Finalizing the CCSDS Space-Data Link Layer Security Protocol: Setup and Execution of the Interoperability Testing

Daniel Fischer[1] and Ignacio Aguilar-Sanchez[2]
European Space Agency

Bruno Saba[3], Gilles Moury[4]
Centre National d'Etudes Spatiales (CNES), France

Craig Biggerstaff[5]
Lockheed Martin, United States

Brandon Bailey[6]
NASA Goddard Space Flight Centre, United States

Howard Weiss[7]
NASA/JPL/Parsons, United States

and

Martin Pilgram[8], Dorothea Richter[9]
German Aerospace Centre (DLR), Germany

The protection of data transmitted over the space-link is an issue of growing importance also for civilian space missions. Through the Consultative Committee for Space Data Systems (CCSDS), space agencies have reacted to this need by specifying the Space Data-Link Layer Security (SDLS) protocol which provides confidentiality and integrity services for the CCSDS Telemetry (TM), Telecommand (TC) and Advanced Orbiting Services (AOS) space data-link protocols. This paper describes the approach of the CCSDS SDLS working group to specify and execute the necessary interoperability tests. It first details the individual SDLS implementations that have been produced by ESA, NASA, and CNES and then the overall architecture that allows the interoperability tests between them. The paper

[1] Data System Manager, Ground Engineering Support Office, ESA/ESOC, Robert-Bosch-Str. 5, 64293 Darmstadt, Germany

[2] Communications System Engineer, Telecommunications, TT&C Systems and Techniques Section, ESA/ESTEC, Keplerlaan 1, 2200 AG Noordwijk, The Netherlands.

[3] Digital Electronics Expert Engineer, On-Board Handling Department, CNES, 18 Avenue Edouard Belin, 31401 Toulouse, France.

[4] Senior Adviser, Co-Chair CCSDS SDLS WG, Spacecraft Technologies Department, CNES, 18 avenue Edouard Belin, 31401 Toulouse Cedex, France

[5] Security Systems Engineer, Mission Systems Division, 2101 NASA Parkway / Mail Code CD22, Houston, Texas 77058 USA

[6] Cybersecurity Lead, NASA's Independent Verification and Validation Program, Goddard Space Flight Centre

[7] Technical Director, Chairman CCSDS Security WG, Cobham, 7110 Samuel Morse Drive, Columbia MD 21046 USA

[8] Project Security Manager, DLR Oberpfaffenhofen, Münchener Strasse 20, 82234 Wessling, Germany

[9] Project Security Manager, DLR Oberpfaffenhofen, Münchener Strasse 20, 82234 Wessling, Germany

reports on the results of the interoperability tests and identifies relevant aspects for the evolution of the test environment.

I. Introduction

INFORMATION security is becoming more and more important for space agencies and spacecraft operators in general. With ubiquitous connectivity and cheap communication technology, the threat related to malicious attacks on spacecraft operation infrastructures is increasing. The Space-Link supports the critical telecommand and telemetry communications between the spacecraft and the ground segments. In an overall Space Mission Security Architecture the protection of the space link is, therefore, a key priority. Threats to this link can affect the command and monitoring of the spacecraft being supported or the instrument data being downlinked. Attacks can range from illegal interception of the communications and the data being transmitted (eavesdropping) to attempt to masquerade and, therefore, take unauthorized control of the spacecraft (spoofing) or deny service to both the spacecraft and the mission control system (e.g. RF jamming).

Most civilian spacecraft operators are using the command & control protocol suite provided by the Consultative Committee for Space Data Systems (CCSDS). Since this protocol suite is currently not supporting the provision of security services (i.e. confidentiality & integrity), space links are not secured at all or secured using proprietary implementations. During the recent years CCSDS has been developing the Space-Data Link Layer Security Protocol (SDLS) to close this gap[11]. The SDLS protocol is conceptually completed but in order to be published as a standard it needs to be validated through at least two independent implementations interoperating with each other. These interoperability implementations and testing campaigns have been completed by three different agencies – CNES, ESA, and NASA.

In our paper, we briefly introduce the SDLS protocol, which has been covered by other publications already[1,2], and then focus on the interoperability testing process and results. We describe in detail the approach to the testing and the test cases, the individual implementations and their operational environment, and the actual test execution also in the light of a successful co-operation between multiple space agencies. We also provide important lessons-learned that will reduce the complexity of future interoperability tests.

A. The Space Data Link Security (SLDS) Protocol

Protecting the space link implies protecting the confidentiality, integrity and availability (CIA) of the space communications links and the telecommand and telemetry data being exchanged. As a matter of fact, priority has been given to the protection of the confidentiality and integrity of these data. In particular, with the specification and integration of security services like authentication, encryption and authenticated encryption through the SDLS protocol the space missions' community have a modular solution to the protection of some of the main threats on the space link. Threats like spoofing and eavesdropping can be mitigated by implementing the desired SDLS services on the space link.

SDLS is compatible with the well-known and widely deployed CCSDS Space Data Link protocols: Telecommand (TC)[3], Telemetry (TM)[4] and Advanced Orbiting Systems (AOS)[5]. Furthermore, it is both compatible to a certain extent, and reliant on, the application of the CCSDS Space Link Extension (SLE) services[6] to effectively extend the protection to the end-to-end data communications between spacecraft and mission control system. With complementary protection of the ground communications network between the ground station(s) and the mission control system as well as the mission control system itself and its interfaces with end users, a complete Space Mission Security Architecture is achieved.

B. Interoperability Testing

CCSDS requires at least two independent implementations that are able to successfully complete a series of end-to-end interaction test cases as prerequisite for the publication of technical standards. The CCSDS SDLS working group defined the necessary test cases to validate the protocol.

However, the testing itself did not prove to be a straight-forward task. One major problem is the fact that not all space agencies are using the full CCSDS Space Data-Link (SDL) communication protocol suite that is protected by SDLS. As an example, some NASA missions e.g. Global Precipitation Measurement (GPM) are mainly working with TC/AOS, while ESA and CNES are using the TM/TC protocols for the command & control of robotic spacecraft. Thus, a baseline implementation was not easily identified for the interoperability implementations. Furthermore, since SDLS is a space-link protocol, not only the ground systems side needs implementation, but an additional implementation is required for spacecraft simulators as well in order to support end-to-end testing.

Finally, the SDLS protocol supports a large number of configurations e.g. in terms of supported crypto algorithms, crypto periods etc. It is not possible to test all these configurations. Thus, the concept of a baseline mode has been introduced. It is a standard configuration in which the protocol is assumed to be deployed. In this paper, we describe each of these issues in more detail.

