NATO Consultation, Command and Control Agency
Agence de Consultation, de Commandement et de Conduite des Opérations de l'OTAN

LINK1 FORWARD FILTER (L1FF)
SECURITY TARGET (Public Version)

Wim Hoekstra / Peter Rehäußer

06-02-07
The Hague

Document information

Date of issue	06-02-07
Author(s)	Wim Hoekstra / Peter Rehäußer
Version number report	1.13
Certification ID	BSI-DSZ-CC-0342
Scheme	BSI (Germany)
Sponsor	NATO C3 Agency
Sponsor address	Oude Waalsdorperweg 61 2597 AK, The Hague The Netherlands
Evaluation Lab	CSC Ploenzke AG
Evaluation Lab address	CoE IT Security and Technology Sandstr. 7-9 80335 Munich Germany
Target of Evaluation (TOE)	Outbound Downgrade Filter of ASDE Link-1 Forward Filter version 1.5
TOE reference name	ASDE-L1FF
CC-EAL number	4
File Name	ST_Public.doc

Document history

Version	Date	Comment
1.0	28-08-03	Formal release to NC3A. Reviewed by NC3A/NOS.
1.1	23-11-04	Update to cover comments from the evaluator
1.2	28-06-05	Update to fulfil the CC requirements
1.3	08-08-05	Update to cover comments from the evaluator
1.4	02-09-05	Update due to some errors in the narrative description
1.5	05-10-05	Update due to comments from the certification body
1.6	10-11-05	Update to cover comments from the evaluator
1.7	09-01-06	Update to cover comments from the certification body
1.8	16-03-06	Update due to changes in the software
1.9	31-03-06	Update to cover comments from the evaluator
1.10	12-04-06	Update to cover comments from the evaluator
1.11	29-05-06	Update to due changes on the software
1.12	06-11-06	Final Version
1.13	06-02-07	Public Version

Contents

DOCUMENT INFORMATION **2**

DOCUMENT HISTORY **2**

1. SECURITY TARGET INTRODUCTION **6**

1.1 ST IDENTIFICATION 6
1.2 ST OVERVIEW 6
1.3 CC CONFORMANCE 7

2. TOE DESCRIPTION **8**

2.1 OVERVIEW 8
2.2 DEFINITION OF THE TOE AND ITS SECURITY SERVICES 11
2.3 UNDERLYING IT PLATFORM 13
2.4 PHYSICAL BOUNDARIES OF THE TOE AND SCOPE OF DELIVERY 13
2.5 LOGICAL BOUNDARIES OF THE TOE 14

3. TOE SECURITY ENVIRONMENT **19**

3.1 DEFINITION OF SUBJECTS, OBJECTS AND OPERATIONS 19
3.2 ORGANISATIONAL SECURITY POLICIES (P) 24
3.3 ASSUMPTIONS (A) 25
3.4 THREATS (T) 26

4. TOE SECURITY OBJECTIVES **28**

4.1 SECURITY OBJECTIVES FOR THE TOE (SOT) 28
4.2 SECURITY OBJECTIVES FOR THE ENVIRONMENT (SOE) 30

5. IT SECURITY REQUIREMENTS **32**

5.1 TOE SECURITY FUNCTIONAL REQUIREMENTS 33
5.2 STRENGTH-OF-FUNCTION CLAIM 47
5.3 TOE SECURITY ASSURANCE REQUIREMENTS 48
5.4 SECURITY REQUIREMENTS FOR THE IT ENVIRONMENT 49

6. TOE SUMMARY SPECIFICATION **50**

6.1 TOE SECURITY FUNCTIONS 50
6.2 ASSURANCE MEASURES 65

7. PP CLAIMS **68**

8. RATIONALE **69**

8.1 SECURITY OBJECTIVES RATIONALE 69
8.2 SECURITY REQUIREMENTS RATIONALE 73
8.3 TOE SUMMARY SPECIFICATION RATIONALE 82
8.4 PP CLAIMS RATIONALE 87

9. APPENDIX A - ABBREVIATIONS 88

10. APPENDIX B - REFERENCES 89

11. APPENDIX C - GLOSSARY OF TERMS 90

12. APPENDIX D - LINK-1 FORWARD FILTER SANITIZATION RULES 91

13. APPENDIX E - THE NEED OF AN EVALUATION 92

List of figures

Figure 1: ASDE system consisting of a Buffer, Forward Filter and diodes. 8
Figure 2: Logical boundaries of the TOE of classified and unclassified data. 15
Figure 3: Overview of the Security Functions of the communication functionality and their relation. 51
Figure 4: Overview of the Security Functions of the filter functionality and their relation. 52

List of tables

Table 1, Assurance requirements for the TOE. 48
Table 2, Environment to Objectives. 69
Table 3, Objectives to SFR. 74
Table 4, Objectives for the IT Environment to SFR for the IT Environment. 77
Table 5, SFR to TSF. 82

1. Security Target Introduction

1.1 ST Identification

Name of the TOE:	Outbound Downgrade Filter of ASDE Link-1 Forward Filter version 1.5
ST Version:	1.13
Keywords:	Trusted guard

This Security Target is produced by the NATO Consultation, Command and Control Agency (NC3A) in response to security requirements of the NATO Office of Security (NOS). The production, registration and certification of a valid Security Target is a mandatory pre-requisite to NC3A achieving approval by NOS to permit operation of a computer-based system that will act as an automated and trusted guard between classified and unclassified IT enclaves to prevent the accidental leakage of classified information.

Comments on the current Security Target should be sent to either the NATO C3 Agency, P.O. Box 174, 2501 CD The Hague, The Netherlands, or to NATO Office of Security, NATO HQ, Brussels, Belgium.

1.2 ST Overview

The Outbound Downgrade Filter of ASDE Link-1 Forward Filter version 1.5 (L1FF) is a software application of an Air Situation Data Exchange (ASDE) that will permit one-way Link-1 message streams to be securely and automatically screened for the contents considered to be classified within a trusted and secure environment (typically a transmitting NATO facility such as a Control and Reporting Centre (CRC), see [SRS]). The screening rules applied depend upon a mode of operation related to times of either peace or differing levels of crisis.

The Link-1 Forward Filter aims at downgrading sanitized outbound CLASSIFIED[1] Link-1 Messages into NATO UNCLASSIFIED/Partner Nations RELEASABLE Link-1 Messages. When classified messages are encountered, the content of these messages will not be transmitted. When Link-1 message fields containing information considered to be classified are encountered, the bits in those fields will be set to zero before the message itself will be transmitted. The Link-1 Forward

[1] The connotation CLASSIFIED is used here and throughout this document to cover all classification levels compliant with the EAL-4 accreditation sought.

