

|  |
| --- |
| Space Data Link Security (SDLS) Core protocol interoperabilityTest report |

CCSDS Record

CCSDS 355.1-Y-1

Yellow Book

March 2015

FOREWORD

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 *Procedures Manual for the Consultative Committee for Space Data Systems*. 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** |
| CCSDS 355.1-Y-1 | CCSDS Space Data Link Security core protocol interoperability test Report, CCSDS Record | March 2015 | Current issue |
|  |  |  |  |
|  |  |  |  |

CONTENTS

Section Page

[DOCUMENT CONTROL iii](#_Toc414467069)

[CONTENTS iv](#_Toc414467070)

[1 Introduction 1-1](#_Toc414467071)

[1.1 purpose 1-1](#_Toc414467072)

[1.2 Scope 1-1](#_Toc414467073)

[1.3 applicability 1-1](#_Toc414467074)

[1.4 rationale 1-1](#_Toc414467075)

[1.5 document structure 1-1](#_Toc414467076)

[1.6 References 1-2](#_Toc414467077)

[2 Acronyms 2-3](#_Toc414467078)

[3 Overview 3-4](#_Toc414467079)

[4 SDLS Protocol testing OBJECTIVEs 4-6](#_Toc414467080)

[5 test CASES 5-8](#_Toc414467081)

[5.1 TEST CASE #1 : SDLS protocol validation over TC link 5-10](#_Toc414467082)

[5.1.1 test description 5-10](#_Toc414467083)

[5.1.2 expected results 5-10](#_Toc414467084)

[5.1.3 Inter-operability tests effective results 5-10](#_Toc414467085)

[5.2 TEST CASE #2 : SDLS protocol / Cop-1 compatibility check 5-11](#_Toc414467086)

[5.2.1 test description 5-11](#_Toc414467087)

[5.2.2 expected results 5-12](#_Toc414467088)

[5.2.3 Inter-operability tests effective results 5-12](#_Toc414467089)

[5.3 TEST CASE #3 : SDLS protocol validation over TM link 5-13](#_Toc414467090)

[5.3.1 test description 5-13](#_Toc414467091)

[5.3.2 expected results 5-13](#_Toc414467092)

[5.3.3 Inter-operability tests effective results 5-14](#_Toc414467093)

[5.4 TEST CASE #4 : SDLS protocol validation over AOS link 5-14](#_Toc414467094)

[5.4.1 test description 5-14](#_Toc414467095)

[5.4.2 Expected results 5-15](#_Toc414467096)

[5.4.3 Inter-operability tests effective results 5-15](#_Toc414467097)

[6 CONCLUSION 6-16](#_Toc414467098)

**ANNEX A: SDLS PROTOCOL VALIDATION / TEST REPORT**

[1 Introduction A-4](#_Toc414467099)

[1.1 Project Description: A-4](#_Toc414467100)

[1.2 Bibliography A-4](#_Toc414467101)

[2 Test Setup A-5](#_Toc414467102)

[3 Test Case #1 A-8](#_Toc414467103)

[3.1 Test Configuration A-8](#_Toc414467104)

[3.2 Test Process A-9](#_Toc414467105)

[3.3 Test Results A-11](#_Toc414467106)

[4 Test Case #2 A-12](#_Toc414467107)

[4.1 Test Configuration A-12](#_Toc414467108)

[4.2 Test Process A-13](#_Toc414467109)

[4.3 Test Results A-15](#_Toc414467110)

[5 Test Case #3 A-17](#_Toc414467111)

[5.1 Test Configuration A-17](#_Toc414467112)

[5.2 Test Process A-18](#_Toc414467113)

[5.3 Test Results A-19](#_Toc414467114)

[6 Test Case #4 A-22](#_Toc414467115)

[6.1 Test Configuration A-22](#_Toc414467116)

[6.2 Test Process A-23](#_Toc414467117)

[6.3 Test Results A-24](#_Toc414467118)

[7 Conclusion A-37](#_Toc414467119)

[8 Acronyms A-38](#_Toc414467120)

Table Page

# Introduction

## purpose

The purpose of this document is to describe the interoperability tests conducted for the validation of the CCSDS Space Data Link Security (SDLS) core protocol specified in CCSDS 355.0-B-1 (reference [1]). The objective of this interoperability testing is to demonstrate that at least 2 independent implementations of the SDLS core protocol recommendation interoperate.

## Scope

The scope of this document is to specify the test objectives, test cases and test results of interoperability testing of the CCSDS SDLS core protocol which provides authentication, integrity, anti-replay and confidentiality to TC, TM and AOS data links. The complete interoperability testing of CCSDS recommended cryptographic algorithms (e.g. AES in various modes) (reference [3]) which are embedded in SDLS protocol, is covered by the CCSDS Cryptographic algorithms test report (reference [4]).

## applicability

This interoperability test plan has been used to validate the interoperability of at least 2 independently developed implementation of the SDLS protocol. It can be further used by any user of the recommendation to test its implementation against reference implementations that could be made available later by CCSDS for conformance testing.

## rationale

The CCSDS Procedures Manual states that for a draft Recommendation to become a Blue Book, the standard must be tested in an operational manner. The following requirement for an implementation exercise were excerpted from reference [2]:

“At least two independent and interoperable prototypes or implementations must have been developed and demonstrated in an operationally relevant environment, either real or simulated.”

This document outlines the Space Data Link Security Working Group’s approach to meeting this requirement.

## document structure

This document describes the testing that must be accomplished to allow the CCSDS Space Data Link Security (SDLS) protocol to proceed forward as a Recommended Standard.

The document is split in 3 parts:

* Test objectives
* Test cases
* Test results

## References

The following documents are referenced in this document. At the time of publication, the editions indicated were valid. All documents are subject to revision, and users of this document 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. *CCSDS Space Data Link Security (SDLS) protocol*. CCSDS 355.0-R-4. Red Book. Issue 4, April 2014
2. *Procedures Manual for the Consultative Committee for Space Data Systems*, CCSDS A00.0-Y-9. Yellow Book. Issue 9. Washington DC: CCSDS, November 2003.
3. *CCSDS Cryptographic Algorithms*. CCSDS 352.0-B-1. Blue Book. Issue 1. Washington DC: CCSDS, November 2012
4. *CCSDS Cryptographic Algorithms test report*. CCSDS 352.1-Y-1. Yellow Book. Issue 1. Washington DC: CCSDS, November 2012
5. *TC Space Data Link Protocol*. CCSDS 232.0-B-2. Blue Book. Issue 2. Washington DC: CCSDS, September 2010
6. *TM Space Data Link Protocol*. CCSDS 132.0-B-1. Blue Book. Issue 1. Washington DC: CCSDS, September 2003
7. *AOS Space Data Link Protocol*. CCSDS 732.0-B-2. Blue Book. Issue 2. Washington DC: CCSDS, July 2006
8. *TC Space Data Link Protocol*. CCSDS 232.0-P-2.1. Pink sheets. September 2013
9. *TM Space Data Link Protocol*. CCSDS 132.0-P-1.2. Pink sheets. October 2013
10. *AOS Space Data Link Protocol*. CCSDS 732.0-P-2.2a. Pink sheets. October 2013

# Acronyms

AES-GCM Advanced Encryption Standard – Galois Counter Mode

CCSDS Consultative Committee for Space Data Systems

IV Initialization Vector

MAC Message Authentication Code

NIS Network Interface System

SA Security Association

SCOS Spacecraft Control and Operations System

SDLS Protocol Space Data Link Security Protocol

SN Sequence Number

SPI Security Parameter Index

TC Telecommand

TM Telemetry

TMTCS Telemetry & Telecommand System

VC Virtual Channel

# Overview

This CCSDS Space Data Link Security Protocol test plan describes the manner in which SDLS protocol tests have been accomplished. It describes the manner in which the protocol is to be implemented, set (e.g. : keys and SAs), and data exchanged between the testing parties to determine if the protocol is performing as expected between 2 independent implementations of the SDLS recommendation.

