

Report Concerning Space Data System Standards

|  |
| --- |
| SDLS Extended Procedures –  concept and rationale |

AUTHORITY

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  | Issue: |  |  |
|  | Date: |  |  |
|  | Location: | Not Applicable |  |
|  | | | |

**(WHEN THIS INFORMATIONAL REPORT 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 reflects the consensus of technical panel experts from CCSDS Member Agencies. The procedure for review and authorization of CCSDS Reports is detailed in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4).

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

This document is a CCSDS Report, which contains background, rationale and a concept of operation to support the CCSDS Recommended Standard on the Space Data Link Security Protocol (reference [1]).

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This document is therefore subject to CCSDS document management and change control procedures which are defined in the *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.
* British National Space Centre (BNSC)/United Kingdom.
* 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 e.V. (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.

Observer Agencies

* Austrian Space Agency (ASA)/Austria.
* Belgian Federal Science Policy Office (BFSPO)/Belgium.
* Central Research Institute of Machine Building (TsNIIMASh)/Russian Federation.
* Centro Tecnico Aeroespacial (CTA)/Brazil.
* 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.
* European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
* European Telecommunications Satellite Organization (EUTELSAT)/Europe.
* 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.
* MIKOMTEK: CSIR (CSIR)/Republic of South Africa.
* Ministry of Communications (MOC)/Israel.
* National Institute of Information and Communications Technology (NICT)/Japan.
* National Oceanic and Atmospheric Administration (NOAA)/USA.
* National Space Organization (NSPO)/Chinese Taipei.
* Naval Center for Space Technology (NCST)/USA.
* Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
* Swedish Space Corporation (SSC)/Sweden.
* United States Geological Survey (USGS)/USA.

DOCUMENT CONTROL

|  |  |  |  |
| --- | --- | --- | --- |
| **Document** | **Title and Issue** | **Date** | **Status** |
|  | Space Data Link Security (SDLS) Extended Procedures –  Concept and Rationale,  , |  | Original issue |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

CONTENTS

Section Page

[DOCUMENT CONTROL iii](#_Toc39222608)

[CONTENTS v](#_Toc39222609)

[1 Introduction 1-1](#_Toc39222610)

[1.1 purpose 1-1](#_Toc39222611)

[1.2 scope 1-1](#_Toc39222612)

[1.3 organization of this report 1-2](#_Toc39222613)

[1.4 conventions and definitions 1-2](#_Toc39222614)

[1.5 References 1-2](#_Toc39222615)

[2 Overview and Rationale 2-4](#_Toc39222616)

[2.1 SDLS 2-4](#_Toc39222617)

[2.2 Extended Procedures 2-4](#_Toc39222618)

[2.3 Design goals and constraints 2-5](#_Toc39222619)

[2.3.1 compatibility with sdl services 2-5](#_Toc39222620)

[2.3.2 requirements 2-6](#_Toc39222621)

[2.4 Key Management 2-6](#_Toc39222622)

[2.4.1 Justification 2-6](#_Toc39222623)

[2.4.2 Summary of capabilities 2-6](#_Toc39222624)

[2.5 Security Association (SA) Management 2-6](#_Toc39222625)

[2.5.1 Justification 2-6](#_Toc39222626)

[2.5.2 Summary of capabilities 2-7](#_Toc39222627)

[2.6 Monitoring & Control 2-8](#_Toc39222628)

[2.6.1 Justification 2-8](#_Toc39222629)

[2.6.2 Summary of capabilities 2-8](#_Toc39222630)

[2.7 Frame Security Report (FSR) 2-8](#_Toc39222631)

[2.7.1 Justification 2-8](#_Toc39222632)

[2.7.2 Summary of capabilities 2-9](#_Toc39222635)

[3 Concept of Operation 3-10](#_Toc39222638)

[3.1 Overview 3-10](#_Toc39222639)

[3.1.1 Protocol Data Units 3-10](#_Toc39222640)

[3.2 Key Management 3-13](#_Toc39222641)

[3.2.1 Cryptographic Key LifecyCle 3-13](#_Toc39222645)

[3.2.2 Key Identifier 3-14](#_Toc39222646)

[3.2.3 Procedures implementing lifecycle transitions 3-14](#_Toc39222647)

[3.2.4 Key renewal schemes 3-15](#_Toc39222648)

[3.2.5 Procedures for confirming key information 3-17](#_Toc39222649)

[3.2.6 Key Management concept of operations 3-18](#_Toc39222650)

[3.2.7 Contingency and off nominal scenarios 3-18](#_Toc39222651)

[3.3 Security Association Management 3-18](#_Toc39222652)

[3.3.1 Guidelines on planning & assigning Security Associations 3-19](#_Toc39222653)

[3.3.2 Normal procedures for SA management 3-20](#_Toc39222656)

[3.3.3 Implementing SA life cycle with the EP procedures 3-23](#_Toc39222657)

[3.3.4 Contingency and off-nominal scenarios 3-23](#_Toc39222658)

[3.4 Monitoring & Control 3-23](#_Toc39222659)

[3.4.1 Monitoring & Control Procedures 3-23](#_Toc39222660)

[3.4.2 Security Log 3-25](#_Toc39222661)

[3.4.3 self-test 3-26](#_Toc39222662)

[3.5 Frame Security Report (FSR) 3-27](#_Toc39222663)

[3.5.1 Relation to space link protocols 3-27](#_Toc39222665)

[3.5.2 How to interpret the flags 3-28](#_Toc39222666)

[3.5.3 Concept of operations for handling alarm flags (e.g.: discriminating transmission problems from security events/attacks, using FSR as a first stage in troubleshooting on the link, …) 3-29](#_Toc39222667)

[4 design concepts 4-31](#_Toc39222668)

[4.1 Error handling 4-31](#_Toc39222669)

[4.1.1 signaling errors 4-31](#_Toc39222670)

[4.1.2 Execution errors 4-31](#_Toc39222671)

[4.2 redundancy 4-31](#_Toc39222672)

[4.2.1 Physical cross-strapping 4-31](#_Toc39222673)

[4.2.2 Logical cross-strapping 4-32](#_Toc39222674)

[4.3 Failure handling 4-33](#_Toc39222675)

[4.3.1 EP PDU on-board path/processing (in-band vs out of band signaling, …) 4-34](#_Toc39222676)

[4.4 Mission Scenarios 4-34](#_Toc39222677)

[4.4.1 “Classical” ground-space Scenario 4-34](#_Toc39222678)

[4.4.2 Single Spacecraft, multiple links to ground 4-34](#_Toc39222679)

[4.4.3 Inter-satellite link Scenario 4-34](#_Toc39222680)

[4.4.4 Constellations of multiple spacecraft 4-39](#_Toc39222681)

[4.5 Relationship to other CCSDS STandards 4-39](#_Toc39222682)

[4.5.1 Cryptographic Algorithms (352.0-B) 4-39](#_Toc39222683)

[4.5.2 Symmetric Key Management (354.0-M) 4-40](#_Toc39222684)

[4.5.3 ??? 4-41](#_Toc39222685)

[4.6 BASELINE MODES A-1](#_Toc39222686)

[4.6.1 Directives Included A-1](#_Toc39222687)

[4.6.2 Directives Excluded A-1](#_Toc39222688)

# Introduction

## purpose

This Report has been developed to present the concept and rationale of the CCSDS Recommended Standard on the Space Data Link Security Protocol Extended Procedures. This Green Book will enable mission designers and protocol implementers to:

1. Understand the purpose and usage of the SDLS Extended Procedures;
2. Select appropriate procedures and parameters for the mission;
3. Cover nominal and contingency scenarios;
4. Understand the performance and limitations of the Extended Procedures.

The Space Data Link Security (SDLS) Protocol [1] is a security protocol that implements user-selected Security Services to the data transported by the Space Data Link (SDL) protocol in space-to-ground and ground-to-space links. The SDLS protects the Service Data Units transported by the SDL protocol and, in addition, selected SDL protocol data structures taking into account compatibility constraints with SDL and Space Link Extension services.

The Recommended Standard for SDLS Extended Procedures [2] extends the SDLS protocol with a standardized set of auxiliary services for managing an implementation of the SDLS protocol. These EP services are categorized into Key Management, Security Association (SA) Management, and SDLS Monitoring & Control. Further, [2] specifies service interfaces and data structures for transport of EP service messages within the Space Data Link (SDL) protocols along with a security unit status reporting mechanism.

SDLS Extended Procedures encompass well-known procedures such as Over-The-Air Rekeying (OTAR) which are documented in this Green Book. Furthermore, this report describes the concept of operations and illustrates normal and contingency scenarios so that mission designers and protocol implementers can make optimal use of the SDLS EP recommendation.

## scope

The information contained in this Report is not part of the CCSDS Recommended Standards on the Space Data Link Security Protocol [1] and [2]. In the event of any conflict between the Recommended Standards and the material presented herein, the Recommended Standards shall prevail.

## organization of this report

Section 2 presents an overview of the Extended Procedures, the rationale for their development, and the major design goals and constraints;

Section 3 provides a concept of operation for using the protocol’s security services; in particular the transmission of EP PDUs within CCSDS protocol stacks, the logical order of EP operations, and its data structures, fields and functions are given;

Section 4 provides a discussion of key design concepts of the protocol, including handling of EP signaling errors and execution failures, implementation of redundancy, and off-nominal operations. It also discusses the use of SDLS within several example mission scenarios;

Annex A elaborates on the baseline implementations;

Annex B provides a list of acronyms and abbreviations.

