



Figure ‑: Key Deactivation Command PDU

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

#### Key Destruction

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

1. Key Destruction Command PDU

##### Key Destruction Command PDU

###### The Key Destruction Command PDU shall be associated with Step 2 of the Key Destruction Procedure as defined in Section 3.2.3.4.

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

1. Key ID field (managed length; mandatory)

NOTE – The number of Key ID fields in the PDU data field and the size of the key ID fields are managed parameters.

NOTE – The format of the Key DestructionCommand PDU is shown in Figure 5‑6.



Figure ‑: Key Destruction Command PDU

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

#### Key Verification

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

1. Key Verification Command PDU
2. Key Verification Reply PDU

##### Key Verification Command PDU

###### The Key Verification Command PDU shall be associated with Step 2 of the Key Verification Procedure as defined in Section 3.2.3.5.

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

1. Key ID field (managed length; mandatory)

NOTE – The number of Key ID fields in the PDU data field and the size of the key ID fields are managed parameters.

NOTE – The format of the Key Verification Command PDU is shown in Figure 5‑7.



Figure ‑: Key Verification Command PDU

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

##### Key Verification Reply PDU

###### The Key Verification Reply PDU shall be associated with Step 4 of the Key Verification Procedure as defined in Section 3.2.3.5.

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

1. (Key ID Field, Verification Status, Key State) triplets

###### The Key Verification Status field shall be indicating status and state as following: 4 MSB: Key Status (0000=OK, 1111= NOK) 4 LSB: Key State (0000=Pre-Active, 0010=Active, 0100 =Deactivated, 0100=Destroyed, 1000=Corrupted)

NOTE – The number of (Key ID Field, Verification Status, Key State) triplets in the PDU data field and the size of the key ID fields and response fields are managed parameters.

NOTE – The format of the Key Verification Reply PDU is shown in Figure 5‑8.



Figure ‑: Key Verification Reply PDU

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

## Security Associations Management

### SA Management Procedures

NOTE – Security Association Management directives and state transitions are shown in 

Figure 5‑9.



Figure ‑: SA Management Procedures Overview

#### Start SA

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

1. StartSA PDU

##### Start SA PDU

###### The Start SA PDU shall be associated with the Start SA Procedure as defined in Section 3.3.2.1.

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

1. SPI of the applicable Security Association (16 bits, mandatory);
2. One or more Global Virtual Channels (GVC) / Global Multiplexer Access Points (GMAP) with which to use the SA (32 bits each; mandatory)

NOTE – 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 fields. The precise method of encoding these into a unique 32-bit field is mission-specific.

NOTE – The format of the Start SA PDU is shown in Figure 5‑3.



Figure ‑: Start SA PDU

###### The SPI field shall signal the applicable Security Association.

###### 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.

#### Stop SA

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

1. Stop SA PDU

##### Stop SA PDU

###### The Stop SA PDU shall be associated with the Stop SA Procedure as defined in Section 3.3.2.2.

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

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

NOTE – The format of the StopSA PDU is shown in Figure 5‑11.



Figure ‑: Stop SA PDU

###### The SPI field shall signal the applicable Security Association.

#### Rekey SA

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

1. RekeySA PDU

##### Rekey SA PDU

###### The Rekey SA PDU shall be associated with the Rekey SA Procedure as defined in Section 3.3.2.3.

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

1. SPI of the applicable Security Association (16 bits, mandatory)
2. New encryption key ID for the SA (managed length, optional)
3. New authentication key ID for the SA (managed length, optional)

NOTE – The format of the RekeySA PDU is shown in 

Figure 5‑12.



Figure ‑: Rekey SA PDU

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

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

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

###### 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.

#### Expire SA

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

1. ExpireSA PDU

##### Expire SA PDU

###### The Expire SA PDU shall be associated with the Expire SA Procedure as defined in Section 3.3.2.4.

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

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

NOTE – The format of the ExpireSA PDU is shown in 

Figure 5‑13.



Figure ‑: Expire SA PDU

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

#### Create SA

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

1. Create SA PDU

##### Create SA PDU

###### The Create SA PDU shall be associated with the Create SA Procedure as defined in Section 3.3.2.5.

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

1. SPI of the applicable Security Association (16 bits, mandatory)
2. SA Service Type flag for Encryption (1 bit, mandatory)
3. SA Service Type flag for Authentication (1 bit, mandatory)
4. Security Header IV Field Length (6 bits, mandatory)
5. Security Header SN Field Length (6 bits, mandatory)
6. Security Header PL Field Length (2 bits, mandatory)
7. Security Trailer MAC Field Length (8 bits, mandatory)
8. Encryption cipher suite length (8 bits, mandatory)
9. Encryption cipher suite identifier (managed length, optional)
10. Initialization vector length (8 bits, mandatory)
11. Initialization Vector (IV) value (managed length, optional)
12. Authentication cipher suite length (8 bits, mandatory)
13. Authentication cipher suite identifier (managed length, optional)
14. Authentication bit mask length (8 bits, mandatory)
15. Authentication bit mask value (managed length, optional)
16. Anti-replay counter (ARC) length (8 bits, mandatory)
17. Anti-replay counter (ARC) value (managed length, optional)
18. Anti-replay counter window length (8 bits, mandatory)
19. Anti-replay counter window value (managed length, optional)

NOTE – The format of the CreateSA PDU is shown in 

Figure 5‑14.