The CCSDS Procedures Manual requires that testing be performed in an “operational-like” setting. This plan provides the details to test the SDLS protocol specification to ensure its completeness, correctness and interoperation. We propose that an independent SDLS implementation is used as the sending end of the data link and another independent implementation is used as the receiving end. This will be performed using all the recommended modes of operation (so called “baseline modes”) defined in annex E of SDLS recommendation (reference [1]) which covers TC, TM and AOS links.

Since SDLS protocol is fully embedded in TC, TM and AOS data link protocols (references [5], [6], [7] amended respectively by [8], [9], [10] integrating SDLS function), the SDLS implementations tested will include necessarily at least TC, TM and AOS protocol simulation.

One important objective of the testing is to validate that there is no interaction between SDLS protocol and TC/TM/AOS transmission error control procedures (e.g. TC COP-1). Therefore, transmission errors and security (intentional) errors must be injected on the physical link between the SDLS sending and receiving ends to check the non-interaction and complementarity of SDLS and data link protocols w.r.t. error handling.

This testing could be performed in a single laboratory by one test bed using multiple implementations of SDLS protocol. However, optimally the testing should be conducted at multiple sites via the internet potentially using something as simple as email to send SDLS secured transfer frames between the testing parties which would then be fed into the various independent algorithm implementations. For example, Test Agent A at site X could apply SDLS to a pre-defined set of transfer frames (TC, TM or AOS) using a pre-distributed set of Security Associations (SA). Test Agent A would send by email the SDLS secured frames as an attachment to Test Agent B at site Y. Using the pre-distributed set of SAs and a different implementation of SDLS protocol than used by Test Agent A, Test Agent B would attempt to validate received transfer frames. If the SDLS protocol receiving end is able to detect and handle correctly all errors (both security and transmission related), the test passes. This is the testing set up that has been used for the interoperability tests performed by ESA, CNES, NASA, whose results are reported in this CCSDS report.

In a more elaborate testing setup, the test sites could be interconnected and a simple network protocol (e.g. UDP/IP) could be used to provide the test framework of generating SDLS secured transfer frames, transmitting it, and on the receiving end, validating the transfer frames. This test setting however requires that 2 agencies establish an internet link between them to exchange TC/TM/AOS secured transfer frames in real-time. This has not been possible in the frame of the SDLS validation due to firewall restrictions in various agencies.

# SDLS Protocol testing OBJECTIVEs

SDLS protocol testing general objectives are the following:

* check completeness, correctness and non ambiguity of SDLS protocol specification for :
  + the 3 types of space data links supported by SDLS : TC, TM, AOS
  + the 3 types of security services that can be provided by SDLS : authentication (authentication, integrity, anti-replay), encryption (confidentiality), authenticated encryption.
  + the 3 cryptographic algorithms/modes recommended in the baseline implementation modes of SDLS (annex E of reference [1]): AES/GCM (TM), AES/CMAC (TC), AES/GCM (AOS)
  + The various types of errors that can be encountered on the link : transmission errors, security intentional errors
* Check interoperability of at least 2 independent implementations of SDLS protocol for all settings listed above
* Check absence of interference between TC/TM/AOS protocols and SDLS protocol, in particular between TC COP and SDLS.

More specifically, the detailed test coverage targeted is the following:

* Check all SDLS protocol procedures as defined in the standard
* Check protection (authentication and/or encryption) of all concerned TC, TM and AOS communication services as defined in relevant TC, TM and AOS SDLP recommendations and identified as services protected by SDLS protocol
* Check all parameters as defined in SDLS protocol (in interoperability testing performed : limited to baseline modes for TC, TM and AOS, recommended for normal operations)
* Check all SDLS functions & security services : SA management, Authentication, Encryption, Authenticated Encryption
* Check for each security service at least one algorithm (i.e. mode of operation & cryptographic primitives) among algorithms recommended by CCSDS (as defined in baseline implementation modes annex E of SDLS protocol recommendation)
* Check all SDLS defined data units
* Validate co-existence of both secure and unsecure (clear) Virtual Channels (TC & TM) / MAP channels (TC only) over a master channel
* Check compatibility of SDLS protocol with relevant TC, TM and AOS space data link protocols as defined by CCSDS
* Check SDLS protocol compatibility with COP-1 procedure over TC space data link
* Check both nominal cases as non nominal / error cases
* Validate SDLS protocol in a fully representative end to end TC and TM link
  + allowing full separation / independence of ground & satellite end users
  + allowing to simulate /configure variable transmission errors (single errors / burst errors)

# test CASES

The following validation steps are performed in sequence first with a single implementation providing both sending and receiving ends, then with 2 independent implementations providing the ground part on one side and the on-board part on the other side:

* SDLS protocol validation over TC link
* SDLS protocol / COP-1 compatibility check
* SDLS protocol validation over TM link
* SDLS protocol validation over AOS link

Transfer of TC/TM/AOS) frames between 2 independent implementations (ground simulator / space simulator) can be done through file exchange or UDP/IP local or remote connection. File exchange by email was used for its simplicity.

For each test case, this document provides:

* Test case description & parameters
* Expected results
* Effective results obtained during the inter-agency interoperability testing.

The detailed test configurations, settings and results are captured in the ESA-CNES-NASA SDLS interoperability test report – see annex A.

Inter-agency interoperability tests were performed in the following configurations:

* Test cases #1, 2 :
  + ESA simulating sending end (ground part) of the SDLS protocol over TC
  + CNES simulating receiving end (on-board part) of the SDLS protocol over TC
* Test cases #3 :
  + CNES simulating sending end (on-board part) of the SDLS protocol over TM
  + ESA simulating receiving end (ground part) of the SDLS protocol over TM
* Test cases #4:
  + NASA simulating sending end (on-board part) of the SDLS protocol over AOS
  + CNES simulating receiving end (ground part) of the SDLS protocol over AOS

The interoperability tests configurations are depicted in figure 1 of Annex A and reproduced hereafter.

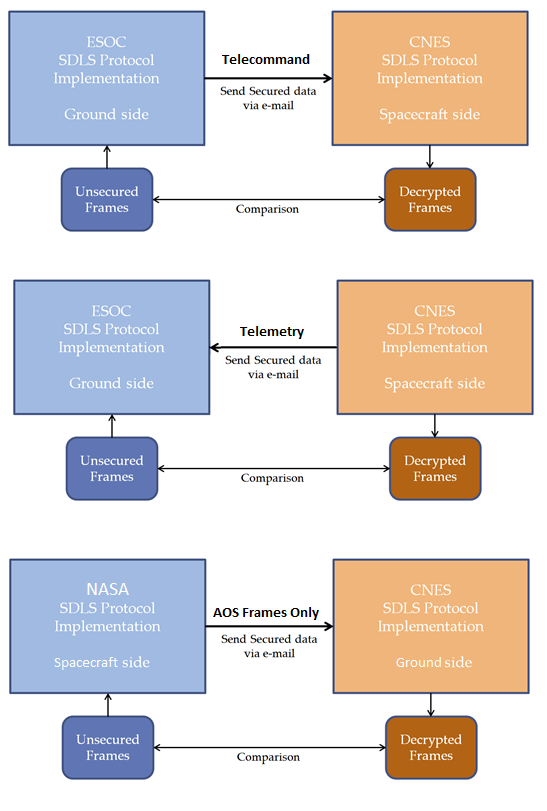


Figure 1 – Interoperability tests configurations

## TEST CASE #1 : SDLS protocol validation over TC link

### test description

SDLS protocol over TC link has been validated in the baseline mode for TC - defined in annex E of SDLS recommendation [1].