Filter sends the downgraded and sanitized messages out over unencrypted and unprotected serial communications lines.

The Link-1 Forward Filter can also be used to verify that the Link-1 data received from the Partner Nations equals the Link-1 format but this is not a function under evaluation.

The Link-1 Forward Filter runs mandated on a secure and certified operating system, that is served by an accompanying hardware platform, which is located in a secured location, that can only be accessed by authorised personnel who have been 'screened' as a condition of their employment by NATO.

1.3 CC Conformance

The evaluation is based upon:

- Common Criteria for Information Technology Security Evaluation, Version 2.3, part 1: General model.
- Common Criteria for Information Technology Security Evaluation, Version 2.3, part 2: Security functional requirements.
- Common Criteria for Information Technology Security Evaluation, Version 2.3, part 3: Security assurance requirements.
- Common Methodology for Information Technology Security Evaluation, Version 2.3: Evaluation Methodology.

The chosen level of assurance is:
EAL4 (Evaluation Assurance Level 4)

This Security Target claims the following conformances for the TOE:

- **CC Part 2 conformant**
- **CC Part 3 conformant**
- **No conformance to any PP**

2. TOE Description

2.1 Overview

NATO peacetime and crisis response operations could result in an operational requirement to use the airspace of Partner Nations. Exchange of air situation data between a Partner Nation (PN) and NATO is only allowed when no sensitive data are exchanged. Sufficient measures shall be implemented to ensure sensitive NATO information is safeguarded at the required level of security.

Part of the safeguarding is the IT system called Air Situation Data Exchange (ASDE). ASDE is a program that allows sharing of portions of the NATO Recognized Air Picture (RAP) with approved PNs [MCM140]. The other part of the safeguard measures is mandated by the regulations of NATO for IT boundary devices or cryptographic devices [NATO-SP]. These regulations consist of physical, personnel, organisational and procedural measures.

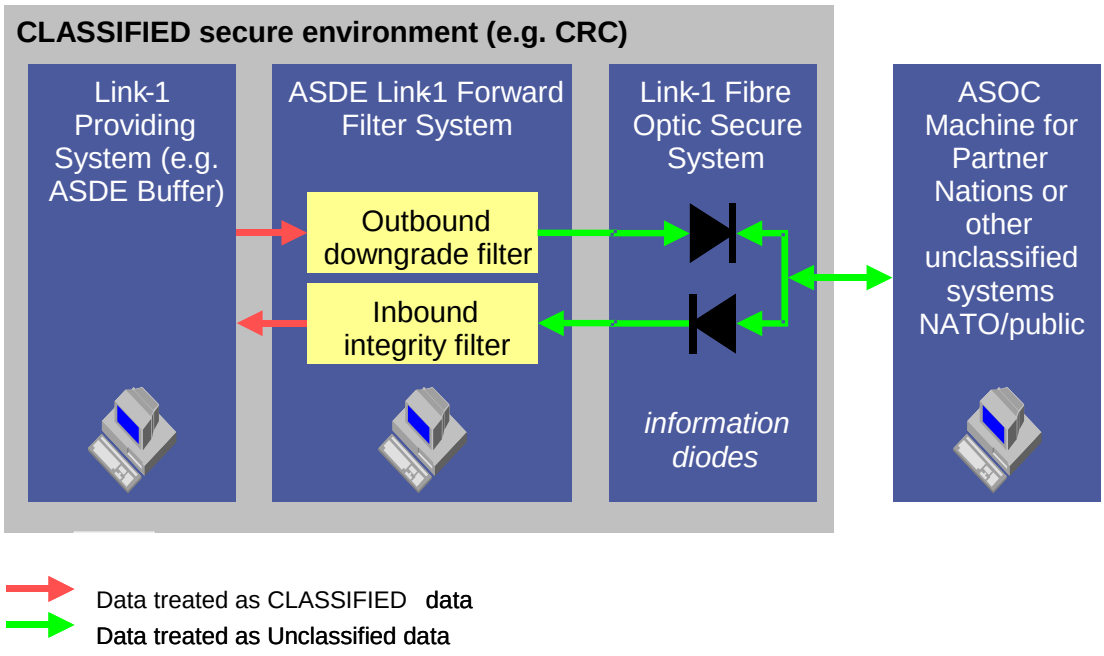

Figure 1: ASDE system consisting of a Buffer, Forward Filter and diodes.

The data to be shared with partner nations will come from the NATO Air Command and Control Centre (CRC) or another Link-1 provider[2] and will take the form of short data messages in Link-1 format that will be transmitted from a CRC to the designated non-NATO air operations centre[3]. Link-1 messages are bit strings that are generated within many NATO and national IT systems from real-time air asset related data. The messages have a fixed format, which is defined in [STANAG5501], and contain a variety of information of which only a very small percentage is classified. The majority of the data within Link-1 messages is unclassified and is suitable for dissemination to persons who do not have clearance. The elements considered classified may never be transmitted beyond the limits of the protected NATO enclave.

The ASDE is located in a CRC or equivalent secure facility and consists of the following four physically separated parts (see Figure 1):

- *ASDE Buffer or other Link-1 Providing System:* This part of the ASDE system executes the normal procedures as required for exchange of RAP information with any other Link-1 site and also implements the mandatory rules as defined by SHAPE[4] for the exchange of information with non-NATO Link-1 sites. The buffer is the primary source of Link-1 data for the ASDE Link-1 Forward Filter.

- *ASDE Link-1 Forward Filter:* This part of the ASDE system consists of the filter functionality between two environments with a different classification. The outbound downgrade part of the L1FF gets its input from the ASDE Buffer, but another Link-1 input source is not excluded[5]. The inbound integrity filter part receives Link-1 messages from Partner Nations. The filter is a hardware and software system that allows the filtering of Link-1 data and consists of:

    - *Outbound Downgrade Filter.* This filter is a software application that allows the filtering of Link-1 data messages to prevent that unauthorized data is sent out. The filter is a trusted guard, i.e. an automated NATO program that allows one-way passage of automatically screened, unclassified and non-sensitive Link-1 data over serial communication lines from an inner, protected and sensitive enclave of NATO IT systems to an external non-NATO enclave where uncleared and untrusted users, IT systems and networks operate. This filter is mandated for outbound messages.

---

2 Here, the providing organisations are in mind, not IT systems.

3 Here, the facilities/sites respectively the organizations are in mind, not IT systems.

4 SHAPE = Supreme Headquarters Allied Powers Europe.

5 If the Link-1 Forward Filter is not working in connection with the ASDE Buffer, it will not be able to support exchange of Link-1 messages; it merely passes sanitized Link-1 output to a Link-1 recipient.