Figure ‑: Create SA PDU

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

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

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

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

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

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

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

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

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

###### 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.

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

###### The Initialization Vector (IV) value shall signal the initial managed value of the Initialization Vector for the SA.

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

###### 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.

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

###### The Authentication bit mask shall signal the authentication bit mask value for the SA.

###### The Anti-replay counter (ARC) length shall signal the length of the Anti-replay counter field in the PDU.

###### The Anti-replay counter (ARC) value shall signal the initial value of the managed anti-replay counter for the SA.

###### The Anti-replay counter window length shall signal the length of the Anti-replay counter window field in the PDU.

###### The Anti-replay counter window value shall signal the initial value of the managed anti-replay counter window for the SA.

#### Delete SA

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

1. Delete SA PDU

##### Delete SA PDU

###### The Delete SA PDU shall be associated with the Delete SA Procedure as defined in Section 3.3.2.6.

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

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

NOTE – The format of the DeleteSA PDU is shown in 

Figure 5‑15.



Figure ‑: Delete SA PDU

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

#### Set Anti-Replay Counter (ARC)

##### The Set Anti-Replay Counter (ARC) Procedure shall support one Extended Procedures PDU data field structure:

1. Set Anti-Replay Counter (ARC) PDU

##### Set Anti-Replay Counter (ARC) PDU

###### The Set ARC PDU shall be associated with the Set ARC Procedure as defined in Section 3.3.2.7.

###### The Set ARC PDU shall consist of a managed number of contiguously positioned mandatory fields:

1. SPI of the applicable Security Association (16 bits, mandatory);
2. New value of the Anti-Replay Counter (managed length; mandatory)

NOTE – The format of the Set ARC PDU is shown in Figure 5‑16.



Figure ‑: Set ARC PDU

###### The SPI field shall signal the SPI of the Security Association whose ARC is to be modified.

###### The ARC field shall signal the new anti-replay counter value.

#### Set Anti-Replay Counter (ARC) Window

##### The Set Anti-Replay Counter Window Procedure shall support one Extended Procedures PDU data field structure:

1. Set ARC Window PDU

##### Set ARC Window PDU

###### The Set ARC Window PDU shall be associated with the Set ARC Window Procedure as defined in Section 3.3.2.8.

###### The Set ARC Window PDU shall consist of a managed number of contiguously positioned mandatory fields:

1. SPI of the applicable Security Association (16 bits, mandatory);
2. New value of the anti-replay counter window (managed length; mandatory)

NOTE – The format of the Set ARC Window PDU is shown in Figure 5‑17.



Figure ‑: Set ARC Window PDU

###### The SPI field shall signal the SPI of the Security Association whose ARC window is to be modified.

###### The ARC Window field shall signal the new anti-replay counter window value.

##### SA Status Request PDU

###### The SA Status Request PDU shall be associated with the SA Status Request Procedure as defined in Section 3.3.2.9.

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

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

NOTE – The format of the SA Status RequestPDU is shown in 

Figure 5‑18.



Figure ‑: SA Status Request PDU

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

##### SA Status Request Reply PDU

###### The SA Status Request Reply PDU shall be associated with the SA Status Request Procedure as defined in Section 3.3.2.9.

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

1. SPI of the applicable Security Association (16 bits, mandatory);
2. Procedure Identification of the last executed state transition directive for the applicable Security Association (8 bits, mandatory)

NOTE – 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.

NOTE – The format of the SA Status RequestReplyPDU is shown in Figure 5‑19.



Figure 5‑19: SA Status Request Reply PDU

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

###### 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.

## SDLS Monitoring and Control (M&C)

### M&C Procedures

#### Ping

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

* 1. Ping Command PDU
  2. Ping Reply PDU

##### Ping Command PDU

###### The Ping Command PDU shall be associated with Step a) of the Ping Procedure as defined in Section 3.4.3.1.

###### The Ping Command PDU shall have no data field

NOTE – The format of the Ping Command PDU is shown in Figure 5‑20



Figure ‑ : Ping Command PDU

##### Ping Reply PDU

###### The Ping Reply PDU shall be associated with step b) of the Ping Procedure as defined in section 3.4.3.1.

###### The Ping Reply PDU shall have no data field

NOTE – The format of the Ping Reply PDU is shown in Figure 5‑21



Figure ‑ : Ping Reply PDU

#### Log Status

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

1. Log Status Command PDU
2. Log Status Reply PDU

##### Log Status Command PDU

###### The Log Status Command PDU shall be associated with Step a) of the Log Status Procedure as defined in Section 3.4.3.2.