Test configuration:

* Data type: CLTUs
* Segment header: present
* Only expedited service (no COP) : BD frames
* Security service type : authentication only
* Cryptographic algorithm : AES-CMAC
* Key : 128 bits
* Anti-replay sequence number : 32 bits transmitted entirely in Security Header
* Total Security Header length : 6 octets
* MAC : 128 bits
* Authentication bit mask : default bit mask as defined in SDLS recommendation §4.2.2.6.2
* Anti-replay sequence number window : 5
* Mix of correct and defective (security errors) CLTUs
* Mix of SDLS secured and unsecured (clear mode) VC and MAP

Test scenario:

* No errors
* Replay
* Anti-replay sequence number mismatch
* MAC error (random errors, forging of frame subfields)
* Invalid SPI
* Key mismatch

### expected results

Correct detection of exceptions and ground-space configuration mismatch.

### Inter-operability tests effective results

Detailed test configuration and results can be found in annex A.

The scenario “Key mismatch” was not simulated as it would simply resolve in a MAC error, which is covered by the test case “MAC error”.

The test was also restricted to one VC and one MAP: the multiplexing/demultiplexing of VCs and MAPs based on VCID and MAPID being part of the TC Space Data Link Protocol itself and therefore not in need of being tested for SDLS validation.

10 CLTUS were successfully exchanged with a mix of:

* “clean” CLTUs (no security errors)
* “corrupted” CLTUs (security errors inserted):
  + Replay
  + Invalid MAC
  + Sequence Number (SN) out of boundary
  + Invalid SPI

All CLTUs were correctly processed by the receiving side, all security errors were detected and duly classified.

## TEST CASE #2: SDLS protocol / Cop-1 compatibility check

### test description

SDLS protocol is in baseline mode for TC (as per annex E: authentication only + AES-CMAC). TC space data link protocol over TC master channel implements COP-1. Objective of this test case is to check the non-interference of the TC-COP protocol (in charge of detecting & retransmitting transmission errors) and the SDLS protocol (in charge of detecting security errors). To achieve that, a mix of transmission and security errors are injected in CLTUs.

Test configuration:

* Data type : CLTUs
* Segment header: present
* 1 single VC
* 1 single MAP per VC
* Secure VC and MAP
* Mix of AD frames and COP-1 control commands (BC frames) on TC link.
* TC security service type : authentication only
* Cryptographic algorithm : AES-CMAC
* Key : 128 bits
* Anti-replay sequence number : 32 bits transmitted integrally in Security Header
* Total Security Header length : 6 octets
* MAC : 128 bits
* Authentication bit mask : default bit mask as defined in SDLS recommendation §4.2.2.6.2
* Anti-replay sequence number window : 5
* Mix of correct and defective (security errors and transmission/FEC errors) CLTUs

Test scenario:

* No errors, no control commands
* Security exceptions/errors:
  + Replay
  + Anti-replay sequence number mismatch
  + MAC error (forging of frame subfields : incorrect MAC but correct FEC)
  + Invalid SPI
  + Key mismatch
* Transmission errors (incorrect FEC):
  + Single bit error
  + Burst errors (impacting multiple frames)
  + Errors on AD and BC frames
* COP-1 reconfiguration
  + Set V[R]

### expected results

No interaction between SDLS and COP-1:

* SDLS should detect and handle security errors/exceptions
* COP-1 should detect and handle transmission errors

### Inter-operability tests effective results

Detailed test configuration and results can be found in ref [8].

Only two security errors (Replay, MAC errors) were tested as that is sufficient to prove the independency of the TC-COP and the SDLS protocol. The other security exceptions were tested during test case #1

10 CLTUS were successfully exchanged with a mix of:

* “clean” CLTUs (no security errors)
* “corrupted” CLTUs (security errors inserted):
  + Replay
  + Invalid MAC
* “corrupted” CLTUs (transmission errors inserted):
  + BCH and frame CRC error
  + BCH error

All CLTUs were correctly processed by the receiving side, all security and transmission errors were detected and duly classified.

## TEST CASE #3: SDLS protocol validation over TM link

### test description

SDLS protocol is in baseline mode for TM (as per annex E : authenticated encryption + AES-GCM). TM channel includes a mix of secure (SDLS protected) VC and unsecure VC. No forward link necessary.

Test configuration:

* Data type: TM frames
* TM security service type : authenticated encryption
* Cryptographic algorithm : AES-GCM
* Key : 128 bits
* Anti-replay sequence number : not needed in AES-GCM
* Total Security Header length : 14 octets (SPI:2 octets, IV: 12 octets)
* MAC : 128 bits
* Authentication bit mask : default bit mask as defined in SDLS recommendation §4.2.2.6.2
* Anti-replay sequence number window : 5
* Multiple VC
* Mix of secure and unsecure VCs
* Mix of correct and defective (security errors) TM transfer frames

Test scenario:

* No error
* Security exceptions:
  + Replay
  + Anti-replay sequence number mismatch
  + MAC error (random errors, forging of frame subfields)
  + Invalid SPI
  + Key mismatch
  + IV mismatch

### expected results

Correct detection of exceptions and ground-space configuration mismatch.

### Inter-operability tests effective results

Detailed test configuration and results can be found in Annex A.

Only 4 security errors (SN/IV Replay, 2 MAC errors, invalid SPI/SA) were tested as that is sufficient to cover all types of security errors.

The test was also restricted to one VC: the multiplexing/demultiplexing of VCs based on VCID being part of the TM Space Data Link Protocol itself and therefore not in need of being tested for SDLS validation.

20 TM transfer frames were successfully exchanged with a mix of:

* “clean” frames (no security errors)
* “corrupted” frames (security errors inserted):
  + Invalid SN and IV (replay or out of SN window)
  + Invalid MAC
  + Invalid SPI and SA

All TM transfer frames were correctly processed by the receiving side, all security errors were detected and duly classified.

## TEST CASE #4 : SDLS protocol validation over AOS link

### test description

SDLS protocol is in baseline mode for AOS (as per annex E : authenticated encryption + AES-GCM). AOS channel includes a mix of secure (SDLS protected) VC and unsecure VC. No forward link necessary.

Test configuration:

* Data type: AOS transfer frames
* AOS security service type : authenticated encryption
* Cryptographic algorithm : AES-GCM
* Key : 128 bits
* Anti-replay sequence number : not needed in AES-GCM
* Total Security Header length : 14 octets (SPI:2 octets, IV: 12 octets)
* MAC : 128 bits
* Authentication bit mask : default bit mask as defined in SDLS recommendation §4.2.2.6.2
* Anti-replay sequence number window : TBD
* Multiple VC
* Mix of secure and unsecure VCs
* Mix of correct and defective (security errors) AOS transfer frames

Test scenario:

* No error
* Security exceptions:
  + Replay
  + Anti-replay sequence number mismatch
  + MAC error (random errors, forging of frame subfields)
  + Invalid SPI
  + Key mismatch
  + IV mismatch

### Expected results

Correct detection/handling of exceptions/errors and ground-space configuration mismatch.

### Inter-operability tests effective results

Detailed test configuration and results can be found in ref [8].

Only 5 security errors (2 SN/IV Replay, 2 MAC errors, invalid SPI/SA) were tested as that is sufficient to cover all types of security errors.

The test was also restricted to one VC: the multiplexing/demultiplexing of VCs based on VCID being part of the AOS Space Data Link Protocol itself and therefore not in need of being tested for SDLS validation.

24 TM transfer frames were successfully exchanged with a mix of:

* “clean” frames (no security errors)
* “corrupted” frames (security errors inserted):
  + Invalid SN and IV (replay or out of SN window)
  + Invalid MAC
  + Invalid SPI and SA

All AOS transfer frames were correctly processed by the receiving side, all security errors were detected and duly classified.

# CONCLUSION

Test cases #1 to 4 were processed as described, with the result that the inter-agency (ESA-CNES-NASA) interoperability tests were successful. The individual results of the test cases are displayed hereafter:

*Test Case #1:*

All test scenarios of test case #1 succeeded, as the spacecraft simulator detected every security error contained in the TC BD Transfer Frames.

*Test Case #2:*

All test scenarios of test case #2 were successful. No interaction between the COP commands and SDLS protected TC transfer frames was found.