- *Inbound Integrity Filter*. This filter is a software application concerned with track data sent by non-NATO nations from their own system to be included in the NATO RAP. These messages will be in Link-1 format. The track data must pass an integrity check to ensure that the NATO RAP is not corrupted accidentally or maliciously with data of non-NATO origin. This filter is optional for inbound messages.

- *Link-1 Fibre Optic Secure System* (LIFOS): This part of the ASDE system consists of information diodes that ensure the flow of serial line data in one direction only. Using this device, covert backdoor entry to the L1FF via the serial line used for Link-1 message output is securely denied.

2.2 Definition of the TOE and its security services

The TOE is the Outbound Downgrade Filter of ASDE Link-1 Forward Filter version 1.5.

The TOE consists of two software applications. The testframe part is the actual filter. The operator console is the interface to the user and controls the filter.

LIFOS information diodes are an already evaluated product and the Link-1 Providing System will not be subject to accreditation. This Security Target defines the claim for the accreditation.

TOE primary security service: Downgrade
The TOE offers one primary security service:

Downgrade CLASSIFIED Link-1 Messages into NATO UNCLASSIFIED/PN RELEASABLE Link-1 Messages.

This security service shall assure that Link-1 messages that are downgraded do not contain any other information than NATO UNCLASSIFIED / PN RELEASABLE. Any message, where it is not certain that it is NATO UNCLASSIFIED / PN RELEASEABLE shall not be sent out (transmitted) by the TOE.

TOE supporting security services
To support the primary security service of the TOE, the TOE offers three security services:
1. Filtering,
2. Filter management,
3. TOE integrity check.

Filtering
This supporting security service aims to assure that a message does not contain any other information than NATO UNCLASSIFIED/PN RELEASABLE. Therefore, this service consists of two functions to assure that messages only contain NATO UNCLASSIFIED/PN RELEASABLE information:
- Completely blocking the content of certain messages[6];
- Zeroizing certain bit fields in messages that are not blocked.

[6] For these types of messages, the classified message will completely be replaced by a blank message. The classified information is blocked. A blank message will be send to allow full traceability of the sanitization process.
The blank message is considered as classified until the message is downgraded according to the standard procedure.

These two functions are known as “Sanitization”.
Details about which messages are blocked and what message fields are zeroized are provided in [Rules]

The filtering functions are executed by applying a fixed rule set that is mandated by NATO regulations. The rules that define which (or parts of) Link-1 messages are authorized to be downgraded depend on the mode of operation of the Link-1 Providing System. The contents of the rule set differ for each mode of operation. There are four distinct operational modes:

1. Peace Operational Mode,
2. Exercise Operational Mode,
3. Crisis Response Operational Mode,
4. Article 5 Operational Mode.

[Rules] lists all rules for all the operational modes.

The rule set of the filtering function to be applied on Link-1 messages conforms to [STANAG5501].

Filter management
This supporting security service is concerned with a number of activities that require management:

- *Operational mode change*: When the operational mode (see downgrade function) on the Link-1 Providing System is changed, the mode on the TOE must be changed accordingly. A time to switch from the current operational mode to another operational mode must be agreed upon with the receiving party.
- *Audit management:* Generation and preserving of audit logs for pre-defined security relevant events.

TOE integrity check
This supporting security service assures that the integrity of the TOE is not violated. The integrity check relates to the following:

- *Downgraded data*: The TOE defines and verifies the checksum over a Link-1 Message before this message is sent out.
- *TOE program*: The TOE performs a test to check whether its code or the rule set has been changed.

2.3 Underlying IT Platform

A trusted IT Platform will host the TOE. The TOE runs on a secure evaluated IT Platform. This IT Platform contains an operating system[7] certified at EAL4. The operating system will be used with the underlying hardware as described in the Security Target of the operating system [ST-Solaris].

The operating system is conformant with the following registered Protection Profiles[8]:

- Controlled Access Protection Profile, Issue 1.d, 8 October 1999;
- Labelled Security Protection Profile, Issue 1.b, 8 October 1999;
- Role Based Access Control Protection Profile, Issue 1.0, 30 July1998.

2.4 Physical Boundaries of the TOE and Scope of Delivery

The TOE consists only of software. Therefore the TOE itself has no physical boundaries. Nevertheless, the following (physical) components build up the scope of delivery and therefore the physical boundaries.

The scope of delivery including the TOE is:

- 1 Sun Microsystems Sparc machine (SunBlade 150) (Hardware)
- 2 SATURN AURORA PCI cards (Hardware)
- 2 AURORA breakout boxes (Hardware)
- 1 SCSI interface card (Hardware)
- 1 4mm DAT recorder (Hardware)
- 1 19” Monitor (Hardware)
- 1 LIFOS with power supply (Hardware)
- 1 Female-Female DB25 gender changer (Hardware)
- 1 Padlock for LIFOS (Hardware)
- 1 L1FF LIFOS connector cable (Hardware)
- Tags for the LIFOS and AURORA breakout box interfaces (Hardware)
- Sun Microsystems Trusted Solaris 8 12/02 (Operating System)
- Testframe part of the Outbound Downgrade Filter of ASDE Link-1 Forward Filter version 1.5 including the configuration file (Software)

---

7 The secure operating system is Sun Microsystems Trusted Solaris 8 4/01. The underlying hardware is the Sun Microsystems Blade 100/150 computer or a Sparc II. Sun ‘s replacement is the Sun Blade 150. The operating system, Trusted Solaris is not yet accredited for the Sun Blade 150. Until this accreditation is obtained the development will proceed using the Sun Blade 100 as the target platform for the TOE.

8 These Protection Profiles can be found via www.commoncriteriaportal.org

- Operator Console of the Outbound Downgrade Filter of ASDE Link-1 Forward Filter version 1.5 including the configuration file (Software)
- Libraries (Software)
  - libaul1ser.a, Version 1.0
  - libgnarl-5.03.so, Version 5.03a
  - libgnat-5.03.so, Version 5.03a
  - libgcc_s.so.1, Version 3.4.4
- System Installation Manual (Guidance)
- System-specific Security Requirements Statement (Guidance)
- Security Operation Procedures (Guidance)
- System User Manual (Guidance)

2.5 Logical Boundaries of the TOE

The logical boundary of the TOE is defined by the interfaces in its series of cooperating software applications. The TOE processes data received through these interfaces and modifies it according to various processing rules before forwarding the data to another component via another interface.

The TOE has the following external interfaces:

- Link-1 Providing System to the testframe part of the TOE,
- User interface to the Operator Console,
- Two external interfaces between each part of the TOE and the loopback device of the operating system. These interfaces build up a logical interface between the two parts of the TOE which is shown in Figure 2 as arrow between the TOE parts.
- Testframe part of the TOE to the LIFOS-information diode,
- Trusted Operating System to the both parts of the TOE (File System; handles both, classified and unclassified data).