Taking into account all these constraints, CNES, ESA, and NASA agreed to develop prototype implementations of the SDLS protocol and then set up an interoperability testing environment. We describe all three implementations in detail as well as the full interoperability test setup, execution and results.

C. Lessons Learned

The definition and execution of the interoperability test for the SDLS protocol was more challenging than initially thought. Besides a number of technical reasons that are mentioned in this paper (baseline mode, ground and space implementations, varying protocol support), the execution of a test, especially a real-time test, between agencies is a challenging task in terms of management and network security concerns. One of the recommendations that were born out of this exercise is the establishment of a testing cloud/ environment for CCSDS. While the setup of such a testing environment brings a number of challenges, once in place, it could significantly simplify the interoperability testing not only for security protocols.

D. Paper Structure

The remainder of this paper is organized as follows. In Section II, we briefly introduce the Space Data Link Security (SDLS) Protocol, putting emphasis and motivation and technical implementation. Section III focuses on the interoperability testing, presenting each implementation (ESA, NASA, CNES), but also discussing the overall testing infrastructure. Section IV is concerned with lessons-learned and will outline a potential test bed proposal for future interoperability testing setups. Finally, Section V concludes the paper and provides an outline of future work.

II. The Space Data Link Security (SDLS) Protocol

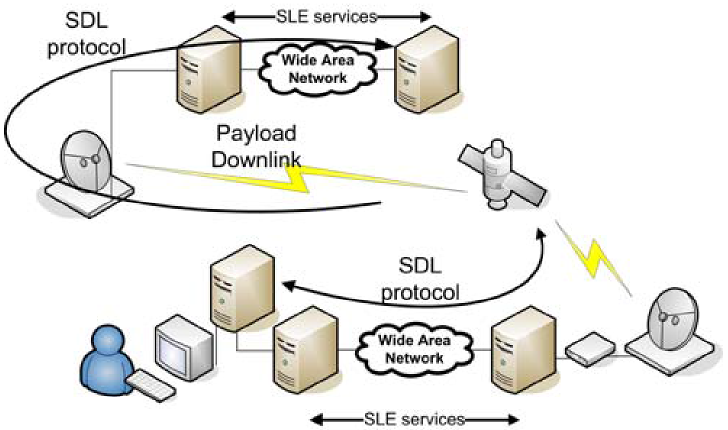


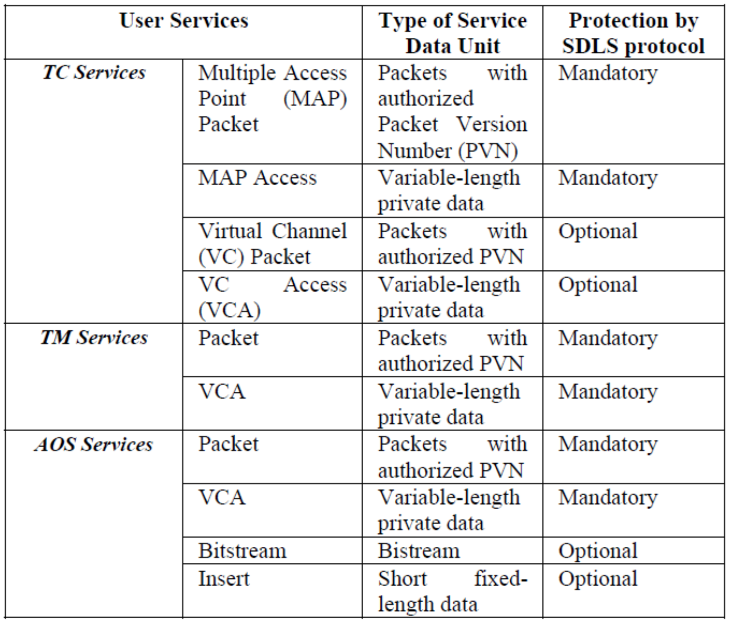
Figure 1. Space-Link Topology. *The Figure illustrates a simple space-link topology as is used for most space missions today. The upper section depicts the payload data distribution and the lower part the housekeeping and commanding segment.*

Currently, there are no security standards for the bulk of civilian space missions where there is a single spacecraft in contact with its control center through a ground station. This topology is illustrated in Fig. 1. In such as case, if uplink command authentication and/or downlink payload data confidentiality is required, up to this point each mission has had to invent its own solution. As a result, space agencies realized that a standardized concept to integrate security on space missions with a simple network topology could be developed for use at the data link layer, thus avoiding individual project ad-hoc solutions and delivering the benefits of standardization. The Consultative Committee for Space Standardization (CCSDS) specified that such a standard should be usable with existing CCSDS telecommand (TC), telemetry (TM), and advanced orbiting systems (AOS) SDL standards without modification. The aim of the link layer security standard development is to allow security services to be used with TC, TM, and AOS without forcing any reengineering of those standards, which are in wide use by many missions and planned for many upcoming missions.

Following an identification and analysis of requirements and constraints, major goals and drivers and some implementation issues, a Space Data Link Security (SDLS) protocol has been developed. The SDLS protocol implements an additional security sub-layer tightly integrated between the Data Link and Network layers of the International Standards Organization Open Systems Interconnection (ISO/OSI) model.

A. SDLS – Requirements

The main requirements of the SDLS protocol are to protect the services offered by the CCSDS SDL protocols. An overview on which user services are protected (mandatory or optional) is provided in Fig. 2. In terms of security objectives, the SDLS protocol supports (for both commanding and telemetry data) three security services: Authentication, Encryption and Authenticated Encryption.



User Services		Type of Service Data Unit	Protection by SDLS protocol
TC Services	Multiple Access Point (MAP) Packet	Packets with authorized Packet Version Number (PVN)	Mandatory
	MAP Access	Variable-length private data	Mandatory
	Virtual Channel (VC) Packet	Packets with authorized PVN	Optional
	VC Access (VCA)	Variable-length private data	Optional
TM Services	Packet	Packets with authorized PVN	Mandatory
	VCA	Variable-length private data	Mandatory
AOS Services	Packet	Packets with authorized PVN	Mandatory
	VCA	Variable-length private data	Mandatory
	Bitstream	Bistream	Optional
	Insert	Short fixed-length data	Optional

Figure 2. Protected SDL user services.

B. SDLS – Key Drivers

The following key drivers had a direct impact on the design of the SDLS protocol:

1) Compatibility with SDL protocols: A minimum impact, ideally none, to existing SDL protocols is sought. In fact, a good modularity between the existing implementations of the SDL protocols (TM, TC and AOS) and the SDLS protocol is envisaged, thus limiting the impacts of inserting security protocol in existing TM/TC ground infrastructure and on-board equipment.