## conventions and definitions

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

## References

The following documents are referenced in this Report. At the time of publication, the editions indicated were valid. All documents are subject to revision, and users of this Report are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS documents.

|  |  |
| --- | --- |
| [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] | *Space Data Link Security Extended Procedures*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 355.1-B-1. Washington, D.C.: CCSDS, February 2020. |
| [3] | *Information Security Glossary of Terms*. Issue 2. Recommendation for Space Data System Practices (Magenta Book), CCSDS 350.8-M-2. Washington, D.C.: CCSDS, February 2020. |
| [4] | *TM Space Data Link Protocol*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 132.0-B-2. Washington, D.C.: CCSDS, September 2015. |
| [5] | *TC Space Data Link Protocol*. Issue 3. Recommendation for Space Data System Standards (Blue Book), CCSDS 232.0-B-3. Washington, D.C.: CCSDS, September 2015. |
| [6] | *AOS Space Data Link Protocol*. Issue 3. Recommendation for Space Data System Standards (Blue Book), CCSDS 732.0-B-3. Washington, D.C.: CCSDS, September 2015. |
| [7] | *Unified Space Data Link Protocol*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 732.1-B-1. Washington, D.C.: CCSDS, October 2018. |
| [8] | *Symmetric Key Management*. Issue 1. Draft Recommendation for Space Data System Practices (Red Book), CCSDS 354.0-R-1. Washington, D.C.: CCSDS, June 2018. |
| [9] | *Space Missions Key Management Concept*. Issue 1. Report Concerning Space Data Systems Standards (Green Book), CCSDS 350.6-G-1. Washington, D.C.: CCSDS, November 2011. |
| [10] | *Space Data Link Security* *— Summary of Concept and Rationale.* Issue 1. Report Concerning Space Data Systems Standards (Green Book), CCSDS 350.5-G-1. Washington, D.C.: CCSDS, June 2018. |
| [11] | *Communications Operation Procedure-1.* Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 232.1-B-2. Washington, D.C.: CCSDS, September 2010. |
| [12] | *CCSDS Cryptographic Algorithms*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 352.0-B-2. Washington, D.C.: CCSDS, August 2019. |
| [13] | *The Application of Security to CCSDS Protocols*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 350.0-G-3. Washington, D.C.: CCSDS, March 2019. |
| [14] | *Security Guide for Mission Planners*. Issue 2. Report Concerning Space Data System Standards (Green Book), CCSDS 350.7-G-2. Washington, D.C.: CCSDS, April 2019. |
| [15] | Morris Dworkin. *Recommendation for Block Cipher Modes of Operation:*  *Galois/Counter Mode (GCM) and GMAC*. National Institute of Standards and  Technology Special Publication 800-38D. Gaithersburg, Maryland: NIST, November 2007. |
| [16] | *ECRYPT II Yearly Report on Algorithms and Keysizes(2011-2012),* ICT-2007-216676, European Network of Excellence in Cryptology II, Revision 1.0, 30 Sept 2012. |
| [17] | M.-J. O. Saarinnen. “*GCM, GHASH and Weak Keys”*. Cryptology ePrint Archive, Report 2011/202, http://eprint.iacr.org/2011/202, 2011. |
|  |  |

# Overview and Rationale

## SDLS

The Space Data Link Security (SDLS) Protocol [1] is a security protocol that implements user-selected Security Services to the data transported by the Space Data Link (SDL) protocol in space-to-ground and ground-to-space links. The SDLS protects the Service Data Units transported by the SDL protocol and, in addition, selected SDL protocol data structures taking into account compatibility constraints with SDL and Space Link Extension services. ***Figure 2‑1*** depicts the basic SDLS capabilities defined in [1].



*Figure 2‑1. Traffic encryption & authentication interface*

## Extended Procedures

The Recommended Standard for SDLS Extended Procedures [2] extends the SDLS protocol [1] with services for managing the security parameters of the space link. The purpose of SDLS Extended Procedures (EP) is to provide a standardized set of auxiliary services for managing an implementation of the SDLS protocol. These EP services are categorized into Key Management, Security Association (SA) Management, and SDLS Monitoring & Control. The SDLS EP specification also includes service interfaces and data structures for transport of EP service messages within the Space Data Link (SDL) protocols, and a security unit status reporting mechanism. ***Figure 2‑2*** depicts the additional capabilities defined in [2].



*Figure 2‑2. Extended Procedures directive interface*

## Design goals and constraints

### compatibility with sdl services

The SDLS standards [1] & [2] have been developed for use with the existing CCSDS TM, TC, AOS, and USLP Space Data Link Protocols defined in references [4], [5], [6], and [7].

As depicted in ***Figure 2‑2***, SDLS Extended Procedures specify a separate logical interface for managing the security of the space data link. They neither replace nor modify the behavior of the SDLS traffic encryption and authentication services defined in [1].

### requirements

SDLS Extended Procedures are designed to operate in a master-slave configuration. For nominal space-to-ground links, the master is the mission operations center. It is also possible to use the Extended Procedures for managing space-to-space links. In all cases, there is no negotiation between endpoints: all directives are issued from a predetermined master (referred to as Initiator in [2]) toward a predetermined slave (referred to as Recipient in [2]).

## Key Management

### Justification

CCSDS recommends a standard cryptographic key lifecycle and a set of key management procedures to enable proper generation, distribution, and handling of cryptographic keys for space missions [8]. The SDLS Extended Procedures for key management represents a specific implementation of these recommendations. In addition, CCSDS has produced general guidelines and practices on key management [9].

### Summary of capabilities

The SDLS security services rely on symmetric cryptosystems in order to operate properly. The following general key management schemes listed in [10] are supported by the Extended Procedures:

* Scheme 1: all session keys are pre-loaded on satellite before launch and cover the whole mission lifetime;
* Scheme 2: a subset of keys (master keys/key encryption keys (KEKs) and session/traffic protection keys) are pre-loaded on satellite before launch; additional session keys are uploaded in encrypted form during satellite operation (Over The Air Rekeying, OTAR);

## Security Association (SA) Management

The SDLS protocol provides encryption, authentication, or authenticated encryption for data link layer services of the TC, TM, AOS, and USLP protocols. Central to the operation of this protocol is the Security Association (SA), a data schema used at both sending and receiving ends of a space link for managing the session state of cryptographic parameters.

### Justification

The Security Association Management Service for the SDLS protocol is designed to carry out the most basic functions of Security Association setup, activation, status, and control necessary to command the configurable Security Association parameters of a remote (slave) system’s SDLS implementation into a state suitable for operations.

### Summary of capabilities

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.

High data rate 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 instantiation of Security Associations on demand. ***Figure 2‑3*** illustrates the state model and related directives for Security Associations.



*Figure 2‑3: Variable State Model for Security Association Management*

If a mission needs the capability to generate or upload new cryptographic keys (or sets of keys) during the mission lifetime (as in the case of Over-The-Air Rekeying (OTAR)), it also needs the capability to change individual Security Associations’ parameters to use new keys instead of any key originally pre-loaded prior to the start of the mission.

Although it is not expected to be common, some long-duration missions may desire the capability to replace existing Security Associations altogether. For example, this could be used in conjunction with re-programmable cryptographic systems (and redundant security units) in order to retire obsolete algorithms and carefully transition to use newer ones.

## Monitoring & Control

The SDLS Extended Procedures specify a set of service procedures for the monitoring and control of the slave (typically on-board) security function. The on-board security unit is an implementation of the SDLS security functions in hardware and/or software. The Monitoring & Control service procedures allow for nominal and contingency scenarios.

### Justification

The master (typically a mission operations center) needs to know the state of the slave (typically on-board) security unit. This state includes the history of security events, in order to be able to investigate encountered anomalies and detect potential attacks.

### Summary of capabilities

The set of Extended Procedures Monitoring & Control service procedures covers several aspects:

* Security unit health monitoring (Ping, Self-Test);
* Management of the log of security events (Log Status, Dump Log, Erase Log);
* General purpose monitoring and control of the on-board security unit (Alarm Flag Reset).

Several on-demand or on-event reporting mechanisms and corresponding messages are specified in the Extended Procedures. They provide non real-time or non-systematic reporting of the frame verification status at the receiving end of the SDLS secured uplink. They enable further investigation of security events occurring on-board.

## Frame Security Report (FSR)

The SDLS Extended Procedures specify [4][6][7]

### 

This Frame Security Report (FSR), which is the protocol data unit transmitted from the Recipient to the Initiator of an SDLS secured TC, AOS, or USLP uplink, provides the systematic, real-time mechanism by which the SDLS function at the receiving end reports the status of uplink TC, AOS, or USLP frame verification to the sending end.

The FSR is similar to the COP-1 Command Link Control Word (CLCW) [11] which provides real-time reporting of the status of uplink TC, AOS, or USLP frame acceptance by the COP-1 function to the sending end.

### 

### Summary of capabilities

The reporting capabilities of the FSR are the following:

* a persistent Alarm flag that will signal any uplink frame rejection by SDLS on-board function. This flag can be reset by the user once the rejection has been taken into account by the MOC.
* non-persistent Security Event Flags which enable to characterize security violation detected on the last received uplink frame: invalid Sequence Number, invalid MAC (failed authentication), invalid SA.
* SPI of the last received uplink frame
* Sequence Number (SN) of the last received uplink frame

The Alarm flag enables the systematic detection on the ground of any uplink frame rejection by SDLS. The latency of this detection is made as low as feasible by the transmission of the FSR in the OCF field which is carried in every downlink frame.

The Security Event Flags and the associated SPI/SN information enable the characterization of major security events occurring on any uplink frame, provided that the FSR is transmitted to ground at least for each uplink frame received (see section 3.5 for discussion of FSR transmission rate).

# Concept of Operation

## Overview

The SDLS Extended Procedures comprise optional capabilities for managing data link security, organized into three functional areas: Key Management, Security Association (SA) Management, and SDLS Monitoring & Control. ***Figure 3‑1*** below illustrates the complete set of Extended Procedures, grouped according to these areas.



*Figure 3‑1. EP Directives*

It is not necessary to implement the entire set of EP directives. Implementation of these functional areas’ Extended Procedures can be tailored to the needs of a mission, although within each functional area, many of the directives correspond to one another closely and the presence of related directives is logically expected. The Extended Procedures specification [2] Protocol Implementation Conformance Statement (PICS) provides further detail about which EP directives should be implemented together.

### Protocol Data Units

#### Use of Packet Service

All SDLS Extended Procedures directives (both Command and Reply PDUs) are transmitted using the Packet Service (VC Packet in AOS and TM, or MAP Packet in TC and USLP) of each supported CCSDS Space Link Protocol. ***Figure 3‑2*** depicts the insertion and extraction of EP PDUs by the supporting Space Link Protocols.