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

NOTE – The format of the Log Status Command PDU is shown in Figure 5‑22.



Figure ‑ : Log Status Command PDU

##### Log Status Reply PDU

###### The Log Status Reply PDU shall be associated with Step c) of the Log Status procedure as defined in Section 3.4.3.2.

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

1. Number of Security Events in the Security Log (managed length; mandatory)
2. Remaining space in the Security Log (managed length; mandatory)

NOTE – The format of the Log Status Reply PDU is shown in Figure 5‑23.



Figure ‑ : Log Status Reply PDU

#### Dump Log

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

1. Dump Log Command PDU
2. Dump Log Reply PDU

##### Dump Log Command PDU

###### The Dump Log Command PDU shall be associated with Step a) of the Dump Log Procedure as defined in Section 3.4.3.3.

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

NOTE – The format of the Dump Log Command PDU is shown in Figure 5‑24



Figure ‑ : Dump Log Command PDU

##### Dump Log Reply PDU

###### The Dump Log Reply PDU shall be associated with Step c) of the Dump Log Procedure as defined in Section 3.4.3.3.

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

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

NOTE – The format of the Dump Log Reply PDU is shown in Figure 5‑25



Figure ‑ : Dump Log Reply PDU

#### Erase Log

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

1. Erase Log Command PDU
2. Erase Log Reply PDU.

##### Erase Log Command PDU

###### The Erase Log Command PDU shall be associated with Step a) of the Erase Log Procedure as defined in Section 3.4.2.4.

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

NOTE – The format of the Erase Log Command PDU is shown in Figure 5‑26.



Figure ‑ : Erase Log Command PDU

##### Erase Log Reply PDU

###### The Erase Log Reply PDU shall be associated with Step c) of the Erase Log Procedure as defined in Section 3.4.2.4.

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

1. Number of Security Events in the Security Log after erasure of the Security Log (managed length; mandatory)
2. 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 5‑27.



Figure ‑ : Erase Log Reply PDU

#### Self-Test

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

1. Self-Test Command PDU
2. Self-Test Reply PDU

##### Self-Test Command PDU

###### The Self-Test Command PDU shall be associated with Step a) of the Self-Test Procedure as defined in Section 3.4.2.5.

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

###### NOTE – The format of the Self-Test Command PDU is shown in Figure 5‑28.



Figure ‑ : Self-Test Command PDU

##### Self-Test Reply PDU

###### The Self-Test Reply PDU shall be associated with Step c) of the Self-Test Procedure as defined in Section 3.4.2.5.

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

1. Self-Test result :

* 0XXXXXXXb: Self-Test OK (8 bit length)
* 1XXXXXXXb: Self-Test not OK (8 bit length)
* The bits having the value ‘X’ in the above definitions are not used by CCSDS. Their values are left to the implementer’s choice.

NOTE – The format of the Self-Test Reply PDU is shown in Figure 5‑29.



Figure ‑ : Self-Test Reply PDU

#### Read Sequence Number

##### The Read Sequence Number Procedure shall support two Extended Procedures PDU data field structures :

1. Read Sequence Number Command PDU
2. Read Sequence Number Reply PDU

##### Read Sequence Number Command PDU

###### The Read Sequence Number Command PDU shall be associated with Step a) of the Read Sequence Number Procedure as defined in Section 3.4.2.6.

###### The Read Sequence Number Command PDU shall consist of one mandatory data field :

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

NOTE 1 – The format of the Read Sequence Number Command PDU is shown in Figure 5‑30.



Figure ‑: Read Sequence Number Command PDU

##### Read Sequence Number Reply PDU

###### The Read Sequence Number Reply PDU shall be associated with Step c) of the Read Sequence Number Procedure as defined in Section 3.4.2.6.

###### The Read Sequence Number Reply PDU shall consist of one mandatory data field:

1. Sequence Number Value (managed length, mandatory)

NOTE – The format of the Read Sequence Number Reply PDU is shown in Figure 5‑31.