*Test Case #3:*

All test scenarios of test case #3 succeeded, as the ground system detected every security error contained in the TM transfer frames.

*Test Case #4:*

All test scenarios of test case #4 succeeded, as the ground system detected every security error contained in the AOS transfer frames.

In summary, 3 independent implementations of SDLS core protocol were used to perform interoperability testing:

* ESA/ESOC implementation of:
  + the TC/SDLS sending end (ground part)
  + the receiving end (ground part) of the SDLS protocol over TM
* CNES implementation of:
  + the TC/SDLS receiving end (on-board part)
  + the sending end (on-board part) of the SDLS protocol over TM
  + the receiving end (ground part) of the SDLS protocol over AOS
* NASA implementation of:
  + the sending end (on-board part) of the SDLS protocol over AOS

All transfer frames exchanged were correctly decoded, demonstrating inter-operability in the baseline mode of the SDLS protocol. No specification error/ambiguity was found in the draft SDLS blue book as a result of these tests. The fact that TC BC-frames are not protected by SDLS is correctly specified in [1] (§5.2). Nevertheless, 2 implementations were initially erroneously protecting those BC-frames. As a return of experience, it was decided to insist on this fact (no protection for BC-frames) in the SDLS green book to avoid similar mistakes.

1. SDLS Protocol Validation / test report

Inter-Agency Testing

Bruno Saba (CNES), Daniel Fischer (ESA), Brandon Bailey (NASA)

Table of Contents

[1 Introduction A-4](#_Toc414009960)

[1.1 Project Description: A-4](#_Toc414009961)

[1.2 Bibliography A-4](#_Toc414009962)

[2 Test Setup A-5](#_Toc414009963)

[3 Test Case #1 A-8](#_Toc414009964)

[3.1 Test Configuration A-8](#_Toc414009965)

[3.2 Test Process A-9](#_Toc414009966)

[3.3 Test Results A-11](#_Toc414009967)

[4 Test Case #2 A-12](#_Toc414009968)

[4.1 Test Configuration A-12](#_Toc414009969)

[4.2 Test Process A-13](#_Toc414009970)

[4.3 Test Results A-15](#_Toc414009971)

[5 Test Case #3 A-17](#_Toc414009972)

[5.1 Test Configuration A-17](#_Toc414009973)

[5.2 Test Process A-18](#_Toc414009974)

[5.3 Test Results A-19](#_Toc414009975)

[6 Test Case #4 A-22](#_Toc414009976)

[6.1 Test Configuration A-22](#_Toc414009977)

[6.2 Test Process A-23](#_Toc414009978)

[6.3 Test Results A-24](#_Toc414009979)

[7 Conclusion A-37](#_Toc414009980)

[8 Acronyms 2-1](#_Toc414009981)

**Figures**

[Figure 1 - Test procedure for TC and AOS 4](#_Toc414009959)

**Tables**

[Table 1 - ESOC procedures for Test Case #1 7](#_Toc414009947)

[Table 2 - CNES procedures for Test Case #1 7](#_Toc414009948)

[Table 3 - Test Results of Test Case #1 9](#_Toc414009949)

[Table 4 - ESOC procedures for Test Case #2 11](#_Toc414009950)

[Table 5 - CNES procedures for Test Case #2 11](#_Toc414009951)

[Table 6 - Test Results of Test Case #2 13](#_Toc414009952)

[Table 7 - CNES procedures for Test Case #3 16](#_Toc414009953)

[Table 8 - ESOC procedures for Test Case #3 16](#_Toc414009954)

[Table 9 - Test Results of Test Case #3 17](#_Toc414009955)

[Table 10 - NASA procedures for Test Case #4 20](#_Toc414009956)

[Table 11 - CNES procedures for Test Case #4 20](#_Toc414009957)

[Table 12 - Test Results of Test Case #4 21](#_Toc414009958)

# Introduction

## Project Description:

The Space Data Link Security (SDLS) Protocol (CCSDS, Space Data Link Security Protocol (CCSDS 355.0-R-4) 2014) is a CCSDS (Consultative Committee for Space Data Systems) standard which describes how Telemetry (TM), Telecommand (TC) and Advanced Orbiting Systems (AOS) data can and should be secured on the space data link layer. Currently the standard is assigned as a “Red Book” meaning it is a draft standard. To finalize such a draft standard (and with that, become a “Blue Book”) it must be implemented, tested/validated and reviewed by participating Agencies. In case of the SDLS Protocol the review has succeeded already. An overview of all the recommended standards is listed on the CCSDS website (CCSDS, CCSDS Official Website n.d.).

This report describes the procedure of testing three individual SDLS Protocol implementations to validate its functionality and interoperability as described in the test cases of the SDLS Protocol Test Report[[1]](#footnote-1). One implementation is provided by ESA/ESOC, the second one is provided by CNES[[2]](#footnote-2), whereas NASA supplies the third one. All these implementations were designed and created independently.

## Bibliography

CCSDS. *CCSDS Official Website.* n.d. www.ccsds.org/.

CCSDS. “Space Data Link Security Protocol (CCSDS 355.0-R-4). April 2014.

CCSDS. “Space Data Link Security Protocol Test Report (CCSDS 355.1-Y-1)

Vega, Telespazio. “GSEC Demonstrator - Software User Manual (EGOS-GEN-GSEC-SUM-1001).” 2012.

# Test Setup

To verify the SDLS Protocol, it is desired to test two individual implementations with each other by generating frames on one and analysing them on the other. During TC and TM protocol testing, CNES simulates the spacecraft meaning that secured telemetry is created and incoming telecommands are decrypted. ESA/ESOC on the other side simulates the ground station, i.e. sends secured telecommands and decrypts incoming telemetry. During AOS testing, NASA simulates the spacecraft meaning that secured telemetry is created and sent to CNES for processing. In all cases, the data is transferred via e-mail. The following image shows an example for the telecommand and AOS process.

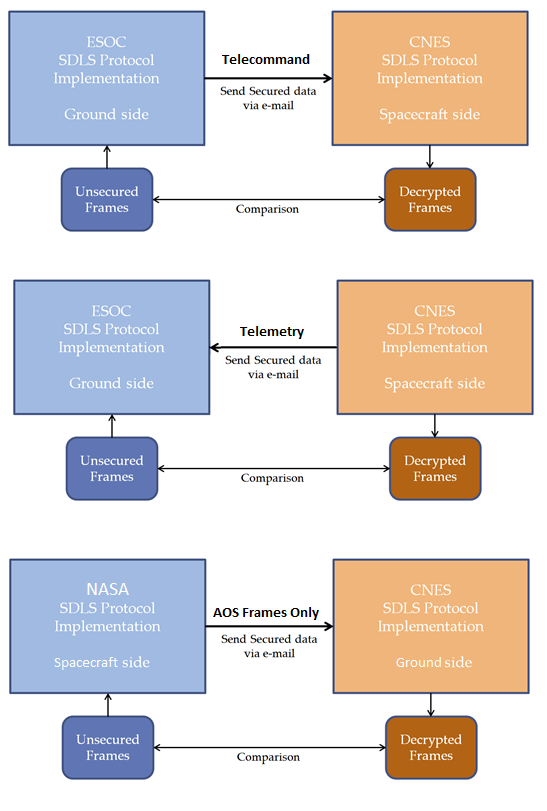


Figure 1 - Test procedure for TC, TM and AOS

For the TC testing, SCOS-2000[[3]](#footnote-3) is used to generate TC frames. After generating the frames the Security Unit (ESOCs SDLS protocol implementation) manipulates the data as configured (authenticates, encrypts or authenticates & encrypts the frames). Finally, the data is converted into CLTUs[[4]](#footnote-4), which is needed by the spacecraft simulator, and sent via e-mail to CNES. On the spacecraft side, the received CLTUs are fed into the spacecraft simulator, (CNES’ SDLS protocol implementation), analysed and the output is compared to the original TC frames of the ground side. On the event that an error is detected it should be notified.