*Figure 3‑2. EP PDU relation to Data Link layer processing*

Specification of the Packet Service for the transmission of SDLS Extended Procedures directives does not mandate any single design for how the PDU interface is implemented. ***Figure 2‑2*** depicts the SDLS EP Command and Reply interface as directly attached to forward and return link processing. This would be a potential architecture for a security function intended to be embedded within the onboard baseband signal processor (e.g. as an integrated hardware unit or even as a software-defined radio).

As depicted in ***Figure 3‑3*** below, it is similarly possible (and certain missions may prefer it) to route SDLS EP packets through the packet processing function of the onboard computer, used in common by all packet types, for simplicity of implementation and validation.



*Figure 3‑3. EP PDU interface via onboard computer*

#### Delivery of Protocol Data Units

SDLS EP PDU exchanges do not contain any built-in mechanism for assuring reliable delivery. Directives lost in transmission will go undetected, unless the Initiator receives telemetry reporting from the Recipient of EP Command PDUs as they are received and executed.

Acknowledgement in telemetry of EP Command PDUs received by the Recipient is a necessary function for maintaining integrity of SDLS, but its implementation is mission-specific.

#### Protection of Protocol Data Units

Because SDLS Extended Procedures provide an in-line capability to modify operational attributes of the onboard security function, it is necessary to protect against insertion of unauthorized Command PDUs. All EP PDUs are transmitted over a SDLS channel protected by authentication or authenticated encryption. (Even though a few directives e.g. OTAR include authentication and/or encryption within the PDU itself, protection of the EP PDU channel should be employed equally across all EP PDU exchanges).

The decision by a mission to implement authenticated encryption, versus authentication-only, for the transmission of EP Service PDUs should be based upon an overall analysis of threats and risks to the mission. Certain PDUs, e.g. Key Inventory or Dump Log Reply PDUs, or Set ARSN Command PDU, could reveal information to a hostile third party which a mission would prefer to keep private. Because the same threats are generally applicable to other spacecraft monitoring and control data exchanges, any risk-based decision is likely to be similarly applied to the protection of telecommands and/or telemetry.

Additional protection of EP PDUs between the first SDLS check point on-board the spacecraft (Security Unit) and the final destination on-board (SDLS EP processor), might be needed for certain mission. Refer to Security GB (350.0-G)

## Key Management

This section is outlining the concept of operations for the key management part of the SDLS Extended Procedures [2].



*Figure 3‑4. Key Management directives*

### Cryptographic Key LifecyCle

The (symmetric) cryptographic key is a core component in every cryptographic operation. It represents the secret that is shared between communication partners and thus forms the basis for any authentication, integrity, and confidentiality services that the communication partners agree to implement.

A cryptographic key is governed by a state-based lifecycle as defined in §3.2 of reference [8]. A key is used differently, depending upon its state in the key’s lifecycle. Key states are applicable system-wide, as opposed to the point of view of a single cryptographic module. SDLS Extended Procedures support most (but not all) of the key states and transitions from [8], as depicted in ***Figure 3‑5*** below.



*Figure 3‑5. Key states and transitions*

A detailed description of the various key states is included in the CCSDS Symmetric Key Management recommendation [8] and not repeated here. It needs to be pointed out however that the SDLS Extended Procedures do not implement:

* the optional Suspended state. In the foreseen use cases for the SDLS protocol, a key suspension does not represent a credible operational scenario.
* the Compromised state. In the SDLS Extended Procedures, the Compromised state applies only to the Initiator.

### 

, – that is, there are not If partitioning is done, it does provide a potential method of implementing a check against inadvertently selecting master keys for use as session keys.

### Procedures implementing lifecycle transitions

This section discusses the subset of the key management-related extended procedures that is directly related to state transitions in the key lifecycle. These procedures are:

• Key Activation

• Key Deactivation

• Key Destruction

#### Key Activation

The Key Activation procedure implements the transition of one or more cryptographic keys from pre-activation to active state. This transition is a pre-requisite for operational use of the key for cryptographic operations.

The only parameter of the key activation commanding directive is a set of Key IDs that indicate to the Recipient which cryptographic keys should be transitioned from pre-active to active state. The Key Activation commanding directive is an atomic transaction: i.e., either all referenced keys are successfully transitioned, or none.

#### Key Deactivation

The Key Deactivation procedure implements the transition of one or more cryptographic keys from Active to Deactivated state. All keys that undergo the transition cannot then be used for cryptographic operations on new data, but remain physically present on the Initiator and Recipient side. This means it is still possible, for example, to decrypt legacy data that has been encrypted with deactivated keys.

The only parameter of the Key Deactivation commanding directive is a set of Key IDs that indicate to the Recipient which keys should be transitioned from Active to Deactivated state. The Key Deactivation commanding directive is an atomic transaction: i.e., either all referenced keys are successfully transitioned, or none.

#### Key Destruction

The Key Destruction procedure implements the transition of one or more cryptographic keys from Deactivated to Destroyed state. As a consequence of this transition, the cryptographic keys at least on the Recipient side are physically removed/deleted or overwritten with random data. The Initiator may keep a copy of the key for legacy purposes however the key is no longer part of the active SDLS Extended Procedures implementation. The Key ID of any destroyed key is considered available again and can be reassigned to a new key.

The only parameter of the Key Destruction commanding directive is a set of Key IDs that indicate to the Recipient which cryptographic keys should be transitioned from deactivated to destroyed state. It should be noted that the Key Destruction commanding directive is an atomic transaction: i.e., either all referenced keys are successfully destroyed, or none.

### Key renewal schemes

As described in Section 2.4, the SDLS Extended Procedures support two main key renewal schemes:

* Scheme 1: No key re-generation during the lifetime of the mission
* Scheme 2: Over-the-air rekeying (OTAR)

The decision which scheme is to be implemented by a mission is based on a risk assessment and trade-off that needs to be done as part of the mission definition. The CCSDS Security Guide for Mission Planners [14] provides background information on performing such risk assessments.

#### Scheme 1: No key re-generation during the lifetime of the mission

In this scheme all cryptographic keys that will be used during the lifetime of the mission are stored at the Initiator and Recipient side (in pre-activation state) before the mission becomes operational. No refresh or replacement of keys is foreseen during the lifetime of the mission. As a consequence, any key that reaches the deactivated or destroyed state of the key lifecycle will not be available anymore for the remainder of the mission lifetime. Any key that is corrupted or suspected to be corrupted cannot be replaced with a new key.

For many missions, this scheme is attractive due to its operational simplicity. Any mission which is implementing scheme 1 will not implement the OTAR procedure. However, it needs to be noted that this scheme also comes with a significant risk. Any weakness that is discovered with keys after launch (up to and including corruption or compromise) can no longer be mitigated.

The key hierarchy under this scheme is usually flat, i.e. all cryptographic keys are session keys (see [8]).

#### Scheme 2: Over-the-air-rekeying

Unlike scheme 1, the Over-the-air-rekeying (or OTAR) scheme allows for transmission of newly generated keys from the Initiator to the Recipient during mission operations using the scheme-specific OTAR Extended Procedure. OTAR requires a two-tier key hierarchy composed of master keys at the higher level and session keys at the lower level. Both key types and the hierarchy concept is described in [8].

The concept of OTAR involves the upload of newly generated session keys from the Initiator to the Recipient in a secure way, using a master key as the shared secret. The OTAR procedure allows multiple session keys to be uploaded at the same time. The maximum number of keys possible for each OTAR command directive is limited only by the packet size of the Space Packet that transports the procedure.

The procedure assumes that the Initiator has the capability to securely generate any number of session keys. How this is done is outside the scope of the SDLS Extended Procedures. The availability of the number of session keys to send to the Recipient is one of the preconditions for the OTAR procedure. The second precondition is the availability of a master key in active state to perform the authenticated encryption of the session keys so that they are protected during transmission.

Each session key uploaded using OTAR is accompanied by a Key ID. This is necessary in order for the Recipient to understand which Key ID needs to be assigned to a newly updated key. As mentioned before, it is possible that a newly uploaded session key is replacing a destroyed key at the Recipient side and re-using the Key ID. The (session key, key ID) pairs are then authenticated and encrypted using a master key (identified by the master Key ID). Depending on the cryptographic algorithm being used (AES-GCM in baseline mode), an Initialization Vector (IV) may need to be transmitted as part of the directive.

When an OTAR directive is received on the Recipient side, the (session key, Key ID) pairs are first authenticated and decrypted and then the individual new session keys are stored in Pre-Activation state in the memory slots identified by their Key ID.

Note that the OTAR Command PDU always contains a MAC. If transmitted over a communication channel using an SDLS Authenticated SA, a MAC would also be present at the transfer frame (data link) layer. While this ‘double’ authentication appears redundant, it is necessary because the two occur at different layers. If, as shown in ***Figure 3‑3***, EP Command PDUs are routed through the onboard computer, then any transfer frame MAC would be removed before the Command PDU arrived at the Extended Procedures directive interface. The OTAR PDU contents would lack integrity protection during its onboard routing and verification processing within the security unit. Similarly, an integrity and confidentiality protection is needed for the OTAR PDUs on the ground (in the Mission Operation Center).

### Procedures for confirming key information

This section discusses the subset of the key management-related Extended Procedures for maintaining the accuracy of the remote Recipient’s key database. These procedures are:

• Key Verification

• Key Inventory

#### Key Verification

The Extended Procedures include a mechanism to verify one or more keys stored at the Recipient side. In a space mission this is important since the space environment may result in bit flips or other unwanted modifications of keys stored onboard the spacecraft. Therefore, it is recommended to verify a key before it is being used for the first time.

The Key Verification directive uses a challenge-response method to verify keys stored at the Recipient side. For each key to be verified, a random number is generated and sent to the Recipient along with the respective Key ID. Upon reception, the Recipient will encrypt the random number using the key indicated by the Key ID and send the result back to the Initiator. The Initiator uses its copy of the respective keys to perform the same encryption operation and compare the results. A match means that the keys on Initiator and Recipient side are identical. A mismatch indicates a problem that would then need to be followed up by the operator. In order to minimize session key exposure, it is recommended to verify a session key immediately prior to its usage for cryptographic operations.