2) Compatibility with SLE services: The currently defined and specified SLE services rely on their ability to identify and process SDL frames or parts of it for further processing and transfer between the SLE end points. The application of security services (e.g., confidentiality) at the SDL protocol with the new SDLS may impact the SLE service ability to 'read' and process the SDL frames, and in turn, the SLE service compatibility with SDLS. This impact is sought to be minimized.

3) Modularity : The SDLS protocol shall offer modularity in selecting security services in accordance with the risk specifics and management decision of a given flight project. In order to support the application of such modularity guidelines will be provided in choosing security services in an accompanying document.

4) Algorithm Independence: The CCSDS has established recommendations for cryptographic algorithms[8]. The SDLS protocol will support those recommended algorithms but should also provide sufficient flexibility to allow the incorporation of operator-specific cryptographic algorithms or future algorithms replacing the currently recommended due to obsolescence This is particularly important for authentication where length of message authentication codes (MACs) may evolve in order to cope with increasing threats.

5) Interoperability: Interoperability plays a key role in CCSDS work. In some missions the Launch and Early Orbit Phase (LEOP) operations are outsourced to other specialized spacecraft operators. Once the LEOP phase is concluded the satellite is handed over to its main operator. The adoption of a standardized SDLS may ease the implementation of secure LEOP operations. Furthermore, there are other mission scenarios where a satellite developed by agency A is routinely operated by agency B from its own control center.

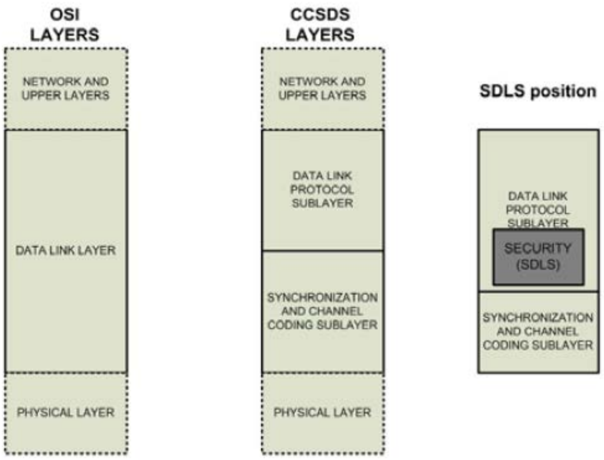


Figure 3. OSI vs. CCSDS layers and SDLS security functions position.

C. SDLS – Main Concepts

In the following, we will outline the main design concepts that were implemented in the SDLS protocol.

1) Security Association: The concept of Security Association (SA), borrowed from IPSec[7] but somewhat adapted to space communications, is crucial to the SDLS protocol. The selected security services for SDLS are implemented

with cryptographic algorithms and functions. The SDLS protocol provides SAs for defining the cryptographic parameters to be used by both the sending and receiving ends of a communications session, and for maintaining state information for the duration of the session. The SA defines a simplex (one-way), stateful cryptographic session for providing authentication, data integrity, replay protection, and/or data confidentiality. All Transfer Frames that share the same SA on a physical channel constitute a Secure Channel. Once an SA is created, the authentication and/or encryption algorithms specified, along with their modes of operation, are fixed and cannot be changed for the duration of the SA.

2) Protocol Position in the CCSDS stack: The objective of the SDLS protocol development is to add a security function at the data link layer of space links using either one of the CCSDS space data link protocols. The relation of CCSDS protocol layers with OSI (Open Systems Interconnection model of ISO) layers, together with position of SDLS security functions are depicted in Fig. 3. Two sub-layers of the Data Link layer are defined for CCSDS space link protocols: data link protocol sub-layer, and synchronization & channel coding sub-layer. SDLS protocol and functions are part of the CCSDS data link protocol sub-layer and fully integrated in the TC, TM and AOS data link protocols. SDLS functions insert themselves inside the stack of functions of CCSDS data link protocols. SDLS protocol is not as such a distinct sub-layer but rather a set of additional security features for existing data link protocols. Each of those data link protocols provides a set of communication services. SDLS protects only part of those services as shown in Fig. 2.

3) Protocol Data Structures: The SDLS encapsulates application-layer data carried in Space Data Link Protocol transfer frames between a Security Header and Trailer. The Security Header and Trailer contain the contextual information necessary to perform decryption and/or integrity verification at the receiving end. This contextual information does impose some additional transmission overhead; the sender must ensure that the overall length of the transfer frame does not exceed the maximum allowed by the underlying Space Data Link Protocol. The amount of overhead will depend upon the options chosen for each Security Association.

III. SDLS Interoperability Testing

In this paragraph, which constitutes our main contribution, we describe the process of the interoperability testing that was applied for the SDLS protocol. We start with discussing the testing requirements and argue why it is necessary to define a baseline configuration for the protocol in order to be able to perform the testing. We will then introduce the three different and independent SDLS implementations that have been created by three agencies before describing the overall interoperability testing setup, execution and results.

A. Overall Testing Requirements

CCSDS requires the testing campaigns for new standard to comply with a few minimal requirements. In particular, it requires at least two independent implementations of the protocol that are able to communicate and work with each other. In addition, the SDLS working group has specified a number of additional requirements for the SDLS interoperability testing campaign. The aim of the test campaign is to validate the completeness, correctness and interoperability of the SDLS protocol.

It is critical that the testing is representative and thus that it is executed in an “operational-like” environment or test bed. Since the SDLS protocol is fully embedded in three different types of space-link protocols (TM, TC, and AOS), it is necessary to test all three of them with SDLS. It is also important to validate that the SDLS protocol implementation does not interfere or interact with the transmission error control procedures (e.g. COP-1[9]). As such, the injection of transmission and security errors during test execution is necessary, thus further increasing the complexity of the test setup.

The test execution could be performed in a single laboratory by one test bed using multiple independent implementations of SDLS protocol. However, optimally the testing should be conducted between the reference implementations of different agencies.

In the following, we first describe the SDLS baseline mode, which is the common SDLS configuration used by all implementations. Then we present the three different reference implementations that have been implemented by ESA, NASA, and CNES. Finally, we describe the overall interoperability testing architecture, the test execution and the results.