Figure ‑ : Read Sequence Number Reply PDU

#### Alarm Flag Reset

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

1. Alarm Flag Reset Command PDU

##### Alarm Flag Reset Command PDU

###### The Alarm Flag Reset Command PDU shall be associated with Step a) of the Alarm Flag Reset Procedure as defined in Section 3.4.3.7

###### 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 5‑32.



Figure ‑ : Alarm Flag Reset Command PDU

# MANAGED PARAMETERS

## 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.

## Requirements

### 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.

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

Table 6‑ : Managed Parameters for SDLS Extended Procedures

| **Managed Parameter** | **Allowed Values** | **Defined In Reference** |
| --- | --- | --- |
| **Key Management managed parameters:** | | |
| Key State |  |  |
| OTAR Number of Keys to be uploaded |  |  |
| OTAR Initialization vector (IV) length |  | [1] |
| OTAR Upload Session Key Size |  |  |
| Key ID Field Size |  |  |
| OTAR CRC Size |  |  |
| OTAR MAC Size |  |  |
| Key Verification Status | See 5.4.2.5.3.3. |  |
| **SA Management managed parameters:** | | |
| Security parameter index (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 |  |  |
| Initialization vector (IV) length |  | [1] |
| Initialization vector (IV) value |  | [1] |
| Authentication cipher suite identifier |  |  |
| Authentication bit mask length |  | [1] |
| Authentication bit mask value |  | [1] |
| Anti-replay counter (ARC) length |  | [1] |
| Anti-replay counter (ARC) value |  | [1] |
| Anti-replay counter window length |  | [1] |
| Anti-replay counter window value | Integer greater than zero (> 0) | [1] |
| Encryption Key ID |  |  |
| Authentication Key ID |  |  |
| Global Virtual Channel ID(s)  Global MAP ID(s) |  |  |
| Procedure Identification of most recent SA state transition directive |  |  |
| **Monitoring & Control managed parameters:** | | |
| Number of Security Events (in Security Log) field length |  |  |
| Remaining Space (in Security Log) field length |  |  |
| Sequence Number Value field length |  |  |

# CONFORMANCE Requirements

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

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

1. Implementation Conformance   
   Statement (ICS) Proforma  
     
   (normative)
   1. Introduction
      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 (PICS). 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.

* + 1. 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 which conforms to this Recommended Standard shall be a conforming PICS proforma completed in accordance with the instructions for completion given in A2.

* + 1. 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.

* 1. Instructions for completing the PICS proforma
     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, where the columns are headed ‘Status’, and ‘Support’. The definition of each of these follows.

* + 1. 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.

* + 1. 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:

**Y** Yes, the feature has been implemented.

**N** No, the feature has not been implemented.

**–** The item is not applicable.

* + 1. Item reference numbers

Each line within the PICS proforma which 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:

1. a reference to the smallest subsection enclosing the relevant item;
2. a solidus character, ‘/’;
3. the reference number of the row in which the response appears;
4. 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 **Error! Reference source not found.**/1, which refers to the SDLS implementation’s support for the TM Space Data Link Protocol.

* + 1. 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 where 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.

* 1. GENERAL INFORMATION
     1. Referenced Base Standards

The Space Data Link Security (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.

* + 1. 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 [C11]). It contains a declaration of the layers of the Reference Model covered by the implementation to be tested.

* + 1. Identification of the system supplier and/or  
       test laboratory client

|  |  |
| --- | --- |
| Organization name |  |
| Contact name |  |
| Address |  |
| Telephone |  |
| E-mail |  |
| Other information |  |

* + 1. IDENTIFICATION OF the IMPLEMENTATION UNDER TEST

|  |  |
| --- | --- |
| Implementation name |  |
| Implementation version |  |
| Machine name |  |
| Machine version |  |
| Operating system name |  |
| Operating system version |  |
| Special configuration |  |
| Other information |  |

* + 1. IDENTIFICATION OF the protocol

|  |  |
| --- | --- |
| Protocol specification / version |  |
| Technical corrigenda implemented |  |
| Other amendments implemented (explain) |  |

* + 1. 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) |  |

* 1. 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 | | | |

* 1. Service Primitives
     1. Key Management Service Primitives

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Protocol Feature | Reference | Status | Support |
| 1 | Over-the-Air-Rekeying (OTAR) |  | M |  |
| 2 | Key Activation |  | M |  |
| 3 | Key Deactivation |  | M |  |
| 4 | Key Destruction |  | O |  |
| 5 | Key Verification |  | M |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |
| 8 |  |  |  |  |
| 9 |  |  |  |  |
|  | C.2: if [ A4/3 ] is supported then M, else n/a  C.3: if [ A4/5 ] is supported then M, else n/a | | | |

* + 1. SA Management Service Primitives

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Protocol Feature | Reference | Status | Support |
| 1 | Start SA |  | M |  |
| 2 | Stop SA |  | M |  |
| 3 | Rekey SA |  | C.2 |  |
| 4 | Expire SA |  | C.2 |  |
| 5 | Create SA |  | C.3 |  |
| 6 | Delete SA |  | C.3 |  |
| 7 | Set ARC |  | O |  |
| 8 | Set ARCW |  | O |  |
| 9 | SA Status Request |  | O |  |
|  | C.2: if [ A4/3 | A4/4 ] is supported then M, else n/a  C.3: if [ A4/5 | A4/6 ] is supported then M, else n/a | | | |