For TM testing, CNES generates TM frames and sends them via e-mail to ESOC. These frames are converted into hexadecimal, saved in a .txt file and loaded into the Security Unit Test Environment[[5]](#footnote-5) to be sent one-by-one to the Security Unit. The decrypted frames are saved to a separate .txt file and can be sent to CNES to be compared with the original data.

For AOS testing, NASA generates AOS frames in binary format and sends them via email to CNES. These frames are processed and the decrypted frames/results are saved in a text file and sent to NASA to be compared with the original data.

# Test Case #1

The first test case, described in (CCSDS, Space Data Link Security Protocol Test Report (CCSDS 355.1-Y-draft) 2014), concentrates on sending TC frames to the spacecraft bypassing the COP-1 protocol (BD frames). The SDLS protocol is validated in the baseline mode of TC - defined in annex E of SDLS recommendation[[6]](#footnote-6).

## Test Configuration

Following parameters are set/used throughout the test:

* Data type: CLTUs
* VC ID: 63
* Segment header: present
* MAP ID: 1
* Only expedited service (no COP): BD frames
* Security service type: authentication only
* Cryptographic algorithm: AES-CMAC
* Key: 86312f73ea6340afc9b5be67fe2a4eeb (128 bits)
* Anti-replay sequence number: 32 bits transmitted integrally in Security Header
* Total Security Header length: 6 octets
* MAC: 128 bits
* Authentication bit mask: default bit mask as defined in SDLS recommendation

§4.2.2.6.2

* Anti-replay sequence number window: 5
* Mix of correct and defective (security errors) CLTUs

## Test Process

To generate the CLTUs for this test case ESOC uses a virtual machine with end-to-end space communication simulation software (including SCOS-2000, NIS, TMTCS and GSTVi-SCM). The CLTUs are generated by SCOS-2000 and secured by the Security Unit. The CLTUs can then be read as incoming data from the GSTVi Log Viewer.

The following table shows the procedure of creating and sending the test CLTUs (ESOC):

Table 1 - ESOC procedures for Test Case #1

|  |  |  |
| --- | --- | --- |
| Step | Input | Output – Expected result |
| 1 | Start the end-to-end space communication simulation VM and configure the Security Units + SCOS as described in 3.1. | All software should be running without problems. TMs and TCs can be processed. |
| 2 | Send TCs using SCOS’ “Manual Stack”. If a forced error is desired, then select an error on the Security Units MMI (ground side) before sending the next frame. | Manual Stack is empty after sending the command(s). |
| 3 | Check TMTCS forward session layer status | “TCReq PDU Id” should be incremented by 1 |
| 4 | Check GSTVi “Log Viewer” | Incoming CLTU is displayed in the Log Viewer. If not, then activate “Raw CLTU” at “S/C Tracing”. |
| 5 | Copy the CLTU from the Log Viewer into a file. | A file contains all created CLTUs. |
| 6 | Send the file to CNES via e-mail. |  |

The simulator used by CNES is composed of processes implementing the ground side or the spacecraft side of the TC link. For this test, the secured CLTUs received via email from ESOC are ingested and processed by the spacecraft side of the SDLS simulator (SDLS Simulator V01\_01).

The following table shows the procedure of receiving and analysing the test CLTUs (CNES):

Table 2 - CNES procedures for Test Case #1

|  |  |  |
| --- | --- | --- |
| Step | Input | Output – Expected result |
| 1 | Edit the simulator’s configuration files to define the SDLS parameters (SAs, keys, etc.) | Configuration files in accordance with the test to be done |
| 2 | Copy the CLTUs received via email from ESOC in a file | File containing the secure CLTUs |
| 3 | Run the simulator | Correct processing of CLTU file. Log files / Output files generated. |
| 4 | Open the log files produced by the simulator | Security errors / transmission errors logged. Complete unsecure frames present. |
| 5 | Extract the errors (if any) and unsecure frames from the log files | Unsecure frames, errors log |
| 6 | Send report to ESOC via email (frames with and without errors, etc.) | CLTU processed as expected, correct detection of all intended errors, no more errors than expected |

The CLTU test set simulates the following scenarios as defined in the CCSDS SDLS Protocol Test Report:

* No errors
* Replay
* Anti-replay sequence number mismatch
* MAC error (random errors, forging of frame subfields)
* Invalid SPI

The scenario “Key mismatch” was not simulated as it would simply resolve in a MAC error.

The test was also restricted to one VC and one MAP.

## Test Results

The tests were successful and all security errors were identified. The following table lists the secured CLTUs, the expected results and the actual results from CNES.

Table 3 - Test Results of Test Case #1

|  |  |  |  |
| --- | --- | --- | --- |
| **CLTU Nr.** | **CLTU** | **Expected Result** | **Results by CNES** |
| 1 | eb902003fc2900c100960300000001181c7cd68a0005190c0abe0066bed2fb6799fe64ae0bf2558a9734e0e130a00488d8b6c5c5c5c5c5c5c579 | CLTU check: OK | CLTU check: OK |
| 2 | eb902003fc2900c100960300000002181c8ad68b0005190c08e800b8bd8ee3323534ea738775c97c256a93a74fa8c056a666c5c5c5c5c5c5c579 | CLTU check: OK | CLTU check: OK |
| 3 | eb902003fc2900c100960300000002181c8ad68c0005190c080e00a1f915876bd6d0edf843ef364c860aec2c818f3d7e7816c5c5c5c5c5c5c579 | CLTU check: NOK  Security Error: Replay | CLTU check: NOK  Security Error: Replay |
| 4 | eb902003fc2900c1009603000000031821c8c00400051904039c00fbb8da2a8a72c4112f4c5aaf2f91c8e85674073f75ed62c5c5c5c5c5c5c579 | CLTU check: OK | CLTU check: OK |
| 5 | eb902003fc2900c100960300000004181cecd68d0005190d0d2c00d15d2d1bf7368c972c85840dcaa30213139aa7d69cab9cc5c5c5c5c5c5c579 | CLTU check: NOK  Security Error: MAC | CLTU check: NOK  Security Error: MAC |
| 6 | eb902003fc2900c100960300000005181cbed68e0005190d0dfe0009df65449308f20d50980ffeda43aa35baf306535ed666c5c5c5c5c5c5c579 | CLTU check: OK | CLTU check: OK |
| 7 | eb902003fc2900c10096030000000b181cd6d68f0005190d0d3600b1be83749a95660f313fef70a91aa4631982efcb5d1678c5c5c5c5c5c5c579 | CLTU check: NOK  Security Error: SN out of boundary | CLTU check: NOK  Security Error: SN out of boundary |
| 8 | eb902003fc2900c10096ff00000006181ce4d6900005190d0d68000c2ce4cd37b25a17e722caab42e3c8c9dd6eccab20ee18c5c5c5c5c5c5c579 | CLTU check: NOK  Security Error: invalid SPI | CLTU check: NOK  Security Error: invalid SPI |
| 9 | eb902003fc2900c100960300000007181c1ad6910005190c0ad4001aeae8b635104a9bb561fb134a926067eaa20d20c815d4c5c5c5c5c5c5c579 | CLTU check: OK | CLTU check: OK |
| 10 | eb902003fc2900c100960300000008181c20d6920005190c089800a40ad6e91f2254ed516f9b92ba7294dfc413fe642bca22c5c5c5c5c5c5c579 | CLTU check: OK | CLTU check: OK |

# Test Case #2

The second test case, described in (CCSDS, Space Data Link Security Protocol Test Report (CCSDS 355.1-Y-draft) 2014), concentrates on sending TC frames to the spacecraft using the COP-1 protocol (AD frames). The TC frames apply the TC baseline mode defined in annex E of the SDLS recommendation[[7]](#footnote-7).