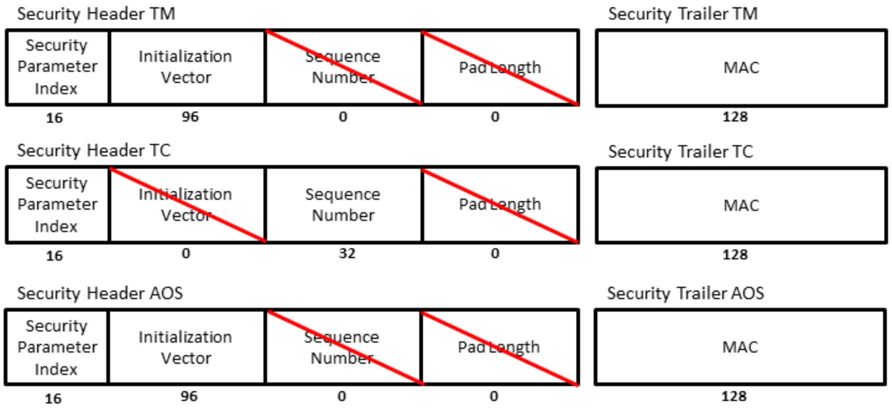


Figure 4. Security Header and Trailer configurations for SDLS baseline mode. *The Figure shows the configurations of the security header and trailer according to the SDLS baseline mode for TM, TC, and AOS data-link layer protocols.*

B. SDLS Baseline Mode

One of the key design drivers (see Section II.B) for the SDLS protocol is flexibility and algorithm independence. This means that the protocol can be operated with the recommended CCSDS algorithms[8] or with other encryption, authentication, and authenticated encryption algorithms. This flexibility makes testing of the SDLS protocol very difficult and with all possible combinations certainly impossible. For this reason, it was agreed to define a so called Baseline Mode (See Appendix E of the SDLS standard[11]), which represents the suggested/standard configuration of the SDLS protocol. This means it is recommended to use the protocol in this configuration. In addition to testing, this also eases interoperability which is one of the design drivers of the protocol. In the following we will describe the baseline modes that have been established for SDLS use which each of the underlying data-link layer protocols. All three baseline mode configurations are implementing the Advanced Encryption Standard (AES) with different modes of operation as specified in the CCSDS crypto algorithms book[8]. The security header and trailer configuration for all three modes in shown in Fig. 4.

For TM, the chosen AES mode is Galois/Counter Mode (GCM) which provides Authenticated Encryption. The specific configuration is as follows:

- The key length is 128 bit,
- The input initialization vector is 96 bit, where all of these bits are transmitted in-line in the Initialization Vector field of the Security Header,
- The output Message Authentication Code (MAC) is 128 bit long.

For TC, the chosen AES mode is Cipher-Based Message Authentication (CMAC) which provides Authentication. The specific configuration is as follows:

- The key length is 128 bit,
- The anti-replay sequence is 32 bits long, where all of these bits are transmitted in-line in the Sequence Number field of the Security Header,
- The output MAC is 128 bit long.

For AOS, the chosen AES mode is GCM (Authenticated Encryption). The specific configuration is as follows:

- The key length is 128 bit,
- The input initialization vector is 96 bit, where all of these bits are transmitted in-line in the Initialization Vector field of the Security Header,
- The output MAC is 128 bit long.

C. Interoperability Test Cases

In order to satisfy the testing requirements, the SDLS working group has devised a number of test cases that need to be executed and successfully completed as part of the interoperability testing campaign[10]. Each of the test cases defines also the configuration parameters for the two SDLS implementations that are used. This includes for example the SA setup, authentication bit mask settings (mask used to select additional authenticated data in the Transfer Frame Header), anti-replay window size and others. The following test cases have been devised:

1) Test Case 1: SDLS Protocol Validation over TC SDL protocol: This test case validates the SDLS specification for TC in the baseline mode. The first part of this test consists of the validation under nominal conditions. For this,

standard test telecommands are generated, protected as per baseline mode configuration, and then send to the (simulated) spacecraft. Following this, also non-nominal TC security processing is tested. In particular, telecommands with manipulated MACs, telecommmands that violate the anti-replay counter window, and replayed commands are injected. The SDLS protocol shall be able to detect these anomalies and ground-space configuration mismatches.

2) Test Case 2: SDLS Protocol/COP-1 Compatibility Check: It is important to ensure that the SDLS protocol is not interfering with other protocols that have similar, but different functionalities. In particular, this applies to the COP-1 protocol (in charge of detecting transmission errors & retransmitting frames if errors detected). The purpose of this test case is to validate that no interference exists between the SDL TC/COP-1 protocols and the SDLS protocol (in charge of detecting security errors). To achieve this, a number of errors (both transmission and security) are injected into protected TC transfer frames. In order to pass the test, these errors should be successfully detected by the right protocol instance (SDLS for security errors or COP-1 for transmission errors). Also, it needs to be guaranteed that e.g. a security error does not have an impact on the COP-1 protocol instance and vice versa.

3) Test Case 3: SDLS Protocol Validation over TM SDL protocol: This test case validated the SDLS specification for TM in the baseline mode. It contains nominal telemetry frames as well as injected mistakes to understand whether the implementation is able to correctly detect exceptions and ground-space configuration mismatches. Also, this test case validates the possibility to maintain a mix of secure (SDLS protected) virtual channels and non-protected virtual channels. This means that multiple virtual channels will be used at the same time, some protected, some not. This test will help to identify possible interference.

4) Test Case 4: SDLS Protocol Validation over AOS SDL protocol: This test case validated the SDLS specification for AOS in the baseline mode. It contains nominal AOS frames as well as injected mistakes to understand whether the implementation is able to correctly detect exceptions and ground-space configuration mismatches.

D. ESA SDLS Implementation Prototype

The ESA test environment for the SDLS protocol is part of the SpaceSecLab at ESAs Operations Center (ESOC) and implements an end-to-end communication chain between the mission control system and the spacecraft. This environment is fully representative, except that instead of a real spacecraft an operational simulator is used to implement the space segment part of the system. All subsystems that are usually used in an ESA mission operations infrastructure have been used for the SDLS testing as well. Fig. 5 provides an overview of all the subsystems used in the test bed and the interfaces they are using.