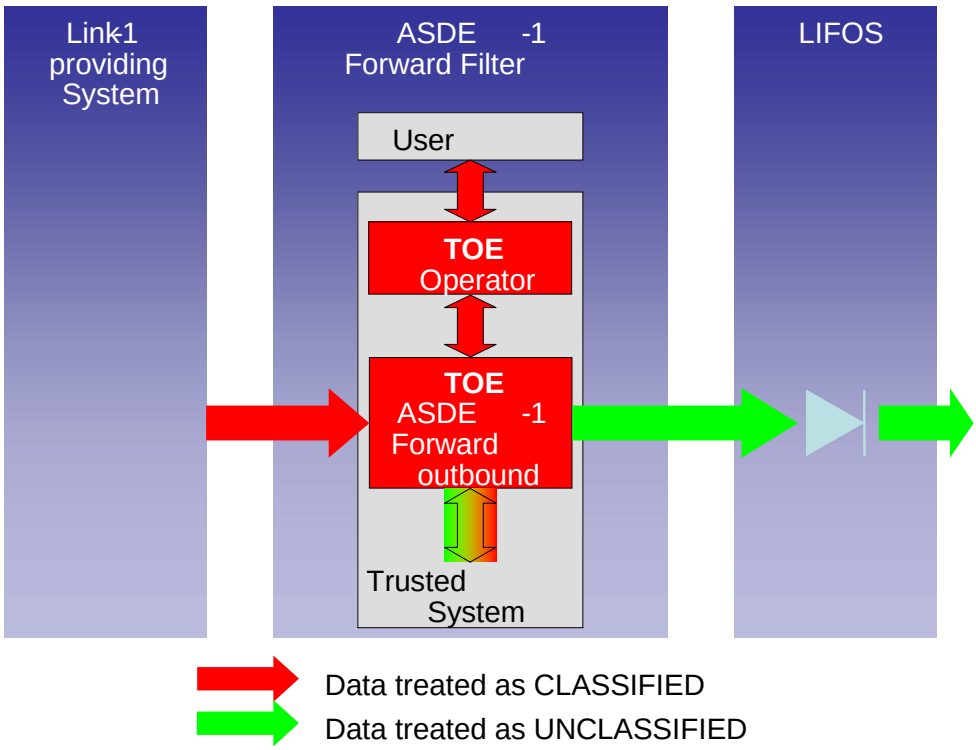

Figure 2: Logical boundaries of the TOE of classified and unclassified data.

2.5.1 Link-1 Providing System – Testframe part of the TOE

The TOE has one interface with the Link-1 Providing System:
This interface provides a bit stream from the Link-1 Providing System to the TOE. Normally, this stream contains Link-1 Messages conforming to [STANAG5501]. However, due to many causes this need not be the case, i.e. the incoming bit stream may contain Link-1 Messages that are not conform to [STANAG5501] or the incoming bit stream does not consist of Link-1 Messages. All input from the Link-1 Providing System is considered to be CLASSIFIED.
The TOE receives the bit stream by using a hardware driver (not part of the TOE) to access the network hardware. Therefore, the external interface of the TOE is actually a pure software interface.

2.5.2 Operator Console – Human User

The Operator console provides an interface to the users of the TOE. Usually, this interface is represented by keyboard and monitor. In normal operation, the Operator Console provides only warnings to the user. Warnings are displayed between two red lines and accompanied by an audible signal. Furthermore, the user is able to manage the TOE by entering special commands.
The Operator Console is implemented as window on the screen, not as hardware terminal. Therefore, this external interface of the TOE is actually a pure software interface.

The only user allowed to interact with the TOE is the operator. Other users have only a supporting role in the environment of the TOE, for example the system administrator, security officer or auditor.

2.5.2.1 Commands

The user can only provide the following single keystroke commands from console keyboard to the operator console:

- *Operational Mode.* The current operational mode is continuously displayed in the header of the Operator Console window. Mode change is performed by entering the digit `1` (Peace Operational Mode), `2` (Exercise Operational Mode), `3` (Crisis Response Operational Mode) or `4` (Article 5 Operational Mode). A time to switch from the current operational mode to another operational mode must be agreed upon with the receiving party.
- *Status display*. The keystroke '`m`' or '`M`' enables the operator to monitor the status of the TOE. This includes the current mode of operation, the number of warnings received since the last mode change, the amount of disk space used for the auditing record.
- *Audit display.* Via the keystroke '`r`' or '`R`', the user obtains information on the Journal-files which are currently stored. This means a list of file names and their sizes will be displayed.
- *Stop/Restart Filter*. The user can stop the transmission of message in case of an emergency by entering the keystroke '`s`' or '`S`'. The user can restart the application manually once the emergency situation is cleared.
- *Interrupt Recording.* The user can interrupt the recording with the keystroke '`i`' or '`I`'. In this case the current Journal file which contains the audit trail of the filter action is closed and a new Journal file is started.
- *Serial port statistics.* The user can display the serial port statistics, i.e. the number of messages and errors transmitted and received on each serial port by means of the keystroke '`l`' or '`L`'. This command resets the statistic of these two serial ports.
- *Input/Output display*. The user can toggle the display of input and output message frames by toggling the keystroke '`v`' or '`V`'. If this function is enabled, the operator sees all input and all output message frames on the screen. Otherwise, no messages will be displayed on the screen.
- *Help:* The user may display a help page on the console screen which lists and explains all available commands. This can be reached by pressing '`h`', '`H`' or '`?`'.
- *Configuration:* The user may display the configuration of the TOE on the screen. This configuration includes the content of the configuration file as well as the name of the current audit file and the current mode of operation. This information can be displayed by pressing '`c`' or '`C`'.
- *Operator Exit:* The user can exit from the operator console by pressing '`x`'. This will only terminate the operator console but not the testframe part of the TOE. This means that messages will be filtered. The journal file handling is not affected by end of the operator console.

Additionally, the operator may enter “Y” or “N” as valid command during start-up of testframe or the operator console in order to approve the CRC checksum of the respective part of the TOE.
Beside that, the operator may enter any other input but this will not affect the operator console or the testframe application.

2.5.2.2 Response to warnings

A warning informs the operator of an unexpected situation. Generally, a user response is not required. In case of a warning the operator has to respond in line with standing operating procedures. This varies between ‘do nothing’ and ‘switch off the Link-1 Forward Filter or the Link-1 Providing System’.