* + 1. 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 | Read Sequence Number |  | M |  |
| 7 | Alarm Flag Reset |  | M |  |
| 8 |  |  |  |  |
| 9 |  |  |  |  |
|  | C.1: if Security Log implemented then Items 2, 3, 4 mandatory | | | |

* 1. Protocol Data Units
     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 |  |
|  |  | | | |

* + - 1. 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 |  |
| 5 | Encrypted CRC |  | M |  |

* + - 1. Key Activiation Command PDU Data Field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Protocol Feature | Reference | Status | Support |
| 1 | Key ID |  | M |  |

* + - 1. Key Deactivation Command PDU Data Field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Protocol Feature | Reference | Status | Support |
| 1 | Key ID |  | M |  |

* + - 1. Key Destruction Command PDU Data Field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Protocol Feature | Reference | Status | Support |
| 1 | Key ID |  | M |  |

* + - 1. Key Verfication Command PDU Data Field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Protocol Feature | Reference | Status | Support |
| 1 | Key ID |  | M |  |
| 2 | Verification Status |  | M |  |
| 3 | Key State |  | M |  |

* + - 1. Start SA PDU Data Field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Protocol Feature | Reference | Status | Support |
| 1 | SPI |  | M |  |
| 2 | GVCID / GMAP ID |  | M |  |

* + - 1. Stop SA PDU Data Field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Protocol Feature | Reference | Status | Support |
| 1 | SPI |  | M |  |

* + - 1. Rekey SA PDU Data Field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Protocol Feature | Reference | Status | Support |
| 1 | SPI |  | M |  |
| 2 | Encryption Key ID |  | O.1 |  |
| 3 | Authentication Key ID |  | O.1 |  |
|  | O.1: Support for at least one of [ A4/2 | A4/3 ] is M | | | |

* + - 1. Expire SA PDU Data Field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Protocol Feature | Reference | Status | Support |
| 1 | SPI |  | M |  |

* + - 1. 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 | ARC Length |  | M |  |
| 17 | ARC |  | M |  |
| 18 | ARCW Length |  | M |  |
| 19 | ARCW |  | M |  |

* + - 1. Delete SA PDU Data Field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Protocol Feature | Reference | Status | Support |
| 1 | SPI |  | M |  |

* + - 1. SET ARC PDU Data Field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Protocol Feature | Reference | Status | Support |
| 1 | SPI |  | M |  |
| 2 | ARC |  | M |  |

* + - 1. Set ARCW PDU Data Field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Protocol Feature | Reference | Status | Support |
| 1 | SPI |  | M |  |
| 2 | ARCW |  | M |  |

* + - 1. SA Status Request PDU Data Field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Protocol Feature | Reference | Status | Support |
| 1 | SPI |  | M |  |

* + - 1. SA Status Request Reply PDU Data Field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Protocol Feature | Reference | Status | Support |
| 1 | SPI |  | M |  |
| 2 | Last State Transition |  | M |  |

* + - 1. Ping Command PDU Data Field

None.

* + - 1. Ping Reply PDU Data Field

None.

* + - 1. Log Status Command PDU Data Field

None.

* + - 1. 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 |  |

* + - 1. Erase Log Command PDU Data Field

None.

* + - 1. Log Status Reply PDU Data Field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Protocol Feature | Reference | Status | Support |
| 1 | Number of Security Events |  | M |  |
| 2 | Remaining Space |  | M |  |

* + - 1. SELF-TEST Command PDU Data Field

None.

* + - 1. Self-Test Reply PDU Data Field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Protocol Feature | Reference | Status | Support |
| 1 | Self-Test Result |  | M |  |

* + - 1. Read Sequence Number Command PDU Data Field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Protocol Feature | Reference | Status | Support |
| 1 | SPI |  | M |  |

* + - 1. Read Sequence Number Reply PDU Data Field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Protocol Feature | Reference | Status | Support |
| 1 | Sequence Number Value |  | M |  |

* + - 1. Alarm-Flag Reset Command PDU Data Field

None.

1. Security, SANA, and Patent Considerations  
     
   (Informative)
   1. Security Considerations
      1. security concerns with respect to the CCSDS document

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 practice supports the management of communication security as established by the SDLS protocol.

* + - 1. Data Privacy

This recommended practice identifies which of the SDLS Extended Procedures are considered sensitive in terms of Privacy and require specific protection.