In this test bed, telecommands are generated by ESA's Mission Control System (MCS), SCOS-2000. The commands are then routed to the security unit which applies SDLS security services (confidentiality, authentication, or authenticated encryption). The security processed telecommands are sent back to the MCS, where they finish the encoding process. They are then sent to the Network Interface System (NIS), which uses the Space Link Extension (SLE) services protocol[6] to further communicate the encoded commands to the Telemetry and Telecommand System (TMTCS), which in a real setup would be located at the ground station. The TMTCS would interface with the ground station systems and ensure radiation of commands to the spacecraft. For the simulated spacecraft environment however, the TMTCS is simply forwarding the commands to the simulator instance (GSTVi). The GSTVi interfaces with the onboard security unit to apply again the necessary security functions. The same processing chain applies to telemetry, the difference being that the source is the spacecraft and the sink is the mission control system. Since only frame-level SLE services are used (F-CLTU, R-CF, R-OCF, and R-AF), the

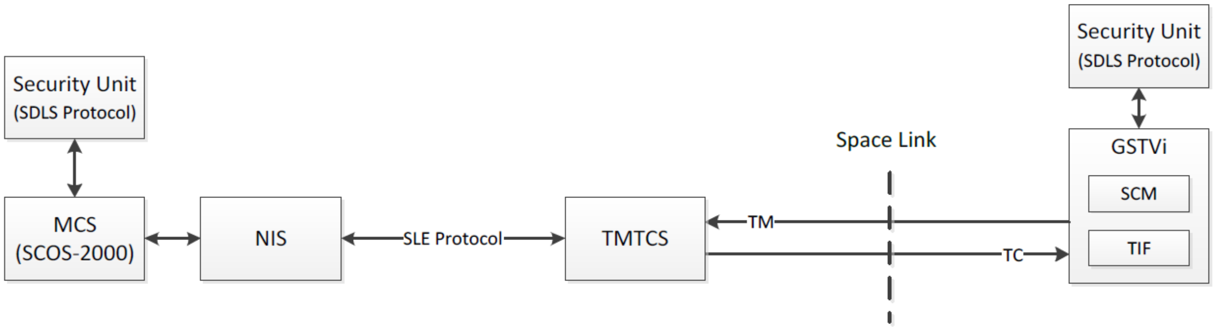


Figure 5. ESA end-to-end SDLS test bed.

security services are completely transparent to intermediate entities such as the ground station.

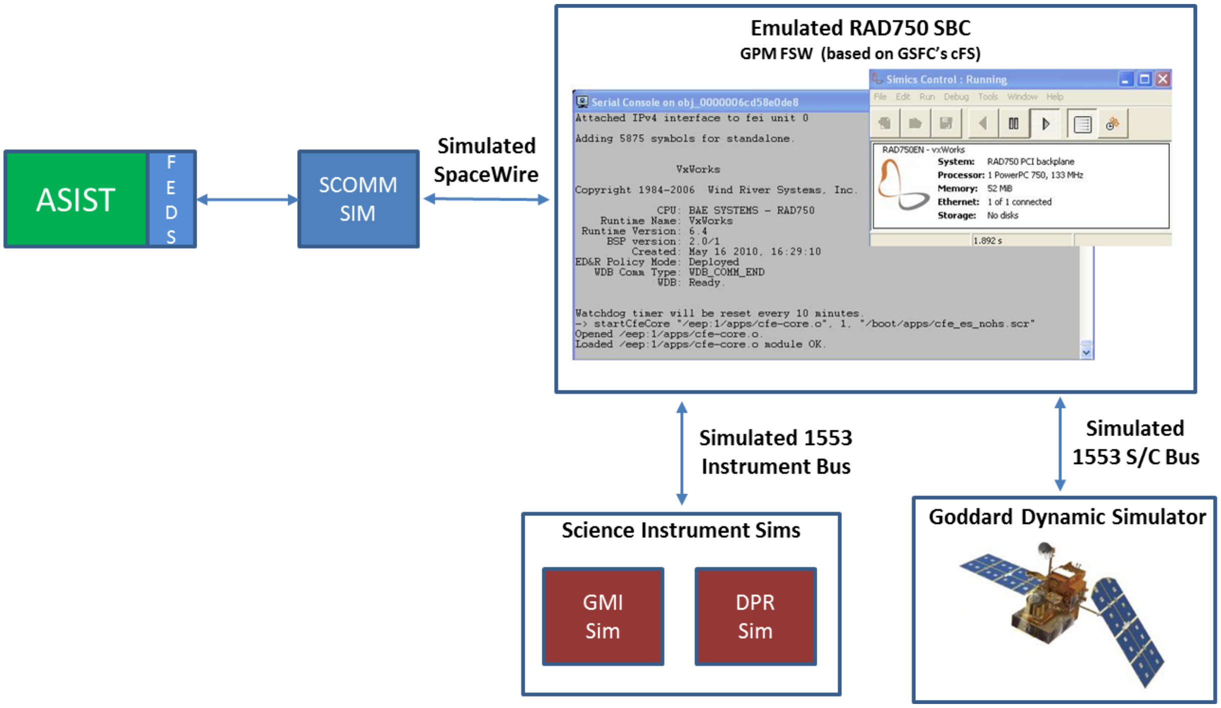
The security unit implementation does not only process the data, it is also managing the security associations i.e. the secure space-link channels and their configuration. Settings such as cryptographic key in use, crypto algorithm, authentication bit mask, etc. can be configured here. This makes it easy to configure an end-to-end SDLS setup in the absence of standardized procedures for security association and key management (such procedures will be introduced in the CCSDS SDLS Extended Procedures Blue Book which is currently under development). Furthermore, also test tools are included. The security unit test configuration allows injecting a variety of errors that the protocol should be able to detect. Examples include an invalid anti-replay counter value, a MAC mismatch, etc.). All error test cases that are specified in the SDLS Test Report[10] are supported.

In order to support interoperability testing, the current version of the ESA reference implementation can be injected with recorded binary TM/TC SDL frames from another reference implementation. Security association and key management configuration information has to be agreed beforehand. More information is provided in Section III.G below when we discuss the inter-agency testing.

ESA is currently in the process of upgrading its SDLS test environment to allow online interoperability testing in the context of the cloud deployment (see Section IV.B) and also to implement the draft SDLS Extended Procedures for a number of initial tests.

E. NASA Test bed

NASA's test bed for SDLS was developed at the John McBride Software Testing and Research (JSTAR) Laboratory within NASA's Independent Verification and Validation (IV&V) Program in Fairmont, West Virginia. The JSTAR Lab contains simulations for many of NASA's current missions and JSTAR contains the domain knowledge, expertise, and infrastructure to support ground and spacecraft simulations. JSTAR provided an ideal environment to implement the SDLS prototype and support the SDLS interoperability testing.



```
Attached IPv4 interface to fei unit 0

Adding 5875 symbols for standalone.


                 VxWorks

Copyright 1984-2006  Wind River Systems, Inc.

               CPU: BAE SYSTEMS - RAD750
      Runtime Name: VxWorks
   Runtime Version: 6.4
       BSP version: 2.0/1
           Created: May 16 2010, 16:29:10
ED&R Policy Mode: Deployed
     WDB Comm Type: WDB_COMM_END
               WDB: Ready.


Watchdog timer will be reset every 10 minutes.
-> startCfeCore "/eep:1/apps/cfe-core.o", 1, "/boot/apps/cfe_es_nohs.scr"
Opened /eep:1/apps/cfe-core.o.
Loaded /eep:1/apps/cfe-core.o module OK.
```

RAD750EN - vxWorks

System: RAD750 PCI backplane
Processor: 1 PowerPC 750, 133 MHz
Memory: 52 MB
Ethernet: 1 of 1 connected
Storage: No disks

Figure 6. Overall NASA Prototype Architecture.