2.5.3 Operator Console – Testframe part of the TOE

From a logical point of view, this is a TOE internal interface. Physically, two separate applications talk together using a defined protocol and a defined communication media.
The communication media is a network interface. Due to the fact that a remote administration of the filter software is not allowed and must not be able, this is always the loopback interface provided by the operating system.
The communication protocol is proprietary to the TOE.
This means, both external interfaces are pure software interfaces.
The testframe part of the TOE receives commands and keep-alive messages from the Operator Console. The Operator Console receives the echo messages of the testframe part.

2.5.4 LIFOS-information diode – Testframe part of the TOE

This interface ensures that Link-1 messages travel in one direction only, from the TOE to the LIFOS. All messages passing from the TOE to the LIFOS-information diode are sanitized Link-1 messages conforming to [STANAG5501] and these messages shall be classified as NATO UNCLASSIFIED/PN RELEASABLE.
The TOE sends the messages by using a hardware driver (not part of the TOE) to access the network hardware. Therefore, the external interface of the TOE is actually a pure software interface.

2.5.5 Trusted Operating System –TOE

There are several points of contact between the parts of the parts of the TOE and the operating system. This section describes all these interfaces at once.[9]

[9] The logical internal interface between the two parts of the TOE is described in chapter 2.5.3 and is not covered here.

The TOE runs on a trusted operating system The TOE uses the following security objectives of this secure operating system [ST-Solaris]:

1. *Authorisation*. Only authorized users can gain access to the TOE and its resources.
2. *Mandatory Access Control*. The TOE and its users are provided with the means of controlling and limiting access to objects and resources, based on sensitivity labels and categories of the information being accessed and the clearance of the subject attempting to access that information in accordance with the NATO policy for declassification of information (see P.DECLASSIFICATION_POLICY in section 3.2).
3. *Audit*. The TOE uses the means of recording any security relevant events to:
   a. assist an administrator in the detection of potential attacks or miss-configuration of the TOE security features that would leave the TOE susceptible to attack; and
   b. hold users accountable for any actions they perform that are relevant to security.
4. *Residual Information.* Any information contained in a protected resource is not accessible when the resource is recycled.
5. *Management.* Support is provided to aid users in managing the TOE and its security functions, and it must ensure that only authorized users are able to access such functionality.
6. *Duty*. The TOE uses the capability of enforcing separation of duties so that no single user (program or human) performs all administrative functions.
7. *Hierarchical.* The TOE uses the hierarchical definitions of profile rights defined by the OS.
8. *Role.* The TOE uses the measures to prevent users (programs and humans) from gaining access to and performing operations on its resources and objects unless they have been granted access by the resource or object's owner or have been assigned a rights profile or role, which permits those operations.

TOE stores the audit records (journal file) on the hard disk of the computer using the usual operating system interfaces. These log files will be created, filled, stored and closed by the TOE.
Therefore, the external interface of the TOE is actually a pure software interface.
The TOE creates the audit records and adds the current date and time to each of them before the records are stored in the journal file. The date and time will be provided by the underlying operating system.
The access to the journal file will be restricted and controlled by the operating system and managed by the system administrator.

3. TOE Security Environment

In this Chapter the security characteristics of the environment in which the TOE is deployed are defined.

3.1 Definition of subjects, objects and operations

To facilitate easy definition of threats, organisational security policies, assumptions, security objectives and security requirements, the subjects, objects and operations to be used in the ST are defined first.

3.1.1 Non-human Subjects

The systems (equipment) that interact with the TOE are:

L1-Provider	Link-1 Providing System (or equivalent system such as an ASDE Buffer) that supplies a Link-1 Stream to the TOE. The L1-Provider is located in an IT environment with the same regime as the TOE, which is authorised to process CLASSIFIED information.
LIFOS	Accredited hardware system consisting of information diodes that ensure the flow of serial line data in one direction only. LIFOS is connected to the TOE and to a non-NATO system, which is expected to follow similar rules as within the NATO establishment, be is not under NATO control. LIFOS is located in an IT environment that is authorised to contain NATO crypto equipment.
Secure_IT_Platform	Certified secure IT Platform on which the TOE runs, consisting of a secure operating system and accompanying hardware. The secure software is the SUN Trusted Solaris 8 12/02 operating system. The hardware comprises the SUN Blade/SPARC 100/150 and serial communication cards (see footnote 7).

3.1.2 Authorized human subjects

The only user that interacts with the TOE is:

S.SysOper	User role defined by Secure_IT_Platform. This role is the operator of the TOE and is allowed to start and stop the TOE (both parts) via the Console. In addition, the role may start and stop the system, allocate system resources such as disks, start and stop queues, etc.

The users that are present within the TOE environment are:

S.Audit	User role defined by Secure_IT_Platform. This role is the Auditor of the audit output of the TOE and of audits in the TOE IT environment. Only the S.Audit role can analyse, back up and restore system audit logs when the testframe part of the TOE is not running. The audit logs are regularly reviewed.
S.ISSO	User role defined by Secure_IT_Platform. This role is the Information System Security Officer of the TOE IT environment. Only the S.ISSO role can create new user accounts and establish or change security related settings like contents of the label encoding file, user clearance limits, etc. At least two on-site named persons shall always be allocated to this role.
S.SysAdmin	User role defined by Secure_IT_Platform. This role is the system administrator of the TOE IT environment. S.SysAdmin shall undertake normal UNIX administration duties such as maintaining user passwords, etc. S.SysAdmin is the only role able to modify user accounts, but cannot create new accounts. No user able to operate in the S.SysAdmin role shall also have the possibility to operate in the S.ISSO or S.Audit role. At least two on-site named persons shall always be allocated to this role.

S.SysOper, S.Audit, S.ISSO and S.SysAdmin are all authorised to access the IT environment of the TOE. Authorisation is settled conform to NATO regulations. These persons are characterized as follows:

- Competent to perform their duties;
- Able to perform the appropriate security procedures;
- Have an appropriate screening of at least the site level of accreditation;
- Are trusted not to abuse his authority;
- Are trusted not to compromise security measures;
- Are not considered to be hostile;
- Are capable of making mistakes (although not intentionally).

Security Attributes of Subjects

SA.Oper_Mode	This security attribute defines the four possible operational modes of the L1-Provider and the TOE. • Peace Operational Mode • Exercise Operational Mode • Crisis Response Operational Mode • Article 5 Operational Mode
SA.OS_MAC_Level	This security attribute defines the four mandatory access control operational levels of the Secure_IT_Platform[10]. These levels are (from highest to lowest classification): • Admin high, Classified (i.e. CLASSIFIED), • Unclassified (i.e. NATO UNCLASSIFIED/PN RELEASABLE), • Software, • Admin Low.
SA.OS_Priv_Level	This security attribute defines the privileges (*privileged* or *unprivileged*) to determine if a subject may execute a trusted system call, or a general system call of the Secure_IT_Platform in a trusted manner (i.e., file write with MAC override). SA.OS_Priv_Level is independent of SA.OS_MAC_Level.
SA.Subject_Identity	Associated security attribute for a subject that equals the name of the subject, i.e. L1-Provider and LIFOS.