Note: the Key Verification procedure necessitates over-the-air use of the specified key(s). For this reason, it is carried out only on keys already in the Active state, as shown in ***Figure 3‑4***. A mission could, if desired, explicitly tie the Key Activation and Key Verification procedures together in implementation, such that a single spacecraft command resulted in both Command PDUs being issued in sequence to the onboard security function. It should be noted that Key Verification will imply start of cryptoperiod of the verified key.

#### Key Inventory

In missions where SDLS EP are used for performing key management tasks remotely, it is useful for the Initiator to obtain key status information from the Recipient being managed in order to compare the status against its own local key database. If the Recipient end is out of synchronization with the Initiator – i.e., the Recipient key states do not match the ground’s expectation – it should prompt the issuance of EP Command PDUs to direct the onboard into the desired state and correct the mismatch.

The Key Inventory directive is used to query the Recipient for its local key state information for a range of Key IDs. The Initiator provides a numerical range of one to *N* Key IDs for which to return the local key state. The Recipient then replies with a list of Key ID-and-state pairs corresponding to that range. The returned pairs provide one of the SDLS EP-supported key states (pre-activation, active, deactivated, destroyed) for each known Key ID within the specified range. Nonexistent Key IDs within the range are omitted from the reply. The values corresponding to each defined key state are mission-specific metadata. Details of how keys are stored in onboard memory are mission-specific and out of scope of this document.

### Key Management concept of operations

#### Interaction between Key Management and SA Management

SDLS Security Associations are dependent upon the existence of cryptographic keys eligible for operational use. Determining which keys are eligible for operational use is the task of key management.

Because it is not mandatory to implement the entire set of SDLS EP directives, there are directives in the Key Management and Security Association Management service groups which are logically related, yet separate because they carry out distinct functions. When the two service groups are implemented together, there is a sequential relationship between them as depicted in ***Figure 3‑1***.

It is a precondition of the Rekey SA directive that its specified key is in the Active state. The Key Activation directive enables a key for operational use, therefore it would be expected to precede any Rekey SA directive to associate the same key with a SA. Likewise, the Key Deactivation directive disables a key for operational use, and would therefore be expected to follow the Expire SA directive. A mission could, if desired, explicitly tie these operations together in implementation, such that a single spacecraft command resulted in both Command PDUs being issued in sequence to the onboard security function.

#### Use of master keys

Master keys and traffic (session) keys may be indistinguishable in terms of key format, but each serves a distinct purpose. To avoid potential compromise, keys of one type should not be used for the functions of another type. SDLS Extended Procedures use master keys only for the OTAR procedure. SDLS Security Associations use keys only for traffic encryption and/or authentication.

The loss or corruption of a master key is a serious event. It is catastrophic if OTAR cannot be used to upload new keys (e.g. including a new master key) because there is no usable master key currently onboard. To protect against this contingency, it is practical to install several master keys onboard in advance.

### 

## Security Association Management

This section is outlining the concept of operations for the Security Association management part of the SDLS Extended Procedures [2].



*Figure 3‑6. SA Management directives*

### Guidelines on planning & assigning Security Associations

#### SAs for nominal traffic

The Security Parameter Index is a 2-byte field of the Security Header, so the number of available SAs per Master Channel has an upper bound of 216 (65536). The actual number of SAs that an implementation needs to assign and prepare for use is notionally equivalent to the number of keys that the spacecraft’s security unit is capable of storing simultaneously.

Specific ranges of SPI values are sometimes assigned for operational convenience to mission-specific operational use cases, where there are use cases whose traffic protection requirements are not interchangeable (e.g. subdivision into SAs used for spacecraft housekeeping and SAs used for private payload data, or SAs used for testing in ‘clear mode’).

#### SAs for SDLS EP traffic and other special uses

SDLS EP traffic can be carried over the same SAs used by nominal traffic, and routed by normal VC or MAP packet processing to the correct remote security unit for PDU processing. It is critical, however, that SDLS EP directives never modify the same SA currently being used to transmit their own EP PDUs (doing so could interrupt the processing of EP PDUs in the middle of a sequence of EP operations, or cause loss of cryptographic synchronization between sender and receiver).

Alternatively, SDLS EP traffic can be carried over different SAs not used by nominal traffic. In any case, VC or MAP packet processing is still necessary for routing EP traffic to the security unit for PDU processing. The baseline mode of SDLS EP ([2] Annex D) mandates the use of those 2 reserved SPI for exchanging EP services PDUs over the space link.

### Normal procedures for SA management

#### Preparing SA for first use

Many SA service parameters are managed. Each SA must specify the values of these parameters, whether implicitly via pre-loaded static definition or explicitly via EP directive. This information is collectively known as the SA database, although that term does not imply a RDBMS-type implementation.

SA parameters which are fixed at the time of creation and do not change thereafter:

a) Security parameter index (SPI);

b) SA Service Type;

c) The field lengths for Security Header and Security Trailer fields;

e) Encryption cipher suite length and identifier;

f) Initialization vector (IV) length;

g) Authentication cipher suite length and identifier;

h) Authentication bit mask length and value;

i) Anti-replay sequence number (ARSN) length; and

j) Anti-replay sequence number window length.

SA parameters which change during use, but must be provided with initial values:

f) Initialization vector (IV) initial value;

i) Anti-replay sequence number (ARSN) initial value; and

j) Anti-replay sequence number window value.

Static pre-loading commonly initializes all of the above managed parameters. If implemented, the Create SA directive accomplishes the same function. The Create SA directive instantiates a new SA in the Unkeyed state containing the initial parameters and context supplied in the directive.

Static pre-loading also commonly associates cryptographic keys with SAs. Since the Create SA directive (if used) does not associate cryptographic keys with the SA, the Create SA directive for an SA should be followed by the Rekey SA directive to transition from Unkeyed to Keyed state, so that the SA is ready for activation later via the Start SA directive.

As depicted in ***Figure 2‑3***, to replace a Security Association altogether, the two EP directives Delete SA and Create SA are needed. The Delete SA directive erases all existing parameters of the SA and its state information, so that the specified Security Parameter Index no longer references any defined SA at all. The Create SA directive can then be used to instantiate a new SA which reuses the SPI previously belonging to the deleted SA.

#### Changing cryptographic keys associated with a SA

If a mission needs the capability to generate or upload new cryptographic keys (or sets of keys) during the mission lifetime (as in the case of Over-The-Air Rekey (OTAR)), it also needs the capability to change individual Security Associations’ parameters to use new keys in place of keys originally pre-loaded prior to the start of the mission.

As depicted in ***Figure 2‑3***, to associate a new key with a Security Association, the two EP directives Expire SA and Rekey SA are employed. The SA’s existing key (presumably not to be used anymore) is removed from the SA via the Expire SA directive, which transitions the SA from Keyed state into Unkeyed state. The new key is associated with the SA via the Rekey SA directive, which transitions the SA from Unkeyed state to Keyed state.

#### Switching between SAs on a channel

As depicted in ***Figure 2‑3***, the most basic operation to perform upon Security Associations is to change which SA is used on a channel (thus, which cryptographic operations are to be performed, using which key, and so on).

This is carried out through the two EP directives Stop SA and Start SA. The Stop SA directive transitions the current (‘old’) SA from its Operational (in use) state into the Keyed (dormant) state. The Start SA directive transitions the ‘new’ SA from the Keyed state to the Operational state. It is expected that most implementations will carry out key changes during nominal operation by iterating through a set of SAs configured ahead of time, as depicted in ***Figure 3‑7*** below. In this illustration, the applicable channel (GVC ID or GMAP ID) remains constant, while the SA used on the channel is replaced at every key change event.



*Figure 3‑7. Operational key change scenario*

#### Seamless key change

Depending on the capabilities of the security units at sending and receiving ends, it is possible for the sending end to change which SA is used on a channel (and which key is in effect) from one frame to the next, without the receiving end dropping frames during the transition.

If frame-upon-frame key change is to be supported, both the sending end’s and receiving end’s security units should be capable of handling more than one active cryptographic session and key simultaneously. The receiving end’s security unit should be capable of supporting more than one SA in the Operational state on a given VC or MAP, so that when newly arrived frames indicate SAs different from previous frames, the security unit can correctly process without delay in transition. The ‘new’ SA should be transitioned into the Operational state at the Recipient end before the Initiator starts sending frames using the ‘new’ SA.

#### Query and modify SA parameters

Certain EP directives are provided to allow mission operations staff to adjust SA parameters in response to observed performance or unexpected behavior on the space link.

The SA Status Request directive queries the Recipient to report the current state of a specified SA. The SA Status Request Reply PDU returns the Procedure ID of the last executed state transition directive (e.g. Start SA, Stop SA, Expire SA, …) for the requested Security Association. The defined Procedure ID values ordinarily returned by this directive, as illustrated in Figure 5-10 of reference [2], implicitly embed both the previous (‘from’) and current (‘to’) states during the last state transition for the applicable SA. In the case of a mission which initializes SAs through static pre-loading prior to the mission, the ‘last state transition’ for SAs which have not yet received EP directives is undefined.

SDLS SAs providing Authentication service protect against “replay attacks” – the potential for an unauthorized party to record and retransmit previously transmitted frames, esp. commands to a spacecraft – by making use of a transmitted sequence counter and a managed “window” indicating how close a sequence number has to be to its expected value to be accepted as valid. The Set Anti-Replay Sequence Number (ARSN) and Set Anti-Replay Sequence Number Window directives are used to adjust the ARSN and ARSN window respectively. In case of loss of synchronization, or when switching to a previously used SA, it may be necessary to use the Read Anti-Replay Sequence Number (ARSN) directive to obtain the stored on-board value of the Sequence Number.

### Implementing SA life cycle with the EP procedures

### Contingency and off-nominal scenarios

#### (recovery SA, …)

## Monitoring & Control

This section is outlining the concept of operations for the SDLS Monitoring & Control part of the SDLS Extended Procedures [2].



*Figure 3‑8. SDLS Monitoring & Control directives*

### Monitoring & Control Procedures

The Extended Procedures define the following Monitoring & Control procedures:

* + Ping
  + Log Status
  + Dump Log
  + Erase Log
  + Self-Test
  + Alarm Flag Reset

#### Ping

The Ping procedure is a simple way to test that the on-board Security Unit is alive and able to process EP directives.