* + - 1. Data Integrity

This recommended practice identifies which of the SDLS Extended Procedures are considered sensitive in terms of Integrity and require specific protection.

* + - 1. Authentication of Communicating Entities

This recommended practice relies on the SDLS protocol to properly authenticate the communicating entities.

* + 1. Potential threats and attack scenarios

The same considerations as for the SLDS protocol [1] apply here. Specific potential threats and attack scenarios are addressed in more detail in reference [D2].

* + 1. 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 worst case the mission. Without confidentiality, session keys and the data field contents of sensitive extended procedures may disclosed to an attacker.

* 1. SANA Considerations

This Recommended Standard defines no new information registries. The recommendations of this document do not require any action from SANA.

* 1. 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.

1. Informative References  
     
   (Informative)

[] *Information Technology—Open Systems Interconnection—Basic Reference Model: The Basic Model*. 2nd ed. International Standard, ISO/IEC 7498-1:1994. Geneva: ISO, 1994.

[] *Information Processing Systems—Open Systems Interconnection—Basic Reference Model—Part 2: Security Architecture*. International Standard, ISO 7498-2:1989. Geneva: ISO, 1989.

[] *Space Data Link Security Concept of Operation*. Report Concerning Space Data System Standards, CCSDS 350.5-G. Washington, D.C.: CCSDS, forthcoming.

[] *The Application of CCSDS Protocols to Secure Systems*. Issue 2. Report Concerning Space Data System Standards (Green Book), CCSDS 350.0-G-2. Washington, D.C.: CCSDS, January 2006.

[] *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.

[] *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.

[] *Overview of Space Communications Protocols*. Issue 2. Report Concerning Space Data System Standards (Green Book), CCSDS 130.0-G-2. Washington, D.C.: CCSDS, December 2007.

[] *National Information Assurance (IA) Glossary*. CNSS Instruction No. 4009. Fort Meade, Maryland: CNSS, April 2010.

[] *Glossary of Key Information Security Terms*. Rev. 1. Edited by Richard Kissel. NIST IR 7298. Gaithersburg, Maryland: NIST, February 2011.

[C10] Elaine Barker, et al. *Recommendation for Key Management—Part 1: General*. National Institute of Standards and Technology Special Publication 800-57. Gaithersburg, Maryland: NIST, March 2007.

[] *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.

[] *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.

1. Baseline Implementation Mode  
     
   (Informative)
   1. Frame Security Report

This section specifies the baseline implementation mode for the Frame Security Report (FSR) and its integration into the transfer service protocol.

* + 1. Transfer Service Interface
       1. FSR USE WITH TM

The baseline implementation mode to be used for integrating the FSR into the TM transfer service shall be as follows:

1. The FSR shall be reported as Operational Control Field (OCF Type 2), see also Section 4.2.2.3.2.
2. In case COP-1 is reporting on the same virtual channel, the FSR reporting shall alternate with the Command Link Control Word (OCF Type 1) reporting.
   * + 1. FSR USE WITH AOS

The baseline implementation mode to be used for integrating the FSR into the AOS transfer service shall be as follows:

1. The FSR shall be reported as Operational Control Field (OCF Type 2), see also Section 4.2.2.3.2.
2. In case COP-1 is reporting on the same virtual channel, the FSR reporting shall alternate with the Command Link Control Word (OCF Type 1) reporting.
   * + 1. INTERFACE WITH SDLS

The SLDS implementation used for the SDLS Extended Procedures shall be the SDLS baseline implementation mode (see reference [1] Annex E).

* 1. Protocol Data Units

The baseline implementation of the TLV format shall not make use of nesting.

* 1. Reserved SPIs/SAs

The baseline implementation mode uses two reserved SPIs for the master key channels as specified in Section 4.3.

These two reserved SPIs are ‘all zeros’ (0) and ‘all ones’ (65535) (see also reference [1]).

* 1. Key Management Service

This section specifies the baseline implementation mode for the Key Management Service Extended Procedures.

The baseline implementation mode shall not include the Key Destruction Procedure.

* + 1. Security Algorithm and Key Configuration

The following security algorithm configuration shall be used to support the Key Management Service Extended Procedures in the baseline implementation mode.