NASA's approach for developing the prototype for testing the SDLS protocol was to reuse existing NASA operational ground and flight systems/simulators. Using existing systems provided two benefits: reduction in time to implement (due to software reuse) and applicability after prototype was completed. Upon completion of the prototype, future NASA missions will be able to take the lessons learned from the prototype and apply them to their mission when implementing SDLS.

The GPM Operational Simulator (GO-SIM) was the starting point for NASA's development of a SDLS prototype. GO-SIM is a pure software based simulator that uses the GPM ground system (ASIST), ground system command and telemetry databases, emulated RAD750 Single Board Computer (SBC), and unmodified GPM flight software binaries. GO-SIM is not a wall clock "real time" model. There is no hardware (e.g., SpaceWire, MIL-STD1553) in the loop. The flight software communicates the TM/AOS protocols over a simulated SpaceWire network and MIL-STD-1553 busses. Even though events are marked and seem to be running from the model in "real time," the simulation may run faster or slower (depending on the level of activity) relative to wall clock real time.

In an effort to reduce code duplication, a simple cross-platformsecurity library was developed to be used in both the ground and spacecraft systems. Due to limited memory on the spacecraft hardware, the library was to be kept as compact as possible.

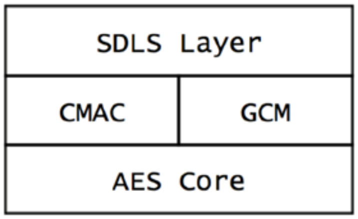


Figure 6. NASA Test Bed Security Library Structure.

Based on Annex E of the SDLS standard[11], Baseline Implementation Mode, the library supports the following capabilities to support the modes for each of the Space Data Link Protocols:

- Advanced Encryption Standard (AES) algorithm.
- Cipher-Based Message Authentication Code (CMAC) for authentication,
  - Based on AES algorithm with 128-bit keys,
  - 32-bit Anti-Replay Sequence Number,
  - 128-bit output Message Authentication Code (MAC).
- Galois Counter Mode (GCM) for authenticated encryption.
  - Based on AES algorithm with 128-bit keys,
  - 96-bit Initialization Vector,
  - 128-bit output MAC.

The base library will be composed of three layers:

- AES Core
  - The AES Core layer will provide the forward and reverse encryption procedures, supporting 128-bit keys.
- CMAC and GCM Layers.
- SDLS Layer
  - Contains support for SDLS-related constructs (e.g., Security Association (SA), Authentication Masks).

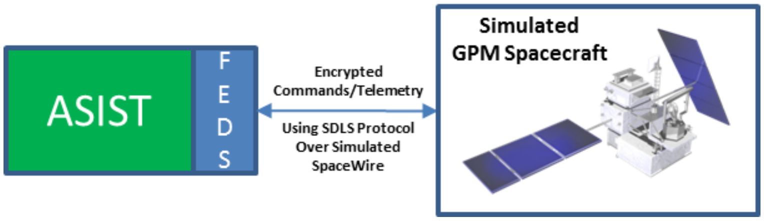

ASIST
F
E
D
S
Encrypted Commands/Telemetry
Using SDLS Protocol Over Simulated SpaceWire
Simulated GPM Spacecraft

Figure 7. Overall NASA Prototype Space-Link.

See Fig. 6 for a graphical representation of the security library structure.

The security library was integrated into both the ground software (ASIST) and spacecraft flight software (FSW). The data (telemetry and telecommand) will be exchanged through the TC and AOS protocols in real time. Telecommands will be generated via the ground system and passed through the security library thereby creating protected telecommands to be uplinked via the SCOMM simulator. The incoming data can then be analysed by the FSW using the security library and processed. Telemetry, sent from the spacecraft to the ground, functions the same way but in reverse. The FSW encrypts the telemetry using the security library and downlinks via the SCOMM simulator to ASIST, which is then decrypted and processed by the ground system.

F. CNES Test bed

Two different simulators have been developed by CNES. The first one was developed in house, at the on-board data processing laboratory. Written in C, it runs on a standard PC with Windows environment. It is able to simulate SDLS over TM or AOS links. It is split in four independent software modules, each module simulating one side

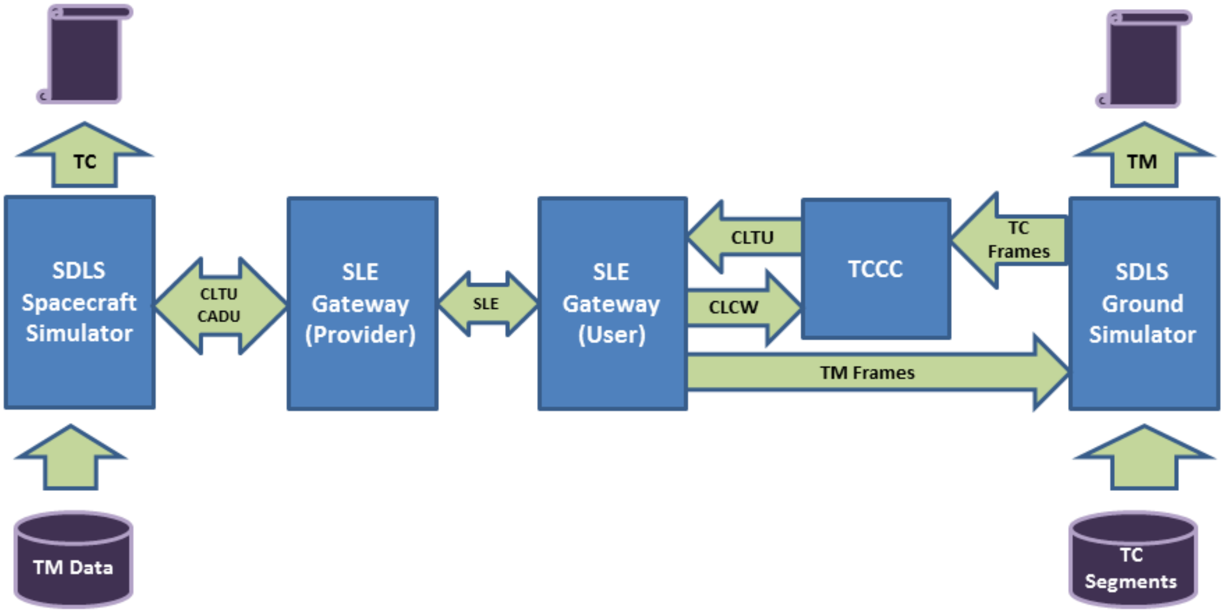


Figure 8. CNES SDLS Prototype Infrastructure.