Upon reception of a Ping command, the on-board Security Unit shall generate a reply and send it to the ground. Neither the Ping command nor the Ping reply transmit a parameter.

#### Log Status

This procedure is related to the management of the security log on the Recipient side.

The Security Log contains a set of security event messages. The format of such messages is implementation-specific. In reply to the Log Status command, the on-board Security Unit shall generate a PDU containing the number of security event messages stored in the log, and the remaining space left in the log for storing new security event messages. The remaining space can be expressed as a value in octets or a percentage of the total log space available (choice left to the implementer).

The Log Status procedure is used by the mission operations center to monitor the usage level of the Security Log and take appropriate measures (i.e. Dump Log, then Erase Log).

See §3.4.2 for more information on the Security Log.

#### Dump Log

This procedure is related to the management of the security log on the Recipient side.

When necessary, the Initiator can use the Dump Log procedure to order the Recipient security unit to send the complete security log to ground. The Dump Log procedure does not affect the contents of the Security Log, which remains unchanged.

Please note that the Security Log may contain sensitive information.

See §3.4.2 for more information on the Security Log.

#### Erase Log

This procedure is related to the management of the security log on the Recipient side.

When the mission operations center has successfully received the security log (by using the Dump Log procedure), it can erase the on-board Security Log with the Erase Log procedure. Upon reception of the Erase Log command, the on-board security unit shall erase all the content of the security log, freeing the memory for new security event messages. It shall then reply to the command with a PDU containing the number of security event messages stored in the log (normally zero, but new events might have occurred) and the remaining space in the log.

See §3.4.2 for more information on the Security Log.

#### Self-Test

This procedure is used by the mission operations center to verify the health state of the on-board Security Unit by initiating a self-test.

See §3.4.3 for more information on the self-test.

#### Alarm Flag Reset

This procedure is associated with the Frame Security Report (FSR) management.

The FSR comprises a persistent Alarm Flag which indicate that at least one forward link Transfer Frame has been rejected by the on-board SDLS function since the last reset of the Alarm Flag. When the Alarm Flag has been taken into account by the mission operations center, the Alarm Flag Reset command is sent to the on-board Security Unit to order it to reset the Alarm Flag.

See §3.5.2 for more information on the Alarm Flag.

### Security Log

The Security Log is a means for recording important events seen by the on-board security unit. These events may affect the security of the protected links and are called security events. They are generated by the on-board security unit, either on its own or in reply to a received command. They can be generated when an error occurred or simply to log an important routine event. Such events could be but are not limited to:

* Frame received with a bad Sequence Number value (replay attack);
* Frame received generating a MAC error;
* Frame received pointing to an inactive SA (bad SPI);
* Sequence Number in use reaching its maximum value (need to change the key);
* Key corrupted;
* New key uploaded (OTAR monitoring);
* SA creation or deletion

Each Security Event Message shall provide, in addition to the type of event encountered, the minimum information necessary for forensic investigation (SA/SPI used, VCID, Sequence Number, time-tag, …).

As the on-board memory is limited, the Security Log may reach its maximum length. This length is implementation-dependent. When the Security Log is full, the on-board Security Unit can deal with new Security Events Messages in two different ways :

* + New Security Event Messages are stored and the oldest are lost, or
  + New Security Messages are lost and the oldest are kept in the log.

The behavior is not defined by CCSDS and the choice is left to the implementer.

The list of security events is not defined by CCSDS and left to the implementer. Simple on-board Security Units may have very simple security log or no log at all. Complex ones may have more verbose security logs.

In reply to a Dump Log command, the on-board Security Unit shall send the complete set of Security Event Messages stored in the log. The format of a Security Event Message is implementation-specific, but it is transmitted via the Dump Log command in a TLV format (hence allowing for future events definition while maintaining compatibility). However, the precise definition of the T, L,V fields is left to the implementer.

### self-test

The Self-Test Command is intended to initiate a series of predefined tests on the on-board Security Unit. These tests are not defined by CCSDS and left to the implementer’s choice. They are supposed to cover the overall Security Unit’s functionality and give confidence that the Security Unit is alive and performs well.

For example, the self-test for a Security Unit providing authenticated encryption on a TM link could be to compute an authenticated encrypted TM frame using a set of predefined test frame, Key and ARSN value. The result is then compared with a reference frame. If the computed frame and the reference frame are the same, the test passes. If not, the test fails.

Another complementary test could be a key database test, for example by associating a CRC with each key and storing it with the key in the key database. The key database test consists in computing sequentially each CRC and comparing it with stored CRC (note : the CRCs shall never go out of the Security Unit, as they give information on the key values. Generally speaking, any result of a computation involving a key – except the result of a crypto algorithm – shall never go outside the Security Unit as this would be a security breach).

Please note that the Self-Test command PDU is self-contained and does not allow to pass any parameter to the Security Unit. The Self-Test Reply PDU, however, has seven bits left in the Self-Test Result field that are not defined by CCSDS and may be used by the implementer to give more information on the test result.

The on-board Security Unit may or may not continue to process normal traffic while performing a Self-Test. This is not defined by CCSDS and left to the implementer’s choice.

## Frame Security Report (FSR)

The Frame Security Report (FSR) is a real-time report of the on-board SDLS function at the receiving end of a TC, AOS, or USLP uplink. FSR is transmitted over a TM, AOS, or USLP downlink in the transfer frames’ Operational Control Field (OCF) using the MC\_OCF or VC\_OCF services as defined in [4], [6], and [7].

SA Management and &

### 

Since a frame cannot contain two OCFs at the same time, insertion of the FSR is multiplexed with the insertion of the CLCW (report word from the Communications Operation Procedure-1 [11]) in implementations where both CLCW and FSR are used on the uplink. The multiplexing scheme is mission-specific. Nevertheless, the following considerations need to be taken into account:

#### Transmission rate of the FSR

Since all fields of the FSR, apart from the Alarm flag (discussed below), are non-persistent and updated with each uplink frame received by the on-board SDLS function, it is desirable that the transmission rate of the FSR on the downlink is equal to or higher than the uplink frame rate. This would guarantee that any security event detected by the on-board SDLS function is reported real-time to ground and that each security event can be associated unambiguously with the uplink frame that caused it (except for spurious frames transmitted by an attacker). This would enable limited investigation of security events by a mission operations center without the need for dumping and analysis of the on-board security log (if implemented) or usage of appropriate Extended Procedures directives.

#### Constraints where COP-1 CLCW is present

This constraint on FSR transmission rate can only be met if the downlink frame rate is significantly higher than the uplink frame rate since some downlink frames’ OCF will be reserved for CLCW. Although it is not necessary that the CLCW reporting rate (from the receiving end to the sending end) match the Transfer Frame rate (from the sending end to the receiving end), some minimum CLCW sampling rate is necessary for the proper operation of the COP.

#### Alarm Flag persistence

In case the above-mentioned constraint on the FSR transmission rate cannot be met, FSR contains a persistent Alarm flag which will in all cases inform the MOC that a security event has occurred on an uplink frame since the Alarm Flag was last reset. Determination of the type of security event(s) and of the uplink frame(s) involved will require sending appropriate SDLS Extended Procedures (EP) directives (see Monitoring & Control and SA management sections).

### How to interpret the flags

The purpose of the FSR being real-time reporting of the on-board SDLS function at the receiving end of a TC, AOS, or USLP uplink, all fields but one (the Alarm Flag) are non-persistent and updated at each uplink frame processed by the SDLS function.

The various information carried by the FSR can be interpreted as follows:

* Alarm flag (persistent): indicate that at least one uplink Transfer Frame has been rejected by the on-board SDLS function since the last reset of the Alarm Flag. This flag can be reset from the ground by sending the Alarm Flag Reset Command PDU. This flag being persistent guarantees that no security event detected on-board will go unnoticed at the MOC, whichever the FSR transmission rate.
* Security event flags (non-persistent): indicate the type of security event triggered by the last received uplink frame by the on-board SDLS function. They are updated at each uplink Transfer Frame processed by the SDLS on-board function; the uplink frame to which those flags relate is identified by the SPI and SN values transmitted in the last part of the FSR. Three generic types are reported:
  + Invalid SN: indicates that the SN carried by the last received uplink Transfer Frame by the on-board SDLS function was invalid (i.e. outside the SN window)
  + Invalid MAC: indicates that the MAC carried by the last received uplink Transfer Frame by the on-board SDLS function was invalid (i.e. did not match the MAC computed over the received Transfer Frame). This flag signals an integrity or authentication error on the related frame.
  + Invalid SA: indicate whether the last uplink Transfer Frame received by the SDLS on-board function failed SA verification, or carried an SPI pointing to a non-Operational SA, or an Operational SA associated with an non-Active key. (SA verification consists in checking that the SPI carried by the received uplink frame is pointing to an SA that is associated with the GVCID/GMAPID of that frame.)
* Last SPI used (non-persistent): indicates the SPI carried in the last received uplink Transfer Frame by the on-board SDLS function. This information, combined with the Sequence Number (SN) information, enables to identify unambiguously the last received uplink transfer frame to which above mentioned Security Event Flags relate.
* Sequence Number (SN) value (non-persistent): contains the 8 Least Significant Bits (LSB) of the Sequence Number (SN) carried in the last received uplink Transfer Frame by the on-board SDLS function. This SN is related to the Security Association that is pointed to by the SPI. SN combined with SPI unambiguously identify the last received uplink transfer frame to which above mentioned Security Event Flags relate.