## Test Configuration

Following test specifications are set/used throughout the test:

* Data type: CLTUs
* VC ID: 63
* Segment header: present
* MAP ID: 1
* Mix of AD frames and COP-1 control commands (BC frames)
* Security service type: authentication only
* Cryptographic algorithm: AES-CMAC
* Key: 86312f73ea6340afc9b5be67fe2a4eeb (128 bits)
* Anti-replay sequence number: 32 bits transmitted integrally in Security Header
* Total Security Header length: 6 octets
* MAC: 128 bits
* Authentication bit mask: default bit mask as defined in SDLS recommendation

§4.2.2.6.2

* Anti-replay sequence number window: 5
* Mixture of correct and defective (security errors and transmission/FEC errors) CLTUs

## Test Process

To generate CLTUs for this test case the virtual machine with end-to-end space communication simulation software is needed. The CLTUs are generated by SCOS-2000 and secured by the Security Unit. The CLTUs can be read as incoming data from the GSTVi Log Viewer. The difference to test case #1 is that the frames are a mixture of AD- and BD-Frames which needs to be considered while setting up SCOS, using the “TC SPACON” tool.

The following table shows the procedure of creating and sending the test CLTUs (ESOC):

Table 4 - ESOC procedures for Test Case #2

|  |  |  |
| --- | --- | --- |
| Step | Input | Output – Expected result |
| 1 | Start the end-to-end space communication simulation VM and configure the Security Units + SCOS as described in 4.1. | All software should be running without problems. TMs and TCs can be processed. |
| 2 | Send TCs using SCOS’ “Manual Stack”. If a forced error is desired, then select an error on the Security Units MMI (ground side) before sending the next frame. | Manual Stack is empty after sending the command(s). |
| 3 | Check TMTCS forward session layer status | “TCReq PDU Id” should be incremented by 1 |
| 4 | Check GSTVi “Log Viewer” | Incoming CLTU is displayed in the Log Viewer. If not, then activate “Raw CLTU” at “S/C Tracing”. |
| 5 | Copy the CLTU from the Log Viewer into a file. | A file contains all created CLTUs. |
| 6 | To generate an FEC failure it is necessary to manipulate the desired frames manually. | The FECFs are manipulated on the selected frames. |
| 7 | Send the file to CNES via e-mail. |  |

The simulator used by CNES is composed of processes implementing the ground side or the spacecraft side of the TC link. For this test, the secured CLTUs received via email from ESOC are ingested and processed by the spacecraft side of the SDLS simulator (SDLS Simulator V01\_01). The following table shows the procedure of receiving and analysing the test CLTUs (CNES):

Table 5 - CNES procedures for Test Case #2

|  |  |  |
| --- | --- | --- |
| Step | Input | Output – Expected result |
| 1 | Edit the simulator’s configuration files to define the SDLS parameters (SAs, keys, etc.) | Configuration files in accordance with the test to be done |
| 2 | Copy the CLTUs received via email from ESOC in a file | File containing the secure CLTUs |
| 3 | Run the simulator | Correct processing of CLTU file. Log files / Output files generated. |
| 4 | Open the log files produced by the simulator | Security errors / transmission errors logged. Complete unsecure frames present. |
| 5 | Extract the errors (if any) and unsecure frames from the log files | Unsecure frames, errors log |
| 6 | Send report to ESOC via email (frames with and without errors, etc.) | CLTUs processed as expected, correct detection of all intended errors, no more errors than expected |

The CLTU test set simulates the following scenarios as defined in the CCSDS SDLS Protocol Test Report:

* No errors, no control commands
* Security exceptions/errors:
  + Replay
  + MAC error (forging of frame subfields : incorrect MAC but correct FEC)
* Transmission errors (incorrect FEC):
  + Single bit error
  + Burst errors (impacting multiple frames)
  + Errors on AD and BC frames
* COP-1 reconfiguration
  + Set V[R]

Only two security errors were tested as that is sufficient to prove the independency of the COP-1 and the SDLS protocol. The other security exceptions were tested during test case #1.

## Test Results

The tests were successful and all errors were identified. The following table lists the secured CLTUs, the expected results and the actual results from CNES.

Table 6 - Test Results of Test Case #2

|  |  |  |  |
| --- | --- | --- | --- |
| **CLTU Nr.** | **CLTU (with COP)** | **Expected Result** | **Results by CNES** |
| 1 | eb903003fc090082004601d858555555552ac5c5c5c5c5c5c579 | COP check: OK  BC-Frame, V(R) => 1 | COP check OK, V(R) = 1 |
| 2 | eb900003fc2901c100300300000002181c8ad73b0005190c0afe006b78208038958cd8890af1d418182a2128d6a103a803f6c5c5c5c5c5c5c579 | COP check: OK  AD-Frame, V(R) = 1  Security check: OK | COP check OK, V(R) = 1.  Security check OK. |
| 3 | eb900003fc2902c100c60300000002181c8ad73c0005190c0a1800723c3463de2ab6b8ecb52fa22ec20a10bd68f90d6b523cc5c5c5c5c5c5c579 | COP check: OK  AD-Frame, V(R) = 2  Security check: NOK  Security Error: Replay | COP check OK, V(R) = 2.  Security check NOK : replay detected. |
| 4 | eb900003fc2903c100940300000003181cd8d73d0005190c0ad000ca5d347db7a80a628e88f320b465dc6842e5bc909d2b32c5c5c5c5c5c5c579 | COP check: OK  AD-Frame, V(R) = 3  Security check: OK | COP check OK, V(R) = 3.  Security check OK |
| 5 | eb900003fc2904c100a00300000004181cecd73e0005190c0a020012df0ac7e5fecee385dd46b93ee65893646a12faa74b74c5c5c5c5c5c5c579 | COP check: NOK  AD-Frame, FEC error | COP check NOK (BCH / CRC error)  Security check OK (the error is located in the last byte of the CRC itself, not in the authenticated data). |
| 6 | eb903003fc090082004601d800555555552ac5c5c5c5c5c5c579 | COP check: NOK  BC-Frame, FEC error | COP check NOK (BCH error) |
| 7 | eb903003fc090082004601d858555555552ac5c5c5c5c5c5c579 | COP check: OK  BC-Frame, V(R) => 1 | COP check OK, V(R) = 1. |
| 8 | eb900003fc2901c100300300000007181c1ad73f0005190c0aca00aabeb5534221668af31efc97927008d7da6af0e060fe6cc5c5c5c5c5c5c579 | COP check: OK  AD-Frame, V(R) = 1  Security check: OK | COP check OK, V(R) = 1.  Security check OK. |
| 9 | eb900003fc2902c100c60300000008181c20d7400005190c0aea00a43639b371a1d81be0a3dd4dc35254acc3fb25c1fa16cec5c5c5c5c5c5c579 | COP check: OK  AD-Frame, V(R) = 2  Security check: OK | COP check OK, V(R) = 2.  Security check OK. |
| 10 | eb900003fc2903c100940300000009181c72d7410005190c0a22001c578a5d9468dcf6990586dd4b5f0062b6b69f6a078fbec5c5c5c5c5c5c579 | COP check: OK  AD-Frame, V(R) = 3  Security check: NOK  Security Error: MAC | COP check OK, V(R) = 3.  Security check NOK : MAC error. |

Note : for CLTU No. 5, the result shows a security check OK despite a COP check NOK (BCH error). This is because it has been deliberately chosen to make the SDLS spacecraft simulator process all the frames, even if an error is detected before, in order to see the SDLS protocol generating a MAC error. However, in this case, there is no MAC error, because the erroneous byte is not part of the authenticated data (last byte of the CRC at the end of the TC frame).

# Test Case #3

The third test case, described in (CCSDS, Space Data Link Security Protocol Test Report (CCSDS 355.1-Y-draft) 2014), concentrates on sending TM frames to the ground. The SDLS protocol is validated in the baseline mode for TM - defined in annex E of SDLS recommendation[[8]](#footnote-8).