(Ground or Spacecraft) of the selected link (TM or AOS). The stream of data representing the TM or AOS link is exchanged between the Ground module and the Spacecraft module either by file exchange or UDP/IP link. Configuration files are used for setting all the SDLS parameters like key, authentication mask, etc. The simulator can be run in local mode, simulating both ends of the SDLS protected TM or AOS link by simultaneously activating the Ground and Spacecraft software modules. Alternatively, the simulator can be used to model one side of the link, with only one module activated, the other side of the link being simulated by another agency's simulator. This configuration was used for testing the TM link with ESA's simulator (CNES simulating the Spacecraft side) and AOS link with NASA's simulator (CNES simulating the Ground side, as depicted in Fig. 9).

The second simulator was developed by a subcontractor under CNES supervision. Its purpose is to simulate the SDLS over TC and TM simultaneously. Running on a PC with linux environment, it is built with already developed software modules used in CNES operational satellite control centers, like TC generation and formatting, FOP-1 engine (ground part of the COP-1 protocol) and SLE gateways. Modules were specifically developed for SDLS and spacecraft simulation. Like ESA's simulator, the ground side implements a complete TC and TM chain, including SLE protocol. This has shown to be useful for testing that SDLS has no side effect to the SLE protocol (operationally used between the control center and the ground stations). Being able to simulate the TC and TM link simultaneously, it also allows for testing the COP-1 protocol in closed loop (retransmission requests of TC frames with errors detected being sent to ground via the TM link). Exchange of data between the Ground side and the Spacecraft side is done using files or SLE services. This SLE feature makes the Ground side see the spacecraft side through SLE services, exactly like in operational control centers. This feature could be used for connecting the simulator (Ground or Spacecraft side) to a distant simulator via Internet, doing "real-time" simulations. Having all the "Bells and Whistles", this simulator is a bit tricky to configure, requiring fine setting of parameters in many configuration files. The Ground side part of this simulator has been used to test the SDLS over TC protocol. The overall setup is depicted in Fig. 8.

G. Overall Test Architecture and Execution

Following the completion of the local implementation and testing of the SLDS protocol at ESA, NASA, and CNES sites, the next step was to set up a cross-agency test bed to actually validate the interoperability of the SDLS implementations. CCSDS standards are only published after successful execution of such interoperability tests as per agreed test plan (See Section III.C above).

One critical aspect is the exchange of data between the different test beds. In the case of SDLS, this includes configuration settings, for example related to the security association in use but also the binary TM, TC, and AOS frames themselves. Following a feasibility analysis, it was decided not to go for a direct on-line connection of the implementation test beds. The main reason for this was the many hurdles that a connection to an external network has to overcome. Thus, in order to avoid a long approval process for this and thus delay the publication of the SDLS standard, it was agreed to exchange the data using an offline channel. Secure email transfer proved to be the most practical approach in this respect. In all tests, the original unsecured frames are compared are saved into text files, send along via secure email and then compared with the finally processed frames at the other end of the communication chain with the aim to validate that they are identical.

The interoperability setup is split into two main test setups between different Agencies. ESA and CNES have tested the SDLS protocol based on the TM and TC SDL protocols[3,4]. The ESA reference implementation plays the role of the ground segment. It prepares protected telecommands for injection into the CNES implementation, playing the role of the space segment, to validate test cases 1 and 2. The same test is inverted for test case 3 (SDL TM protocol). The second setup between the CNES and NASA implementations has been prepared to validate test case 4 (SDL AOS protocol). The CNES implementation was selected to represent the ground segment side and the NASA implementation was used as space segment. Fig. 9. shows the overall setup.

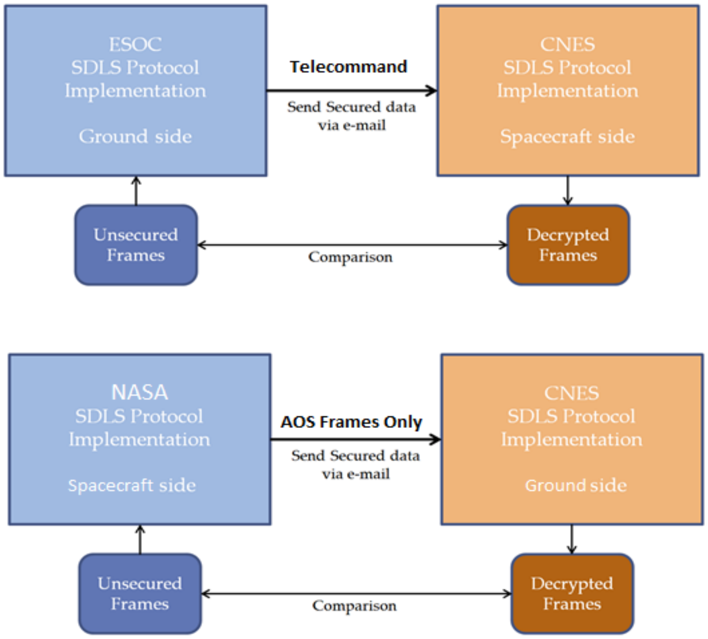


Figure 9. Overall Test Architecture. *The first diagram shows the CCSDS TC part of the test architecture between ESA and CNES while the second diagram shows the AOS part of the architecture between NASA and CNES.*

In all test cases, the processing of the SDL transfer frames between the three different implementations has been successful. Furthermore, all deliberately injected faults (e.g. bit flip, wrong anti-replay counter value) were handled according to the specification. No interference with other protocols such as COP-1 could be identified. As a result, the interoperability testing campaign has been declared successful.

IV. Lessons-Learned

A. Technical Improvements

While the interoperability testing was successful and the SDLS protocol baseline mode has been validated, a number of issues have been identified and will be considered as technical improvements for the future. One key aspect certainly is the limited flexibility of the offline connection of the test environments. A solution for this could be exploiting a cloud-based environment. Section IV.B describes in detail this possible evolution.