### Concept of operations for handling alarm flags (e.g.: discriminating transmission problems from security events/attacks, using FSR as a first stage in troubleshooting on the link, …)

While operating a secured uplink to a spacecraft with limited contact time, it is of utmost importance to detect as promptly as possible any link disruption and be able to discriminate between the 2 main causes of disruption, namely:

* Transmission problems causing outage or frame rejection due to transmission errors
* Security events/attacks causing frame rejection by the on-board SDLS function

#### Discriminating transmission problems from security events/attacks

At the uplink receiving end, two frame validation processes operate in sequence:

* the Frame Acceptance & Reporting Mechanism (FARM) of the Communications Operation Procedure (COP-1) specified in [11]. This mechanism checks the validity of the uplink transfer frame based on:
  + the results of the decoding of the uplink channel code (presence of uncorrectable errors)
  + the results of the check of the frame CRC (FEC)
  + the result of the check of structure of the frame and validity of its header fields
* the on-board SDLS function specified in [1]. This mechanism checks the validity of the uplink transfer frame based on:
  + validity of the MAC which guarantees if valid the integrity and authenticity of the frame
  + validity of the SN which guarantees if valid that the frame is not a replay from a previously sent frame
  + validity of the SPI which guarantees if valid that an appropriate active key and SA have been used to protect the frame

The COP FARM and the SDLS function will reject/discard any frame that fails their respective checks. Both FARM and SDLS functions have their real-time reporting message that will enable the MOC to detect and discriminate between transmission errors and security events/attacks:

* Command Link Control Word (CLCW) for the COP
* Frame Security Report (FSR) for SDLS

Both types of report messages (CLCW and FSR) will be multiplexed in the Operational Control Field of downlink TM, AOS, or USLP frames. In most cases, downlink frame rate being significantly higher than uplink frame rate, at least one CLCW and one FSR can be transmitted for each uplink frame received enabling full real-time reporting of any communication (COP) or security (SDLS) related discarding of an uplink frame.

SDLS can secure forward link (e.g. uplink using TC, AOS or USLP) and/or return link (e.g. downlink using TM, AOS or USLP). Nevertheless, FSR will only be generated at the Recipient (typically Spacecraft) and sent to the Initiator (typically Spacecraft Control Center) to report the status of the Recipient Security Unit and Security Events detected at the Recipient. For the return link, there is no reporting mechanism from the Initiator (ground) to the Recipient (spacecraft) on the Security Events detected on the return link. This is operationally not needed.

# design concepts

## Error handling

### signaling errors

As noted in §3.1.1.2, SDLS Extended Procedures PDU exchanges do not contain any built-in mechanism for assuring reliable delivery. EP PDUs provide a limited set of directives and replies. The Extended Procedures do not, however, define the mechanism for acknowledging that EP PDU exchanges are received and executed. This is expected to be communicated using spacecraft telemetry.

### Execution errors

* n invalid
* error reporting

#### Key Management

The directives for Key Activation, Key Deactivation, and Key Destruction are explicitly stated to be atomic operations: if any part fails, the entire operation should be rolled back and treated as failed. The Key Verification directive, on the other hand, provides individual challenge responses for each key, and so is successfully executed even if verification fails for any individual key.

The Key Inventory directive may be useful for troubleshooting discrepancies after errors are encountered.

#### SA Management

Several of the SA Management Procedures direct the Recipient to verify preconditions before commencing any execution of operations. Wherever preconditions cannot be verified on-board, the operation should be halted and treated as failed. Status communicated back to the Initiator using telemetry should report the failed directive.

The SA Status directive may be useful for troubleshooting discrepancies after errors are encountered.

#### Monitoring & Control

The Log Status and Dump Log directives may be useful for troubleshooting discrepancies after errors are encountered.

## redundancy

Most spacecraft implementing SDLS will also have redundancy of frame processing and associated security units. It is possible to manage security units through the SDLS Extended Procedures such that secure communications is maintained while the security unit is actively being managed. Two implementation scenarios are discussed below.

### Physical cross-strapping

Scenario 1: Redundancy provided where each communications ‘string’ (i.e. each side of a redundant prime/backup pair) has its own independent virtual channel(s) so that RF data link traffic is directed explicitly to use a specific string (‘Side A’ vs. ‘Side B’).

In Scenario 1, each security unit is addressed using the virtual channels and SAs which belong to that string. Nominal RF traffic is addressed to one string ‘Side A’ using Side A’s virtual channels at the same time SDLS EP directives are addressed to the other string ‘Side B’ using Side B’s virtual channels. There is no ambiguity about which security unit can be addressed by a specific SDLS EP directive.



*Figure 4‑1. Physical cross-strapping*

### Logical cross-strapping

Scenario 2: Redundancy provided where both communications strings of a redundant pair share the same virtual channel(s), processing traffic in parallel so that RF traffic is output by whichever specific string currently acts as prime.

In Scenario 2, even though nominal RF traffic may continue along the virtual channel(s) shared by both strings, it is necessary that each string’s security unit be addressable using virtual channels and/or MAPs and/or SAs which belong to it alone. It is further necessary that, in addition to each communications string being able to route SDLS EP directives to its own security unit, that it also be able to route SDLS EP directives to the security unit belonging to the other string.

In this case, assignment of separate virtual channels (not used by nominal traffic) and/or SAs for each side’s security unit will prevent ambiguity about which security unit is addressed by a specific SDLS EP directive. Use of the two reserved SPI values (0 and 65535) to address separate security units is one possible method of accomplishing this.

* Use separate VCs from normal traffic, which route to specific Security Processor
* Use separate MAPs to route to specific Security Processor
* Partition SPI space to ensure SPI uniqueness across strings – SA state not shared across strings



*Figure 4‑2. Logical cross-strapping*

## Contingency and off-nominal scenarios

* Clear Mode
* Master Key recovery

### EP PDU on-board path/processing (in-band vs out of band signaling, …)

Discuss architectures where security unit implemented in series vs. implemented as a branch

## Mission Scenarios

### ‘Classical’ ground-space Scenario

For space-to-ground links in which a mission operations center controls a single spacecraft, the “Initiator” (master) and source of all uplinked EP Command PDUs is the mission operations center. The “Recipient” (slave unit being managed) is the spacecraft, and any applicable Reply PDUs and Frame Security Reports are sent via the telemetry downlink path.

Depending on the mission design, both the uplink and the downlink may implement SDLS and therefore it is necessary to indicate which directional link (up- or down-) is being managed by specific Security Association Management directives. The Initiator signals SA Management direction using the two-bit Service Group field in the Extended Procedures PDU. One Service Group value is provided for directives managing SAs which secure the ground-to-space link from Initiator to Recipient, and a separate Service Group value is provided for directives managing SAs which secure the space-to-ground link from Recipient to Initiator.

### Single Spacecraft, multiple links to ground

For space-to-ground links in which a single spacecraft has multiple downlink paths (e.g. separate high- and low-rate links), the two-bit Service Group field in the Extended Procedures PDU is insufficient for the Initiator to specify which of the available downlinks is applicable to a given SA Management directive. In this case, the applicable security unit should be explicitly addressed using one of the methods described in §4.2.2, e.g. using a unique Space Packet APID for carrying all Command and Reply PDUs to a given security unit.

The Frame Security Report is not used for downlink not addressed to the Mission Operation Center.

### Inter-satellite linkS

It may be desirable to employ Extended Procedures PDUs over some space-to-space links. In these scenarios, a spacecraft could be used as a proxy or relay node for managing SDLS links between itself and other spacecraft. This would facilitate management of endpoints where ‘classical’ ground-to-space telecommand is rendered difficult by frequent obstructions to direct RF communications.

#### Scenario 1: ground as master, with a hierarchy of inter-satellite control

Since the SDLS Extended Procedures operate as a master-slave protocol (with no negotiation as would be common in a peer-to-peer protocol), it is necessary for a mission using SDLS EP over space-to-space links to predetermine the relationship between nodes regarding how EP PDUs are to be exchanged.

In this scenario, a full hierarchy is defined across the constellation where for each possible inter-satellite link there is a master and a slave. The mission operations center is a ‘master of masters’, controlling one or more intermediate slave nodes using EP directives.

These intermediate slave nodes then, according to the predetermined hierarchy, autonomously control other nodes via space-to-space links to carry out configuration changes uploaded by the MOC for application to distant nodes, and/or propagate updates applied by the MOC to the intermediate slave configuration. Success or failure notification for these autonomous control operations would be provided to the MOC in telemetry from the intermediate slave nodes. ***Figure 4‑3*** depicts such a hierarchical relationship. If used for managing SDLS Security Associations, both directions of the space-to-space link (forward and return, from the perspective of the intermediate node) are managed by the intermediate node.

The primary difficulty is that hierarchical EP exchanges necessarily provides the ground lesser situational awareness and real-time control over distant individual nodes than it would enjoy with direct ground-space control. However, as noted earlier, the rationale for attempting this scenario would most likely be the mission’s expected difficulty of carrying out direct ground-space control. If a mission attempts to support *both* hierarchical control by intermediate nodes *and* direct ground-space control of distant nodes whenever possible, there needs to be a method to prevent conflicting or overlapping control directives. Autonomous management of SAs between intermediate and distant nodes appears to be particularly challenging to implement such that operational control is not lost in the absence of ground intervention.

The potential benefit is that it would provide a mechanism suited to bulk propagation of configuration data intended for all nodes in a constellation, particularly for updates which do not need to be completed in real time. For example, globally applicable Key Management directives (e.g. OTAR) could be propagated to a sizable number of distant nodes autonomously, according to communications availability.



*Figure 4‑3. Inter-satellite propagation using a hierarchy*

#### Scenario 2: ground as master, with inter-satellite relay of EP directives

As in the ‘classical’ ground-space scenario, a mission operations center acts as master to control one or more slave nodes using EP directives. But in this scenario, certain nodes are used to relay EP directives to other nodes via space-to-space links.

Because Extended Procedures PDUs are transferred using Space Packets, it is possible to uniquely assign and identify each actual destination node using the Application Process ID (APID). The intermediate relay would use the APID to recognize which received EP PDUs pertain to itself, and which PDUs pertain to other destination nodes. EP Request PDUs received from the mission operations center with a ‘foreign’ APID would be forwarded to the actual destination node being managed, and likewise EP Reply PDUs received from the managed destination nodes would be forwarded back to the mission operations center. ***Figure 4‑4*** illustrates such a scenario. (This is a Space Packet-level forwarding scheme analogous to how physical-layer data relay satellites act as repeaters with little local processing of the data being relayed.)