## Test Configuration

Following parameters are set/used throughout the test:

* Data type: TM Frames
* OCF: not present
* FECF: not present
* SCID: 3
* VCID: 2
* Security service type: authenticated encryption
* Cryptographic algorithm: AES-GCM
* Key: 000102030405060708090a0b0c0d0e0f (128 bits)
* Anti-replay sequence number: not needed in AES-GCM
* Total Security Header length: 14 octets (SPI = 2 octets, IV = 12 octets)
* MAC: 128 bits
* Authentication bit mask: default bit mask as defined in SDLS recommendation

§4.2.2.6.2

* Anti-replay sequence number window: 5
* Mixture of correct and defective (security errors) frames

## Test Process

The simulator used by CNES is composed of processes implementing the ground side or the spacecraft side of the TM link. For this test, a file containing unsecure TM frames is processed by the spacecraft side of the simulator (SCTM v1.01). The resulting output file contains SDLS secure TM frames, and is sent to ESOC via email for processing.

The following table shows the procedure of creating and sending the test TMs (CNES):

Table 7 - CNES procedures for Test Case #3

|  |  |  |
| --- | --- | --- |
| Step | Input | Output – Expected result |
| 1 | Edit the simulator’s configuration files to define the SDLS parameters (SAs, keys, etc.) | Configuration files in accordance with the test to be done |
| 2 | Create or edit the file containing the unsecure TM frames (if necessary) | File containing the TM frames to be processed by the TM spacecraft simulator |
| 3 | Run the simulator | Correct processing of the TM file. Output file generated, containing the secure TM frames. |
| 4 | Edit the output file, to manually insert errors (MAC errors) if necessary | File containing secure TM frame (some of them with MAC errors) |
| 5 | Send secure TM file via email to ESOC | Secure TM file correctly processed by ESOC’s simulator, correct detection of all intended errors, no more errors than expected. |

After receiving the binary file containing the secured TM frames, ESOC converts it into a .txt file with a hexadecimal frame format. The file can then be loaded into a test environment which connects directly to the Security Unit to analyze the frames one by one.

The following table shows the procedure of receiving and analyzing the test TMs (ESOC):

Table 8 - ESOC procedures for Test Case #3

|  |  |  |
| --- | --- | --- |
| Step | Input | Output – Expected result |
| 1 | Start and configure the Security Unit (using eclipse). Furthermore, start the Security Unit Test Environment. | All software should be running without problems. |
| 2 | Load the .txt file containing the TM frames into the Test Environment | The TM frames should be displayed in the text box on the Test Environments MMI. |
| 3 | Select a TM frame and send it to the Security Unit. | If the software is running in eclipse, data should be written to the console. Moreover, a .txt file should be created after the process containing the decrypted frame. |
| 4 | Repeat step 3 with all TM frames. | With each selected frame the .txt file should become longer. |
| 6 | Send the decrypted frames to CNES via e-mail to compare them with the original. |  |

## Test Results

The tests were successful and all security errors were identified. The following table lists the secured TM frames, the expected results from CNES and the actual results.

Table 9 - Test Results of Test Case #3

|  |  |  |  |
| --- | --- | --- | --- |
| **FrameNr.** | **TM Frames** | **Expected Result** | **Results by ESOC** |
| 1 | 003410001800007b0000000000000000000426920413f836a…d7944701f56559818e564222e98b11 | TM decryption: No errors | TM decryption: No errors |
| 2 | 003411011800007b000000000000000000042693c754a3f32…4d04dc9ed32bd92e686dd4d7072045 | TM decryption: No errors | TM decryption: No errors |
| 3 | 003412021800007b00000000000000000004269462357abab…0bc31dd427f8579557f8dac74ad4be | TM decryption: No errors | TM decryption: No errors |
| 4 | 003413031800007b000000000000000000042695cce952bdc…73758f66db248f3e5fc2f67e0162aca | TM decryption: No errors | TM decryption: No errors |
| 5 | 003414041800007c00000000000000000004269618f854d25…e2e75d8b4bdf627d5e6e9cefd13c870 | TM decryption: Invalid SPI/SA | TM decryption: Invalid SPI/SA |
| 6 | 003415051800007b00000000000000000004269774c5b2732…198397258be3291302f2673c7b0fab9 | TM decryption: No errors | TM decryption: No errors |
| 7 | 003416061800007b0000000000000000000426971f9fe3089…f25c9a8c5c321748189d82724acc9de4 | TM decryption: SN/IV Replay | TM decryption: SN/IV Replay |
| 8 | 003417071800007b000000000000000000042699d1113c5802…2d16c9a1c46d68375878f5fbd418bae | TM decryption: No errors | TM decryption: No errors |
| 9 | 003418081800007b00000000000000000004269a6b7160f58166…8f724161735f08b282465472cf51cf0 | TM decryption: No errors | TM decryption: No errors |
| 10 | 003419091800007b00000000000000000004269b166267f43…ac70b96c06583a1c5d8f73b06a599828 | TM decryption: MAC error | TM decryption: MAC error |
| 11 | 00341a0a1800007b00000000000000000004269c84b6c2b8b58d2…64dc2673e90b3daaceb471d2cca8ee895761 | TM decryption: MAC error | TM decryption: MAC error |
| 12 | 00341b0b1800007b00000000000000000004269d2f8b7d642e…775b585f0271ae54f7c3599cbce93d945e | TM decryption: No errors | TM decryption: No errors |
| 13 | 00341c0c1800007b00000000000000000004269e757d385075af40…ef9aa509593d0fbcdb6656ef655adbe9f5f977 | TM decryption: No errors | TM decryption: No errors |
| 14 | 00341d0d1800007b00000000000000000004269f260dee4d9e8…d674b54134fea3b4a0066638913ab3d1f0ed5a | TM decryption: No errors | TM decryption: No errors |
| 15 | 00341e0e1800007b0000000000000000000426a0f6741dec3bb8…8e776240db42a13d5cc07cb1fa3d061743e8 | TM decryption: No errors | TM decryption: No errors |
| 16 | 00341f0f1800007b0000000000000000000426a1cb39811…1a9aa0ffe8ca6b86e201d9747a95f97a27b23 | TM decryption: No errors | TM decryption: No errors |
| 17 | 003420101800007b0000000000000000000426a269d47d69…591843c34044ea5208d6546a746565ef | TM decryption: No errors | TM decryption: No errors |
| 18 | 003421111800007b0000000000000000000426a319283759…574d3f6a92928c7b35f3de35cb09ab2 | TM decryption: No errors | TM decryption: No errors |
| 19 | 003422121800007b0000000000000000000426a46be51381cc869c…f154619ce822becb39438da5bbd95cb003848bc5e34420 | TM decryption: No errors | TM decryption: No errors |
| 20 | 003423131800007b0000000000000000000426a5ab008738c5…2eaea07e6ae1146167c2383e1099df374ef40 | TM decryption: No errors | TM decryption: No errors |

All successfully decrypted TM frames contained the following data: 00112233445566778899aabbccddeeff0011223344556677889… 9aabbccddeeff00ffffffffffff

The test was limited to one virtual channel only.

# Test Case #4

The forth test case, described in (CCSDS, Space Data Link Security Protocol Test Report (CCSDS 355.1-Y-draft) 2014), concentrates on sending AOS frames to the ground. The SDLS protocol is validated in the baseline mode for AOS - defined in annex E of SDLS recommendation[[9]](#footnote-9).

## Test Configuration

Following parameters are set/used throughout the test:

* Data Type: AOS Frames
* Transfer Frame Length: 292 octets
* FHEC: not present
* Insert Zone: not present
* OCF: not present
* FECF: not present
* SCID: 0x28
* VCID: 0
* Security service type: authenticated encryption
* SPI : 1
* Cryptographic algorithm: AES-GCM
* Key: b2838fafb8533e422920565540b2112b (128 bits)
* Anti-replay sequence number: not needed in AES-GCM
* Total Security Header Length: 14 octets (SPI = 2 octets, IV = 12 octets)
* Initial IV: 00000000000000004e415341
* MAC: 128 bits
* Authentication bit mask: default bit mask as defined in SDLS recommendation
* Anti-replay sequence number window: 10
* Mixture of correct and defective (security errors) frames

## Test Process

Secure AOS frame are generated by NASA using a simulator implementing the spacecraft side of SDLS. These AOS frames are then sent by email to CNES for processing.