[10] All authorized human subjects have a SA.OS_MAC_Level defining in which operation level they are allowed to operate:
- S.SysOper, S.Audit, S.ISSO operate at SA.OS_MAC_Level ‘Admin high, Classified’
- S.SysAdmin operates at SA.OS_MAC_Level ‘Admin Low’.

3.1.3 Objects

For all objects the following security attribute holds:

SA.Security_Label	This security attribute defines the two classification levels that data processed by the TOE and its environment can have. The classification levels are CLASSIFIED and NATO UNCLASSIFIED/PN RELEASABLE.

The (data) objects for the TOE that the TOE will operate upon are:

O.Data_Audit	Audit data log record produced by the TOE. The data has SA.Security_Label 'CLASSIFIED.
O.Data_Class	A packet of data having a sequence number and SA.Security_Label 'CLASSIFIED'. The packet can take the following forms: 1. *Bit stream:* Series of bits that are probably a Link-1 message. 2. *Link-1 Message:* Link-1 Message as defined by [STANAG5501]. 3. *Sanitized Link-1 Message*: Link-1 Message sanitized by the operation R.Sanitize (see section operations).
O.Data_Unclass	A sanitized O.Data_Class having SA.Security_Label 'NATO UNCLASSIFIED/PN RELEASABLE'.
O.Filter_Rule_Set	The set of rules that define which (parts of) O.Data_Class need to be sanitized given by the SA.Oper_Mode of the L1-Provider. The set of rules is listed in Appendix D - Link-1 Forward Filter Sanitization Rules of this ST. The set has SA.Security_Label 'CLASSIFIED'.
O.Command	Messages send from the operator console to the testframe part of the TOE. These messages contain commands for the testframe entered by the user at the operator console.
O.Ping	A special O.Command the operator console sends regularly to the testframe. This informs the testframe that the operator console is running.
O.Output_Message	Messages send from the testframe part of the TOE to the operator console. These messages contain information the operator console has to display.

3.1.4 Operations

R.Audit_Trail	This operation writes O.Data_Audit to an audit trail of the Secure_IT_Platform.
R.CRC_Check	This operation confirms or denies whether the cyclic redundancy check of O.Data_Unclass equals the cyclic redundancy check calculated by R.CRC_Pack for the corresponding sanitized O.Data_Class.
R.CRC_Pack	This operation calculates a cyclic redundancy check over a sanitized O.Data_Class and the cyclic redundancy check is added to this sanitized O.Data_Class.
R.Disregard	This operation disregards all data in O.Data_Class or O.Data_Unclass.
R.Downgrade	This operation generates a new O.Data_Unclass with the data of a sanitized O.Data_Class.
R.Sanitize	This operation applies O.Filter_Rule_Set on O.Data_Class. This means this operation generates a new O.Data_Class that contains a [STANAG5501] compliant Link-1 Message which fulfils O.Filter_Rule_Set (some bits are zeroed or a blank message).
R.Set_Mode	This operation sets the O.Filter_Rule_Set to one of the SA.Oper_Mode values.
R.Test	This operation checks the integrity of the TOE and the presence of the Secure_IT_Platform.
R.Verify_Outbound	This operation confirms or denies whether O.Data_Class coming from L1-Provider conforms syntactically to [STANAG5501].

3.1.5 Non-Authorized subjects (Threat Agents)

The following subjects are capable to effectuate threats for the TOE (i.e. Threat Agents):

TA.Erroneous_User	S.SysOper, S.Audit, S.ISSO or S.SysAdmin capable of making mistakes with organizational security policies or accidentally modifying the Secure_IT_Platform or the TOE configuration, thereby allowing security violations to occur.
TA.Unclass_Receiver	Entity, human person or IT system not authorised to receive O.Data_Class. This entity is capable of receiving an outgoing Link-1 data stream from the TOE outside the TOE environment.

3.2 Organisational Security Policies (P)

The main purpose of the TOE is to implement the NATO policy for declassification in an automated way. This is defined by P.DECLASSIFICATION_POLICY.

P.DECLASSIFICATION_POLICY
The TOE shall implement and comply with the NATO declassification policy appropriate for downgrading classified information [SRS]. This policy defines the

- *Filter rules*: the set of rules for the circumstances under which information will be allowed for declassification. In [Rules] this policy is fully defined.
- *Condition*: the condition for an automated system under which the filter rules are allowed to be applied. The condition is: It shall be retrievable when an O.Data_Class has been sent out.

P.INTER-TOE-COMMUNICATION
The two parts of the TOE shall establish a communication in such a way that

- the testframe receives all O.Command's from the operator console
- only the testframe receives the O.Command's
- the operator console receives all O.Output_Message's from the testframe
- only the operator console receives the O.Output_Message's

P.KEEP-ALIVE-POLICY

- If there is no other O.Command communication the operator console must send an O.Ping message to the testframe every 10 seconds.
- The testframe must be able to work without a running operator console but for three (3) minutes maximum.
- After this period of time the testframe has to stop working. This means, O.Data_Class from L1-Provider must be blocked.

P.TOE_DATA_INPUT
Outbound is defined as coming from the L1-Provider to the TOE.
The TOE shall be able to handle input streams with the following characteristics: A bit stream coming from an L1-Provider can have any form and can possibly conform to [STANAG5501].

P.TOE_FAIL_INSECURE
If the testframe part of the TOE software fails, a TA.Unclass_Receiver is able to read O.Data_Class either immediately or in some point in the future because the failure results in a forwarding of unsanitized messages.
The TOE shall be able to handle failures in the hardware, in the operating system or the TOE itself in such a way that unsanitized messages will not be forwarded.

3.3 Assumptions (A)

Assumptions may be assumptions of the intended usage of the TOE (A.U) or assumptions regarding the environment of use (A.E).

A.U.ONLY_WAY
The TOE assumes that it is the only path for the O.Data_Class to be downgraded to O.Data_Unclass so it can be passed on from an L1-Provider to LIFOS.

A.E.OUTSIDE
From the outside, attacks can only be performed via a data stream from the Partner Nation. It is assumed that this data stream has to pass a LIFOS and can therefore not reach the TOE.
Therefore, it exist no possibility that incoming messages from the outside interfere with the sanitization and downgrading process.

A.E.INSIDE
It is assumed that from the inside, Link-1 messages are received from a Link-1 Provider, which is assumed to be a NATO certified system.

A.E.RECORDING
The Trusted Operating System keeps a record of all actions on the system on the level of the operating system.