In this scenario, all nodes are essentially peers to one another whether acting as a relay or as a destination. Each node would neither locally process, nor retain a history of, PDUs pertaining to other nodes. Because of this peer-to-peer relationship, the space-to-space links should use bi-directional protocols (i.e. AOS or USLP).

An inter-satellite relay scheme for EP PDUs necessitates a unique APID assignment for each possible destination node’s SDLS EP interface. It would not require that every spacecraft’s local APIDs must be unique across the entire constellation of participating nodes. It would indicate a requirement that the APIDs assigned to SDLS EP directive interfaces must be considered ‘non-local’ and therefore must be globally unique to avoid confusion. The partitioning of APIDs into local versus non-local assignments for a constellation of multiple spacecraft is mission-specific.

Adding to the complexity, the mission operations center would have to maintain knowledge of which potential relay nodes at any particular time have available space-to-space links for contacting a particular destination node. It is not expected that relay nodes would act autonomously to establish RF communications with a peer node.



*Figure 4‑4. Inter-satellite relay of EP directives*

## Relationship to other CCSDS STandards

### Cryptographic Algorithms (352.0-B)

SDLS was designed to be compatible with a variety of algorithms. Neither the base SDLS protocol nor the SDLS Extended Procedures mandate the use of a specific cryptographic algorithm. The (non-normative) baseline modes specified in [1] and [2] for interoperability testing use the AES-GCM algorithm with 256-bit keys, 96-bit IV, and 128-bit MAC. In the case of TC Space Link Protocol, the baseline modes use the AES-CMAC algorithm with 256-bit keys and 128-bit MAC.

#### Implications for SA creation

1. ARSN: Where AES-GCM is used (as in the baseline mode for TM, AOS, and USLP), SDLS uses the IV and ARSN as a single field. The Create SA procedure initializes the length and initial values of the anti-replay sequence number (ARSN), which also serves a double function as the initialization vector (IV).
2. IV: As an AES-GCM IV, it comprises a ‘fixed’ field (a value static to the originating device/context) and an ‘invocation’ field (a value different with every invocation). This field therefore limits both the number of distinct devices/contexts that can call the GCM authenticated encryption function with a single key, and the number of times each one can call it. For example, a 32-bit fixed field implies a limit of 2*32* on the number of distinct devices/contexts; a 64-bit invocation field implies a limit of 2*64* on the number of invocations of the GCM authenticated encryption function. See reference [15] for more detail.
3. MAC: The length of the authentication tag constrains the safe number of operations over the lifetime of the key. Up to half the length of the MAC is thought to be a reasonable limit; in other words, a 128-bit MAC would provide authentication assurance for up to 2*64* frames. However, message size provides an additional constraint as the MAC length is shortened. Most supported CCSDS transfer frame sizes qualify as ‘short’ messages for AES-GCM algorithm considerations. [15] states that where the MAC is 64 bits long, and the maximum combined length of ciphertext and AAD in a message is 2*15* bytes, the maximum invocations of the authenticated decryption function should not be greater than 2*32*.

#### Implications for key verification

For each Key ID passed by the Key Verification procedure, a plaintext challenge is provided in the Request PDU, and the same challenge is returned as ciphertext in the Reply PDU along with an Initialization Vector (IV) and a Message Authentication Code (MAC). Because of this, the algorithm used must be resistant to known-plaintext attacks. AES-GCM has no known vulnerability to known-plaintext attacks, as long as the rule against ever repeating the use of an IV with the same key is followed. This rule must be obeyed, even within a single Reply PDU. The challenge should also be a random pattern to increase the difficulty of this type of attack.

### Symmetric Key Management (354.0-M)

#### Key Management schemes not implemented

There is a third key management scheme listed in [8] but not directly supported by the SDLS Extended Procedures:

* Scheme 3: a subset of keys (master keys/KEKs and session keys) are pre-loaded on satellite before launch; session keys are generated on-board from master keys and an uploaded non-secret seed.

The complexity of this scheme necessitates autonomous procedures operating beyond the scope of the SDLS Extended Procedures. It is heavily dependent upon a subset of cryptographic operations, including random number generation and key derivation algorithms, for which no CCSDS recommendation currently exists. As such, any implementation of this scheme would be mission-specific. For additional procedures to support such a scheme, the reader is referred to [8], section 4.3.7.

#### Key states not implemented

The SDLS Extended Procedures do not implement the full range of key states described in recommendation [8].

First, there is an optional Suspended state in [8] which anticipates a temporary operational restriction on the use of previously activated keys. One such use case would be for setting aside a set of still-unused keys which had been activated in the expectation of their being used in the near term, and which (for whatever reason) are no longer anticipated to be needed soon. This key state makes sense only for key management systems capable of storing a very large quantity of keys. Since the size constraints of on-board key storage typically preclude storing very many keys at any one time, use of the Suspended state in space systems is not anticipated. Even if ground-based key management systems supported the Suspended state, its implementation on-board would be superfluous and no SDLS Extended Procedures are provided to support it.

Second, the Compromised state in [8] prevents the operational use of keys which are unfit due to their having been disclosed. In the SDLS Extended Procedures, the Compromised state is listed as a state applicable only to the Initiator. Ground-based key management systems will often preserve Compromised keys in storage for record-keeping. The use case is not applicable to space systems, so no SDLS Extended Procedures are provided to support it. In the event the Initiator (master) needs to transition keys stored locally into the Compromised state, it would issue the Key Destruction directive to the Recipient (slave) to destroy the same keys.

## ???

Notes from MoM:

* §3.3.5 and 3.6.2: Handling redundancy N/R security unit and routing of EP PDUs on-board:
  + SPI space (i.e. SAs) could be portioned between Nominal and Redundant strings to guarantee uniqueness across strings, which is a necessity since SA states are a priori not shared across strings.
  + A discussion of the possible hardware architectures where security unit is implemented in series vs. implemented as a branch, should be added in §3.6 Various types of implementation.

1. BASELINE MODES
   1. **Introduction**

This annex provides the rationale for the baseline implementation mode specified in Annex D of the SDLS Extended Procedures Blue Book, reference [2].

* 1. Frame Security Report

The Frame Security Report (FSR) is the protocol data unit transmitted from the Recipient to the Initiator of an SDLS secured TC uplink. It provides 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.

The baseline implementation mode specified for integrating the FSR into the TM, AOS and USLP transfer service is as follows:

1. The FSR is reported as Operational Control Field (OCF Type 2).
2. In case COP-1 is reporting on the same virtual channel, the FSR reporting alternates with the Command Link Control Word (OCF Type 1) reporting.

The purpose of the Operational Control Field is to provide a standardized mechanism for reporting a small number of real-time functions such as supporting the reporting mechanism for the on-board SDLS security function. Two types of OCF have been specified in TM, AOS and USLP Space Data Link Protocol:

* Type 1 for the Command Link Control Word (CLCW) of the COP-1 retransmission protocol
* Type 2 for the Frame Security Report (FSR) of the SDLS security protocol

Both reporting mechanisms are usually needed on TC or USLP uplinks. Therefore, OCF needs to be shared between COP-1 and SDLS reporting as specified in the baseline mode. Ideally, a CLCW and a FSR should be transmitted to ground for each received uplink TC or USLP transfer frame. If the downlink frame rate is at least twice the uplink frame rate, this is feasible by interleaving the two types of reports in the OCF of the TM, AOS or USLP frames. If this condition is not met, subsampling of CLCW, FSR or both will need to be done. In that case, all Security Event Flags in the FSR being non persistent, the initiator (Mission Control Center) will not be able to relate a given security event to a specific frame, directly from the FSR analysis. The occurrence of a Security Event will be signaled to the initiator by the Alarm Flag which is a persistent flag in the FSR. The initiator will then have to investigate through analysis of the recipient (on-board Security Unit) telemetry, to determine which frame has triggered the security event.

* 1. Protocol Data Units (PDU)

SDLS 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.

The TLV concept allows nesting: the Value field can itself be composed of one or more TLV messages. Given the procedures selected for the baseline mode, there is no need for nested TLV PDUs.

* 1. Reserved SPI/SA

Sensitive EP Service PDUs need to be communicated over a SDLS channel protected by authenticated encryption to guarantee integrity, authenticity and confidentiality. All other EP Service PDUs need to be at least authenticated to guarantee integrity and authenticity before execution or processing.

In the baseline implementation mode, the two SDLS reserved SPIs (values of ‘all zeros’ (0) and ‘all ones’ (65535)) defined in reference [1] are used for exchanging EP Service PDUs. This allows for the use of dedicated Security Associations (SAs) to protect the transmission of EP Service PDUs, and therefore the use of different SA from the one being affected by the EP Service Command PDU preventing unintentional loss of control of an SA.

* 1. Key Management Service
     1. General

The baseline implementation mode includes all EP key management procedures except Key destruction and Key inventory:

* Key destruction is not needed in most mission scenarios: Key revocation at both end of the Secure Channel by the Key Deactivation procedure is sufficient to guarantee that a compromised key or “burnt” key (all IV or ARSN used) cannot be used for cryptographic operations anymore. A deactivated key can only be used to decrypt previously encrypted data.
* Key inventory is not absolutely necessary to manage the on-board set of keys. On-board key states can be inferred from command execution verification reports of other baseline mode directives.
  + 1. Security algorithm and key configuration
       1. Selection of cryptographic algorithm for OTAR and key verification

The cryptographic algorithm is selected from the CCSDS Standard on Cryptographic Algorithms [12], in particular from the recommended algorithms for Authenticated Encryption. Therefore, the AES-GCM is the recommended algorithm for the OTAR and key verification operations which requires authentication and encryption of the uploaded keys.

Recent cryptographic research on AES-GCM has identified a weakness concerning certain keys [16,17]. The user is invited to carefully consider the key generation and selection process in order to avoid the use of ‘weak’ keys.

* + - 1. Design of Cryptographic Algorithm Parameters: MAC and key lengths

With the selection of AES-GCM, the selection of MAC and key length is as follows:

* The MAC length is automatically set to 128 bits, which is the maximum possible value. This value is considered sufficiently secure for civilian missions as justified by the security analysis in A2.3 of reference [10].
* The key length is limited to three possible values: 128, 192 and 256 bits. A value of 128 bits is considered sufficient for civilian missions as justified by the security analysis in A2.3 of reference [10] but a margin on key length is necessary to anticipate the threat of quantum computers. This leads to a selection of a 256-bit key for the SDLS Extended Procedures.
  + - 1. IV Construction

AES-GCM requires an Initialization Vector. There are too specified approaches to construct an IV for AES-GCM (see section 8.2 of [15]). The recommended construction is the following: deterministic with 96 bits in total length.