Another aspect that was noticed during the execution of the test was the inability to run a large number of automated test cases. The executed test cases were enough to validate the protocol, however the capability to perform automated testing would have increased the confidence in the protocol. The exchange of configuration information has been noted to be inflexible, also due to the offline channel. However, in general, the test environment should be capable of changing the configuration settings on both receiver and sender side in a fast and user friendly way. This would have allowed further test cases beyond the baseline mode. Part of this will be addressed with the implementation of the SDLS Extended Procedures but even then a simple user interface to do this will prove very useful.

B. Cloud-Based Interoperability Testing

The success of testing the baseline mode of the SDLS protocol was not dependent on real-time connection between the participating agencies. The interoperability testing success criteria were met even though the transfer frames had to be emailed from one agency to another. This however will not be a feasible approach for future testing campaigns (i.e. SDLS extended procedures). In the past it has proven to be difficult performing point-to-point interoperability testing between two agencies due to the multiple layers of approval required to implement firewall change requests or create new VPN tunnels. A potential solution to alleviate these networking issues is to perform the interoperability tests in "the cloud". For testing the SDLS protocol outside of the baseline mode, real-time communications between the agencies will be required. While the setup of such a testing environment brings a number of challenges, once in place, it could significantly simplify the interoperability testing not only for security protocols but for all CCSDS protocols. Several approaches have been discussed for cloud based testing of CCSDS protocols but the participating agencies involved with SDLS testing (NASA, ESA, and CNES) are investigating the possibility of creating a specific SDLS cloud environment. Upon success, other CCSDS protocol implementers will be able to apply the lessons learned and approaches used for SDLS testing. Two approaches are being investigated for a SDLS specific implementation. The preferred approach is to utilize the same cloud provider but different virtual machines connected by their own private virtual network. The contingency approach is for each agency to deploy on their own separate cloud providers and perform the testing over public IP space. See Fig. 8 for a graphical representation of the two approaches. The conclusions to be drawn from performing the initial SDLS cloud pilot will provide a basis for the SDLS working group, and other CCSDS groups, to make an informed decision on the feasibility of using the cloud to perform interoperability of other CCSDS protocols/standards.

V. Conclusion

C. Conclusion

In this paper, we described the Space Data-Link Security (SDLS) protocol and the security services it provides to enable the protection of space-links that are based on CCSDS TM, TC, or AOS protocols. We introduced the three different prototype implementations done by CNES, ESA, and NASA and their respective testing environments before describing the interoperability test and the test execution between these implementations. All four test cases of the interoperability test have been validated and the overall test has been declared successful. Thus, this necessary precondition for the publication of the SDLS protocol standard has been fulfilled.

D. Future Work

This paper focuses on the interoperability testing that has been executed for the baseline mode of the SDLS core protocol. The CCSDS SDLS working group is already in the process of preparing the SDLS Extended Procedures which will provide and formalize the auxiliary procedures necessary to successfully run the SDLS protocol. More concrete, it will specify procedures for security association management, key management, and security unit management. It will also define a new Operational Control Field for the TM and AOS frames that serves as a security unit reporting mechanism. The interoperability testing and validation of these extended procedures is much more challenging than for the core protocol. The main reason here is that this can only be done in an online setup. Also, it requires much more rigorous confidence testing to be sure that no interference with the main data-link layer protocols is created. The upgrade of the various testing environments to support SDLS Extended Procedures interoperability testing will be one central element of future work together with taking on-board some of the lessons-learned identified earlier.

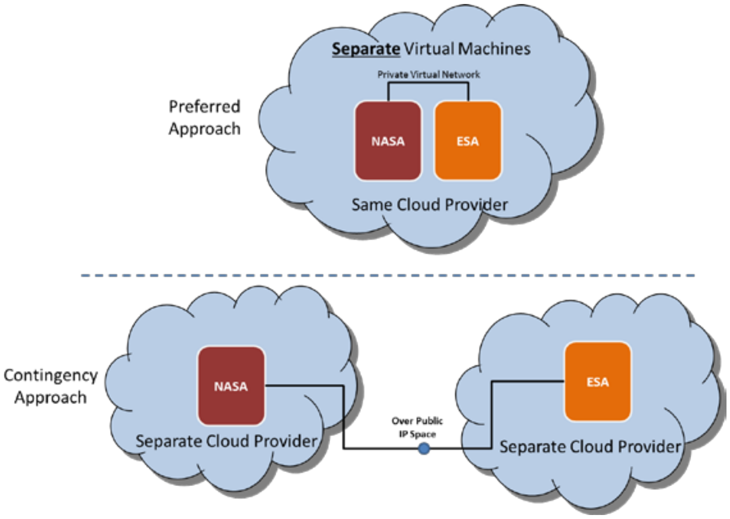


Figure 10. Cloud Based Interoperability Testing. *The diagram shows the two possible cloud setups that are considered as a baseline for future interoperability testing.*

References

[1] I. Aguilar-Sanchez, D. Fischer, "The CCSDS Space Data Link Security Protocol - Design Aspects" *5th ESA International Workshop on Tracking, Telemetry and Command Systems for Space Applications, Noordwijk, The Netherlands, 2010*

[2]I. Aguilar-Sanchez, C. Biggerstaff, D. Fischer, G. Moury, B. Saba, H. Weiss, "Towards Completion of the CCSDS Space Data Link Security Protocol," *2012 IEEE Aerospace Conference*

[3] CCSDS 232.0-B-1, "TC Space Data Link Protocol", CCSDS Recommended Standard, September 2003

[4] CCSDS 132.0-B-1, "TM Space Data Link Protocol", CCSDS Recommended Standard, September 2003

[5] CCSDS 732.0-B-1, "AOS Space Data Link Protocol", CCSDS Recommended Standard, July 2003

[6] CCSDS 910.0-G-2, "Space Link Extension Services – Executive Summary", CCSDS Green Book, March 2006

[7] RFC 4301,"Security Architecture for the Internet Protocol", The Internet Society, December 2005

[8] CCSDS 352.0-B-1, "CCSDS Cryptographic Algorithms", CCSDS Recommended Standard, November 2012

[9] CCSDS 232.1-B-2, "Communication Operations Procedure-1", CCSDS Recommended Standard, September 2010

[10] CCSDS 355.0-Y-1, "Space Data-Link Security (SDLS) Protocol Test Report", CCSDS Record, March 2015

[11] CCSDS 355.0-B-1, "Space Data-Link Security Protocol", CCSDS Recommended Standard, November 2015