A.E.NATO_SECURITY_POLICY
The NATO security policy concerning security principles, personnel security, physical security, security of information and information security (INFOSEC) is mandated for the TOE and its IT environment [NATO-SP]. The IT environment operates within a CLASSIFIED accredited facility for boundary protection devices and crypto devices. Application of the policy includes the following:

1. Logical
    a. Only authorized personnel can have access to the Secure_IT_Platform.
    b. Remote access to the Secure_IT_Platform is not allowed.
    c. All users of the Secure_IT_Platform are appropriately identified and authenticated, and have the appropriate access rights and are held accountable for their actions.
    d. No user (program or human) of the Secure_IT_Platform can unintentionally delete, overwrite or manipulate any system programs, logs, or data.
2. Organisational
    a. S.Audit shall immediately notify S.ISSO in case of any threats or vulnerability that impacts P.DECLASSIFICATION_POLICY.
    b. Information shall be used only for its authorized purpose(s).

3. Personnel
   a. The personnel who need access to the TOE or the environment running the TOE must be screened according to site accreditation requirements.
   b. S.SysOper, S.Audit, S.ISSO and S.SysAdmin shall be held accountable for their actions.
   c. Only S.SysOper, S.Audit, S.ISSO and S.SysAdmin shall be able to access O.Data_Class.
4. Physical
   a. The TOE shall be located in a physically secured room within a NATO facility accredited for the site level of accreditation.
   b. Access to this room is restricted to authorized persons listed in access lists.

A.E.TOE_ACCESS_POLICY
S.SysOper is the only user role that is allowed to interact with the TOE.

A.E.INTER-TOE-COMMUNICATION
It is assumed that the operating system does not deny a communication between the two parts of the TOE.

3.4 Threats (T)

T.BYPASS
O.Data_Class are passed from the Link-1 Providing System to the TOE. In the TOE these data are processed and recorded. After the processing these data become NATO UNCLASSIFIED/PN RELEASABLE. The O.Data_Class are only available on the interface with the Link-1 Providing System, within the TOE or from the recording (Audit_Trail).
A TA.Unclass_Receiver is able to read O.Data_Class either immediately, or in some point in the future, because TA.Erroneous_User has logically or physically bypassed the protection functions of the TOE. This may be possible due to errors in or an erroneous configuration of the underlying operating system or failures of the physical access controls to the hardware. This threat may occur at each time a TA.Erroneous_User has logical or physical access to the hardware, operating system or the TOE or when an already existing bug within the operating system becomes effect.

T.MODE_SYNC
A TA.Unclass_Receiver is able to read O.Data_Class because TA.Erroneous_User has not synchronized SA.Oper_Mode of the TOE with SA.Oper_Mode of the L1-Provider. This threat occurs when TA.Erroneous_User does not perform a required change of SA.Oper_Mode. Due to the fact that TA.Erroneous_User is allowed to change SA.Oper_Mode, only communication problems with the other L1-Provider or human failure could be the reason.

T.NEGLIGENCE
A TA.Erroneous_User makes a mistake, for instance inserting a wrong operational mode in the TOE (e.g. Exercise instead of Peace) that possibly violates P.DECLASSIFICATION_POLICY causing that TA.Unclass_Receiver is able to read O.Data_Class and S.Audit does not notice. This threat may occur when TA.Erroneous_User performs a change of SA.Oper_Mode. Due to the fact that TA.Erroneous_User is allowed to change SA.Oper_Mode, only human failures could be the reason.

T.OPERATOR_DOES_NOT_EXIT
A TA.Erroneous_User logs out of the operating system but does not exit the operator console before. This may happen because the user starts the operator console as independent process in the background or the operating system puts the process in the background during log off of the user. Therefore, this threat may occur at any time. The operator console keeps running and the time-out mechanism of the TOE testframe part does not work. Therefore, there is no human operator to monitor the warning messages the TOE generates. This may result in O.Data_Class sent out without appropriate sanitization to TA.Unclass_Receiver.

T.TOE_REPROGRAM
A TA.Erroneous_User may reprogram or modify the TOE binary stored on the hard disk, causing it to pass through O.Data_Class either immediately or in some point in the future. For this purpose TA.Erroneous_User can use the tools usually installed with the underlying operating system. This threat is possible because TA.Erroneous_User must have access to the TOE binary for his normal work and the appropriate tools are installed on the system. Due to the fact that the access to the TOE is not restricted for TA.Erroneous_User, this attack or mistake may occur every time TA.Erroneous_User works on the system.

4. TOE Security Objectives

This section defines the Security Objectives of the TOE and its environment. The Security Objectives reflect the stated intent to counter all identified threats. They comply with all organizational security policies identified and uphold all assumptions.

4.1 Security Objectives for the TOE (SOT)

The Security Objectives for the TOE are divided into the primary Security Objective and supporting Security Objectives.

Primary Security Objective: Downgrade

SOT.DOWNGRADE
The TOE shall implement the operation R.Downgrade on sanitized O.Data_Class. In order to verify the downgrade operation, the R.CRC_Pack is performed before and the R.CRC_Check after the operation. After R.Downgrade and the CRC check are performed the TOE can send the data to LIFOS.

Supporting Security Objectives

SOT.CONSIDER_LOGOUT
The operator console shall be able to recognise user logout or equivalent events[11] in order to exit in a controlled way. The operator console process must not be kept in memory (running or not) when the user is logged out. Furthermore, the operator console process must not be able to block the log out process of the operating system.

SOT.DATA_AUDIT
The TOE shall generate O.Data_Audit after performing one of these individual operations R.Verify_Outbound, R.CRC_Check, R.Downgrade, R.Sanitize, R.Set_Mode, R.Test as well as start and stop of the TOEs audit function.

SOT.DATA_EXPORT
The TOE shall perform the operation R.Audit_Trail to enable S.Audit to read O.Data_Audit generated by the TOE.

[11] When the operator console runs in an X window, closing this window is equivalent to a log out of the user (from the programs point of view).

SOT.FAIL_SECURE
A failure in the operation R.Sanitize shall not cause the TOE to pass through O.Data_Class to the operation R.Downgrade.

SOT.FILTER_RULE
The TOE shall ensure that the operation R.Sanitize uses the O.Filter_Rule_Set according to SA.Oper_Mode set by S.SysOper.

SOT.KEEP_ALIVE
The two parts of the TOE shall establish a communication in such a way that the testframe part of the TOE stops if a communication with the operator console is not possible for 3 minutes or longer. This means, after that period of time without communication (keep-alive messages) all messages from the Link-1 providing system will be blocked until the user enables the filter again.
The testframe part must have the ability to work without a communication with the operator console.
The testframe part must recognise nearly immediately when the operator console does not run. For this purpose, the operator console has to send an O.Ping every 10 seconds to the testframe.