To maintain security, it is essential to avoid a repetition of the IV with the same cryptographic key. Failure to meet this requirement will imply a security leakage. Further details can found in [15].

* + 1. Key management services parameters
       1. OTAR

The format of the OTAR command PDU is shown in Figure A-1



Figure A-1: OTAR Command PDU

The baseline implementation configuration selected for OTAR procedure operation is:

1. The Master Key ID field of the OTAR Command PDU shall have a size of 16 bit.  
     
   It is up to the implementer to decide if master keys are assigned a special range from the total key ID range. A 16 bit ID for Master Key & session keys allows for 65.536 keys in total which is largely sufficient for most missions.
2. The Initialization Vector field of the OTAR Command PDU shall have a size of 96 bit.

This size of 96 bit derives from the choice of AES-GCM as cryptographic algorithm for OTAR. See justification in A5.2.3.

1. Each Encrypted Key Block of the OTAR Command PDU shall have a size of 272 bit, consisting of
   * The Key ID fields shall have a size of 16 bit,
   * The Session Key fields shall have a size of 256 bit,

The size of Key ID and Session Key fields (respectively 16 and 256) derives from the settings of the SDLS baseline mode defined in Annex E of reference [1] and justified in Annex A of [10].

1. The MAC field of the OTAR Command PDU shall have a size of 128 bit.

This size of 128 bit for the MAC derives from the choice of AES-GCM as cryptographic algorithm for OTAR. See justification in A5.2.2.

1. In baseline mode, OTAR Command PDU allows for the transfer of up to 16 session keys (N<=16). This limitation while acceptable operationally, allows for the complete Command PDU to fit into one TC frame.
   * + 1. Key Activation

The format of the Key Activation command PDU is shown in Figure A-2



Figure A-2: Key Activation Command PDU

1. In baseline mode, Key ID length is 16 bit which allows for 65.536 keys in total (Session Keys + Master Keys) which is largely sufficient for most missions, especially with the possibility to upload new keys in flight with OTAR procedure.
2. Baseline mode allows for up to 32 keys to be activated with a single Command PDU (N<=32). This limit is coherent with OTAR procedure and acceptable operationally.
   * + 1. Key Deactivation

Same parameters as Key Activation command PDU. See A5.3.2

* + - 1. Key Verification

The format of the Key Verification command PDU is shown in Figure A-3

SDLS Extended Procedures Red 1v5 CTB

Figure A-3: Key Verification Command PDU

1. In baseline mode, Key ID length is 16 bit which allows for 65.536 keys in total (Session Keys + Master Keys) which is largely sufficient for most missions, especially with the possibility to upload new keys in flight with OTAR procedure.
2. Challenge size is 128 bits which is coherent with the algorithm selected for key verification which is AES-GCM operating on 128-bit blocks. Challenge should be a pure random pattern to avoid clear-cipher text attacks, the potential attacker being able to intercept both the challenge (clear text) in the command PDU and the encrypted challenge (in the reply PDU), even though AES-GCM has no known vulnerability to clear-cipher text attacks.
3. Baseline mode allows for up to 32 keys to be verified with a single Command PDU (N<=32). This limit is coherent with OTAR procedure and acceptable operationally.

The format of the Key Verification reply PDU is shown in Figure A-4

SDLS Extended Procedures Red 1v5 CTB

Figure A-4: Key Verification Reply PDU

1. In baseline mode, Key ID length is 16 bit which allows for 65.536 keys in total (Session Keys + Master Keys) which is largely sufficient for most missions, especially with the possibility to upload new keys in flight with OTAR procedure.
2. The Initialization Vector field of the Key Verification Reply PDU shall have a size of 96 bit.

This size of 96 bit derives from the choice of AES-GCM as cryptographic algorithm for key verification. See justification in A5.2.3.

1. Encrypted challenge size is 128 bits which is coherent with the algorithm selected for key verification which is AES-GCM operating on 128-bit blocks. The encrypted challenge enables the initiator to check the integrity of each of the keys loaded through the OTAR procedure. AES-GCM has no known vulnerability to clear & cipher text attacks at this current time. Therefore, it is acceptable to transmit unencrypted between initiator and recipient both challenge and encrypted challenge.
2. MAC is 128-bit coherent with AES-GCM.
3. Baseline mode allows for up to 32 keys to be verified with a single Command PDU (N<=32).
4. Before verifying a key, this key needs to be activated which starts its operational lifetime.
   * 1. Security association management service parameters

Baseline mode of extended procedures retains six SA Management procedures : Start SA, Stop SA, Rekey SA, Expire SA, Set ARSN, Read ARSN. Create SA, Delete SA, Set ARSN Window and SA Status Request are not selected for baseline mode for the following reasons:

* In most missions, there is no need to create or to delete an SA in flight. All SAs needed for the mission duration are preloaded on-board. Up to 65.536 SAs can be loaded onboard before launch which is largely sufficient to cover the lifetime.
* The ARSN Window can be selected statically for the mission. Most missions will select a window of maximum size allowing any up counting ARSN. This protects against replay while allowing for any type of gaps in the reception of frames at the recipient.
* SA Status Request: SA status can in most cases be managed from the ground.

Note on ARSN length: The Rekey SA and Set ARSN procedures specified in the baseline mode allocate 96 bits to be able to carry an ARSN for any of the supported Space Link Protocols.

1. As used in the baseline mode for TM, AOS, and USLP, the ARSN is 96 bits in length. Since the ARSN is identical to the IV for the baseline mode AES-GCM algorithm, executing this procedure also sets the IV.
2. As used in the baseline mode for TC, the ARSN is 32 bits in length. If this ARSN field carries an ARSN for TC SAs, the left-most 64 bits are zeroed.
   * + 1. Start SA

The format of the Start SA command PDU is shown in Figure A-5



Figure A-5: Start SA Command PDU

The baseline implementation configuration selected for Start SA procedure operation is:

1. SPI field is 16 bits in length.  
     
   It allows for 65.536 security associations in total which is largely sufficient for most missions.
2. Each GVCID/GMAPID can fit into a 32 bit length field. Each GVCID/GMAPID listed is associated with the SA identified by the SPI. GVCID and GMAPID are specified in the relevant Space Data Link Protocol (SDLP) recommendation [4][5][6][7] for the space data link protocol used on the space link associated with the SPI. The GVCID is the concatenation of: GVCID = TFVN + SCID + VCID. GMAPID is the concatenation of: GMAPID = TFVN + SCID + VCID + MAPID. In all cases (whichever the SDLP used), GVCID and GMAPID can each be coded on less than 32 bits, the leftmost bits being filled with “0” to complete the 32-bit field.
   * + 1. Rekey SA

The format of the Rekey SA command PDU is shown in Figure A-6



Figure A-6: Start SA Command PDU

The ARSN field length needs to be 96 bits to be able to carry either ARSN (TC SAs) or IV (TM, AOS, or USLP SAs). If this ARSN field carries an ARSN (TC SAs) and not an IV, the left-most 64 bits are zeroed.

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

The cryptographic algorithms selected for the baseline mode are:

* AES-GCM for TM, AOS and USLP
* AES-CMAC for TC

Therefore, in all cases, only one single key and key ID is required for operation in baseline mode whichever the Space Data Link Protocol used on the secured link.

* + - 1. Set ARSN, Read ARSN

The format of the Set ARSN command and Read ARSN reply PDUs are shown in Figure A-7





Figure A-7 Set ARSN Command and Read ARSN reply PDUs

The ARSN field length needs to be 96 bits to be able to carry either ARSN (TC SAs) or IV (TM, AOS, or USLP SAs). If this ARSN field carries an ARSN (TC SAs) and not an IV, the left-most 64 bits are zeroed.

Since the ARSN is identical to the IV for the SDLS baseline mode AES-GCM algorithm, executing the Set ARSN procedure will set the IV.

* + 1. Monitoring and control service parameters

The baseline implementation mode of Extended Procedures includes the Ping and Alarm Flag Reset procedures.

*Not* selected for the baseline mode are:

1. The Log Status, Dump Log, and Erase Log procedures: in most missions, an on-board Security Log is not needed. The Frame Security Report (FSR) provides enough observability to record on the ground all security events, provided that FSR is sampled at each received TC frame.
2. The Self-Test procedure: Self-test is usually implemented in the security unit in a mission-specific way for which interoperability is not needed.

1. Acronyms and Abbreviations

This annex lists the acronyms and abbreviations used in this Report.

AAD Additional Authenticated Data

AEAD Authenticated Encryption with Associated Data

AES Advanced Encryption Standard

AOS Advanced Orbiting Systems

CCSDS Consultative Committee for Space Data Systems

COP-1 Communications Operation Procedure-1

CRC Cyclic Redundancy Check

CTR Counter Mode

FSP Forward Space Packet

HMAC Hash-based Message Authentication Code

HPC High Priority Command

IEC International Electrotechnical Commission

IP Internet Protocol

ISO International Standards Organization

IV Initialization Vector

KEK Key Encryption Key

MAC Message Authentication Code

MAP Multiplexer Access Point

MC Master Channel

MCID Master Channel Identifier

N/A Not Applicable

OCF Operational Control Field

OID Only Idle Data

OSI Open Systems Interconnection

OTAR Over-the-air Rekeying

PDU Protocol Data Unit

PVN Packet Version Number

RDBMS Relational Database Management System

RS Reed-Solomon

SA Security Association

SCID Spacecraft Identifier

SDL Space Data Link

SDLS Space Data Link Security Protocol

SDU Service Data Unit

SLP Space Link Protocol

SPI Security Parameter Index

TC Telecommand

TM Telemetry

URD User Requirements Document

USLP Unified Space Link Protocol

VC Virtual Channel

VCA Virtual Channel Access

VCID Virtual Channel Identifier