* + - 1. Algorithm for OTAR

The baseline implementation to be used for OTAR operation is:

* For authentication of the key block: AES-GCM as defined in reference [7];
* The keys shall be 128 bits in total length;
* The IV shall be 96 bits in total length;
* The output MAC for the authentication shall be 128 bit in length.
  + 1. Key Management Services Parameters
       1. OTAR

The baseline implementation configuration to be used for OTAR procedure (Sections 3.2.3.1 and 5.4.2.1) interoperability testing and operation is:

* The Master Key ID field of the OTAR Command PDU shall have 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.
* The Initialization Vector field of the OTAR Command PDU shall have a size of 96 bit.
* Each Encrypted Key Block of the OTAR Command PDU shall have a size of 176 bit, consisting of
  + The Key ID fields shall have a size of 16 bit,
  + The Session Key fields shall have a size of 128 bit,
  + The CRC fields shall be calculated as specified in D4.3
* The MAC field of the OTAR Command PDU shall have a size of 128 bit.
* The total length of the PDU (Header + Data Field) shall not exceed 1004 octets (=upload of 32 session keys). This is to ensure that the complete Command PDU fits into one frame of 1024 octets.
* The Length field shall indicate a number of N\*176+240, where N is the number of session keys to be uploaded, N<=32.



Figure 7‑1:Baseline Implementation Mode OTAR Command PDU

* + - 1. Key Activation

The baseline implementation configuration to be used for Key Activation procedure (Sections 3.2.3.2 and 5.4.2.2) interoperability testing and operation is:

* The Key ID fields of the Key Activation PDU data field structure shall have a size of 16 bit.
* The total length of the PDU (Header + Data Field) shall not exceed 81 octets (=activation of 32 session keys).
* The Length field shall indicate a number of N\*16, where N is the number of session keys to be activated, N<=32.



Figure ‑: Baseline Implementation Mode Key Activation Command PDU

* + - 1. Key Deactivation

The baseline implementation configuration to be used for Key Deactivation procedure (Sections 3.2.3.3 and 5.4.2.3) interoperability testing and operation is:

* The Key ID fields of the Key Deactivation PDU data field structure shall have a size of 16 bit.
* The total length of the PDU (Header + Data Field) shall not exceed 81 octets (=deactivation of 32 session keys).
* The Length field shall indicate a number of N\*16, where N is the number of session keys to be deactivated, N<=32.



Figure ‑: Baseline Implementation Mode Key Deactivation Command PDU

* + - 1. Key Verification

The baseline implementation configuration to be used for Key Verification procedure (Sections 3.2.3.5 and 5.4.2.5) interoperability testing and operation is:

* The Key ID fields of the Key Verification PDU data field structures shall have a size of 16 bit.
* The total length of the Command PDU (Header + Data Field) shall not exceed 81 octets (=verification of 32 session keys).
* The Length field shall indicate a number of N\*16, where N is the number of session keys to be verified, N<=32.
* The Key Validity fields shall have a length of 4 bit.
* The Key State fields shall have a length of 4 bit.
* The total length of the Reply PDU (Header + Data Field) shall not exceed 81 octets (=verification of 32 session keys).



Figure ‑: Baseline Implementation Mode Key Verification Command PDU



Figure 7‑5: Baseline Implementation Mode OTAR Reply PDU

* + 1. CRC-32 specification to protect Encrypted Key Block in the OTAR procedure of the Extended Procedures
       1. CRC-32 Encrypted Key Block Error Control Field Coding Procedures
          1. Conventions

The following convention is used to identify each bit in an *N*-bit field. The first bit in the field to be transmitted (i.e., the most left justified when drawing a figure) is defined to be ‘Bit 0’; the following bit is defined to be ‘Bit 1’ and so on up to ‘Bit *N*–1’. When the field is used to express a binary value (such as a counter), the Most Significant Bit (MSB) shall be the first transmitted bit of the field, i.e., ‘Bit 0’ (see figure 7‑6).



Figure 7‑ : Bit Numbering Convention

* + - * 1. CRC-32 Encoding Procedure

For the encoding procedure, the (*n–*32)-bit Encrypted Key Block shall be the information message.

The encoding procedure shall accept an (*n–*32)-bit Encrypted Key Block and generate a systematic binary (*n*,*n*–32) block code by appending a 32-bit Cyclic Redundancy Check (CRC-32) as the final 32 bits of the Encrypted Key Block.

NOTES

1. The Bit Numbering Convention as specified in 1.1 is applicable below.

If M(*X*) is the (*n*–32)-bit information message to be encoded expressed as a polynomial with binary coefficients, with the first bit transferred being the most significant bit M0 taken as the coefficient of the highest power of *X,* then the equation for the 32-bit Cyclic Redundancy Check, expressed as a polynomial R(*X*) with binary coefficients, shall be:

R(*X*) = [*X*32 ∙ M(*X*)] modulo G(*X*)

where G(*X*) is the generating polynomial given by:

G(*X*) = *X*32 + *X*23 + *X*21 + *X*11 + *X*2 + 1

and where the first transferred bit of the Cyclic Redundancy Check is the most significant bit R0 taken as the coefficient of the highest power of *X*.