The following table shows the procedure of creating and sending the test AOS frames (NASA):

Table 10 - NASA procedures for Test Case #4

|  |  |  |
| --- | --- | --- |
| Step | Input | Output - Expected result |
| 1 | Edit the associated configuration files to define the SDLS parameters (SAs, keys, etc.) | Configuration files in accordance with the test to be done |
| 2 | Run the simulator | Output file generated, containing the secure AOS frames. |
| 3 | Send secure AOS file via email to CNES | Secure AOS file correctly processed by CNES’s simulator, correct detection of all intended errors, no more errors than expected. |

The simulator used by CNES is composed of processes implementing the ground side or the spacecraft side of the AOS link. For this test, the file containing secured AOS frames received from NASA is processed by the ground side of the simulator (AOS Receiver v1.02). The resulting output file contains SDLS unsecure AOS frames, and is sent to NASA via email for comparison against the original data. An additional log file contains the status of each processed frame (no error, MAC error, SPI error, etc.)

The following table shows the procedure of receiving and analysing the test AOSs (CNES):

Table 11 - CNES procedures for Test Case #4

|  |  |  |
| --- | --- | --- |
| Step | Input | Output - Expected result: |
| 1 | Edit the simulator’s configuration files to define the SDLS parameters (SAs, keys, etc.) | Configuration files in accordance with the test to be done |
| 2 | Copy the AOS frames received via email from NASA in a file | File containing the secure AOS frames |
| 3 | Run the simulator | Correct processing of AOS frames file. Log file / Output file generated. |
| 4 | Open the log file produced by the simulator | Security errors / transmission errors logged. |
| 6 | Open the output file produced by the simulator | Complete unsecure frames present. |

## Test Results

The tests were successful and all security errors were identified. The following table lists the secured AOS frames, the expected results from NASA and the actual results from CNES.

Table 12 - Test Results of Test Case #4

|  |  |  |  |
| --- | --- | --- | --- |
| **Frame Nr.** | **AOS Frames** | **Expected Result** | **Results by CNES** |
| 1 | Primary\_Header: 4a0049544300  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: No errors | AOS decryption: No errors |
| 2 | Primary\_Header: 4a0049544400  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: No errors | AOS decryption: No errors |
| 3 | Primary\_Header: 4a0049544500  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: No errors | AOS decryption: No errors |
| 4 | Primary\_Header: 4a0049544600  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: No errors | AOS decryption: No errors |
| 5 | Primary\_Header: 4a0049544700  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: No errors | AOS decryption: No errors |
| 6 | Primary\_Header: 4a0049544800  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: SN/IV Replay error | AOS decryption: SN/IV Replay error |
| 7 | Primary\_Header: 4a0049544900  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: No errors | AOS decryption: No errors |
| 8 | Primary\_Header: 4a0049544a00  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: No errors | AOS decryption: No errors |
| 9 | Primary\_Header: 4a0049544b00  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: No errors | AOS decryption: No errors |
| 10 | Primary\_Header: 4a0049544c00  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: MAC error | AOS decryption: MAC error |

|  |  |  |  |
| --- | --- | --- | --- |
| 11 | Primary\_Header: 4a0049544d00  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: No errors | AOS decryption: No errors |
| 12 | Primary\_Header: 4a0049544e00  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: No errors | AOS decryption: No errors |
| 13 | Primary\_Header: 4a0049544f00  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: SPI Error | AOS decryption: SPI Error |
| 14 | Primary\_Header: 4a0049545000  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: No errors | AOS decryption: No errors |
| 15 | Primary\_Header: 4a0049545100  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: No errors | AOS decryption: No errors |
| 16 | Primary\_Header: 4a0049545200  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: No errors | AOS decryption: No errors |
| 17 | Primary\_Header: 4a0049545300  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: No errors | AOS decryption: No errors |
| 18 | Primary\_Header: 4a0049545400  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: No errors | AOS decryption: No errors |
| 19 | Primary\_Header: 4a0049545500  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: No errors | AOS decryption: No errors |
| 20 | Primary\_Header: 4a0049545600  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: MAC error | AOS decryption: MAC error |
| 21 | Primary\_Header: 4a0049545700  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: No errors | AOS decryption: No errors |
| 22 | Primary\_Header: 4a0049545800  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: SN/IV Replay error | AOS decryption: SN/IV Replay error |
| 23 | Primary\_Header: 4a0049545900  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: No errors | AOS decryption: No errors |
| 24 | Primary\_Header: 4a0049545a00  Security\_Header: 0000000000000000000000000000  Data\_Field: 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f303132333435363738393a3b3c3d3e3f404142434445464748494a4b4c4d4e4f505152535455565758595a5b5c5d5e5f606162636465666768696a6b6c6d6e6f707172737475767778797a7b7c7d7e7f808182838485868788898a8b8c8d8e8f909192939495969798999a9b9c9d9e9fa0a1a2a3a4a5a6a7a8a9aaabacadaeafb0b1b2b3b4b5b6b7b8b9babbbcbdbebfc0c1c2c3c4c5c6c7c8c9cacbcccdcecfd0d1d2d3d4d5d6d7d8d9dadbdcdddedfe0e1e2e3e4e5e6e7e8e9eaebecedeeeff0f1f2f3f4f5f6f7f8f9fafbfcfdfeff  Security\_Trailer: 00000000000000000000000000000000 | AOS decryption: No errors | AOS decryption: No errors |

The test was limited to one virtual channel only.

# Conclusion

Test cases #1-4 were processed as described, with the result that the inter-agency static test was successful. The individual results of the test cases are displayed below.

*Test Case #1:*

All test scenarios of test case #1 succeeded, as the spacecraft simulator detected every security error contained in the BD-Frames[[10]](#footnote-10).

*Test Case #2:*

All test scenarios of test case #2 were successful. No interaction between the COP commands and SDLS protected frames was found.

*Test Case #3:*

All test scenarios of test case #3 succeeded, as the ground system detected every security error contained in the telemetry frames.

*Test Case #4:*

All test scenarios of test case #4 succeeded, as the ground system detected every security error contained in the AOS frames.

# Acronyms

CCSDS Consultative Committee for Space Data Systems

TM Telemetry

TC Telecommand

SDLS Protocol Space Data Link Security Protocol

SCOS Spacecraft Control and Operations System

GSTVi Ground Systems Test and Validation infrastructure

TMTCS Telemetry & Telecommand System

TIF TMTCS Direct Interface

NIS Network Interface System

VM Virtual Machine

SA Security Association

MMI Man-Machine Interface

MAC Message Authentication Code

SN Sequence Number

IV Initialisation Vector

AES-GCM Advanced Encryption Standard – Galois Counter Mode

VC Virtual Channel

1. (CCSDS, Space Data Link Security Protocol Test Report (CCSDS 355.1-Y-draft) 2014) [↑](#footnote-ref-1)
2. The French space agency [↑](#footnote-ref-2)
3. MCS of ESOC [↑](#footnote-ref-3)
4. TC Frames with additional data to match the SLE protocol [↑](#footnote-ref-4)
5. ESOC tool to send TC and TM frames directly to the Security Unit [↑](#footnote-ref-5)
6. (CCSDS, Space Data Link Security Protocol (CCSDS 355.0-R-4) 2014) [↑](#footnote-ref-6)
7. (CCSDS, Space Data Link Security Protocol (CCSDS 355.0-R-4) 2014) [↑](#footnote-ref-7)
8. (CCSDS, Space Data Link Security Protocol (CCSDS 355.0-R-4) 2014) [↑](#footnote-ref-8)
9. (CCSDS, Space Data Link Security Protocol (CCSDS 355.0-R-4) 2014) [↑](#footnote-ref-9)
10. Telecommand frames not protected by the COP-1 protocol [↑](#footnote-ref-10)