The *n*-bit CRC-32-encoded block, expressed as a polynomial C(*X*) with binary coefficients, shall be:

C(*X*) = *X*32 ∙ M(*X*) + R(*X*)

The (*n*–32) bitsof the message are input in the order *M*0,…, *Mn*−33, and the *n* bits of the codeword are output in the order *C*0,…, *Cn*−1 = *M*0,…, *Mn*−33, *R*0,…, *R*31.

NOTE – A possible implementation of an encoder is described in figure **D‑7**. For each frame, the shift register is preset to the ‘all zero’ state prior to encoding. This initialization differs from that performed for the 16-bit CRC described in other CCSDS books, for which the cells are initialized to all ‘ones’. The ganged switch is in position 1 while the information bits are being transferred and in position 2 for the 32 Cyclic Redundancy Check bits.



**Figure** **D‑7** **: A Possible Implementation of the CRC-32 Encoder**

* + - * 1. CRC-32 Decoding Procedure

The decoding procedure shall accept an *n*-bit received codeword, including the 32-bit Cyclic Redundancy Check, and generate a 32-bit syndrome.

An error shall be detected if and only if at least one of the syndrome bits is non-‘zero’.

The received block C\*(*X*) shall equal the transmitted codeword C(*X*) plus (modulo two) the *n*-bit error block E(*X*), C\*(*X*) = C(*X*) + E(*X*), where both are expressed as polynomials of the same form, i.e., with the most significant bit C0 or E0 taken as the binary coefficient of the highest power of *X*.

With C\*(*X*) being the *n*-bit received codeword with the first transferred bit being the most significant bit C0\* taken as the coefficient of the highest power of *X*, then the equation for the 32-bit syndrome, expressed as a polynomial S(*X*) with binary coefficients, shall be:

S(*X*) = [*X*32 ∙ C\*(*X*)] modulo G(*X*)

The syndrome polynomial will be ‘zero’ if no error is detected, and non-‘zero’ if an error is detected, with the most significant bit S0 taken as the coefficient of the highest power of *X*.

NOTE – A possible implementation of the syndrome polynomial generator is described in figure **D‑8**. For each frame, the shift register cells are initialized to ‘zero’. This initialization differs from that performed for the 16-bit CRC described in other CCSDS books, for which the cells are initialized to ‘all ones’. The codeword includes *n* bits, i.e., (*n*–32) information message bits plus the 32 bits of the Cyclic Redundancy Check. All the *n* bits of the codeword are clocked into the input and then the storage stages are examined. For an error-free block, the contents of the shift register cells will be zero. A non-zero content indicates an erroneous block.

****

**Figure** **D‑8** **: A Possible Implementation of the CRC-32 Decoder**

* 1. Security Associations Management Service

This section 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:

#### Start SA

* The SPI field shall be 16 bits in length.
* The GVC ID / GMAP ID field shall be a multiple of 32 bits in length. Each GVC ID / GMAP ID entered shall consist of a concatenation of the following values from the underlying Space Link Protocol:
  + Transfer Frame Version Number (4 bits, right-justified);
  + Spacecraft ID (16 bits, right-justified);
  + Virtual Channel ID (6 bits, right-justified);
  + Multiplexer Access Point ID (6 bits).
* The total length of the PDU (Header + Data Field) shall not exceed 988 octets. This is to ensure that the complete Command PDU fits into one frame of 1024 octets.



Figure 7‑9: Baseline Mode Start SA PDU

#### Stop SA

* The SPI field shall be 16 bits in length.



Figure ‑: Stop SA PDU

#### Rekey SA

* The SPI field shall be 16 bits in length.
* The New Encryption Key ID field shall be 0 bits in length.
* The New Authentication Key ID field shall be 16 bits in length, right-justified.

Figure ‑: Rekey SA PDU

#### Expire SA

* The SPI field shall be 16 bits in length.



Figure ‑: Expire SA PDU

#### Set Anti-Replay Counter (ARC)

* The SPI field shall be 16 bits in length.
* The New ARC Value field shall be 96 bits in length, right-justified.

Figure ‑: Set ARC PDU

* NOTE: Since the ARC is identical to the IV for the SDLS baseline mode AES-GCM algorithm, executing this procedure will set the IV.

SDLS Monitoring and Control Service

This section specifies the baseline implementation mode for the Monitoring and Control Service Extended Procedures.

* + 1. SDLS Monitoring and Control Services Parameters
       1. Ping
* No specific configuration needed.
  + - 1. Read Sequence Number

The baseline implementation configuration to be used for Read Sequence Number procedure interoperability testing and operation is:

* For TC : The length of the Sequence Number is 0x0004 (32 bits)
* For TM / AOS : The length of the Sequence Number / Initialization Vector is 0x000C (12 octets, 96 bits)
  + - 1. Alarm Flag Reset

The baseline implementation configuration to be used for Alarm Flag Reset procedure interoperability testing and operation is:

* No specific configuration needed.