SOT.NO_BYPASS
The TOE shall enforce P.DECLASSIFICATION_POLICY on all data that passes through the TOE from a L1-Provider to LIFOS.

SOT.NO_REPROGRAM
Changes to the integrity of the TOE shall be detected at start-up of the TOE. This includes the binary of the TOE as well as SA.Oper_Mode and O.Filter_Rule_Set. The TOE shall record this event and fail into a secure state.

SOT.NO_RESIDUAL
The TOE shall perform the operation R.Disregard to ensure that no O.Data_Class / O.Data_Unclass is available in the main memory of the underlying platform when

- one of the operations R.Verify_Outboud, R.Sanitize or R.CRC_Check has decided to reject (parts of) this data.
- the TOE is stopped.

SOT.SANITIZE
The TOE shall implement the following policy:

- The TOE shall perform the operations R.Verify_Outbound and R.Sanitize on all O.Data_Class transferred from the L1-Provider to LIFOS.
- The TOE shall have completed the operation R.Sanitize on O.Data_Class before operation R.Downgrade is performed.
- The TOE shall not change O.Data_Unclass after operation R.Downgrade is performed.

SOT.SECURE_COMMUNICATION
In order to protect the authenticity, integrity and confidentiality of the communication between the two parts of the TOE

- The configuration of the two parts of the TOE shall ensure that exactly these two programs communicate with each other.
- The configuration of the two parts of the TOE shall ensure that the TOE does not try to build up or accept connections across a network.

4.2 Security Objectives for the Environment (SOE)

The security objectives for the environment are divided into security objectives for the IT environment and the non-IT environment. One security objective is relevant for the IT and the non-IT environment.

4.2.1 Security Objectives for the IT Environment

SOE.SECURE_IT_PLATFORM
The TOE environment shall mandate that the TOE runs on the Secure_IT_Platform, having the following characteristics:

- Secure storage of O.Data_Audit,
- Restricted access to the TOE to S.SysOper,
- Enabling that operations are performed on the right SA.OS_MAC_Level and SA.OS_Priv_Level,
- Prevent the existence of residual information after a stop of the TOE
- Preserve the secure state of Secure_IT_Platform,
- Separate the logical execution of the TOE from any other program running on Secure_IT_Platform.
- Recording of all security relevant events on the level of the operating system

4.2.2 Security Objectives for the non-IT Environment

SOE.AUDIT_REVIEW
The TOE environment shall provide S.Audit with means to access and regularly review O.Data_Audit generated by the TOE as made available by SOE.DATA_AUDIT.

SOE.DATA_AUDIT
The TOE environment shall implement procedures to store O.Data_Audit generated by the TOE compliant with the accompanying manuals, including:

- Short-term storage on the Secure_IT_Platform;
- Long-term storage on a long-term storage medium.

SOE.SECURE_ENVIRONMENT
The TOE environment shall implement
- A.E.NATO_SECURITY_POLICY
- A.E.TOE_ACCESS_POLICY

SOE.SECURE_USAGE
The TOE environment shall establish and implement procedures to ensure that the TOE is installed, used and maintained compliant with the accompanying manuals. S.SysOper has to be trained to maintain the TOE in an appropriate way.

SOE.TOE_LOCATION
The TOE environment shall ensure that the TOE is the only communication path between the L1-Provider and LIFOS. No other devices than LIFOS are connected between an unclassified environment and the TOE (outer side). Only NATO certified L1-Provider shall be connected to the inner side of the TOE.

4.2.3 Security Objectives for the IT and the non-IT Environment

SOE.MODE_SYNC
The TOE environment shall have a procedure in order to keep SA.Oper_Mode of the TOE synchronised with SA.Oper_Mode of the L1-Provider.

SOE.SECURE_COMMUNICATION
The configuration of all other programs on the system and the configuration of the system itself shall ensure that no other process tries to communicate with one of the TOE applications.

5. IT Security Requirements

This section defines the IT security requirements of the TOE, consisting of:

- TOE security functional requirements (SFR). All SFRs in this ST were drawn from Part 2 of the CC.
- TOE security assurance requirements (SAR). All SARs in this ST were drawn from Part 3 of the CC.
- Security requirements for the IT environment were drawn from [CC] Part 2.

Operations applied on requirements are identified by the following means:

- Assignment: written bold
- Selection: written underlined
- Refinement: written italic
- Component Iteration: The complete component is repeated. All repeated components are identified by an ongoing number in brackets after their unique component identification number in the head line of this component. The element identifiers do not contain this additional attribute.
  Example:
  FDP_IFF.1 (1) Simple security attributes
  FDP_IFF.1.1 The TSF shall …
  FDP_IFF.1 (2) Simple security attributes
  FDP_IFF.1.1 The TSF shall …
- Simplified Component Iteration:
  According to [CC] part 1 section 171 it is not necessary to repeat all identical parts of a component in case of iteration. Only the respective element of the component is repeated. All repeated elements are identified by an ongoing number in brackets after their unique element identification number.
  Example:
  FMT_MSA.1.1 (1) The TSF shall …..
  FMT_MSA.1.1 (2) The TSF shall …..

  Due to the fact that dependencies between components must be on the level of single iterations of single components, the following will be defined:
  If the simplified component iteration will be applied, all dependencies/references to this component and all the dependencies/references from this component must be valid for all iterations. This is also valid for security objectives and security functions related to the component be iterated.

5.1 TOE Security Functional Requirements

5.1.1 FAU Security audit

5.1.1.1 FAU_GEN.1 Audit data generation

Hierarchical to: No other components.

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:
a) Start-up and shutdown of the audit functions;
b) All auditable events for the not specified level of audit; and
c) **after the operations R.Verify_Outbound, R.CRC_Check, R.Downgrade, R.Sanitize, R.Set_Mode, R.Test, everytime an operator enters an input and terminates the operator console.**

FAU_GEN.1.2 The TSF shall record within each audit record *O.Data_Audit which contains* at least the following information:
a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the ST[12],
   - **In case of message filtering, the input data O.Data_Class**
   - **In case of message filtering, the sequence number O.Data_Class**
   - **In case of message filtering, the filter rules from O.Filter_Rule_Set that are applied**
   - **In case R.Downgrade is performed and this operation is successfully completed, the resulting O.Data_Unclass.**
   - **In case R.Downgrade is performed and this operation is not successfully completed, a blank message**
   - **In case of operator input, the specific character input by the operator.**
   - **In case of operator input, the user ID of the user who started the operator console**

Dependencies: FPT_STM.1 Reliable time stamps (included (environment))

[12] PP was omitted.

5.1.2 FDP User data protection

5.1.2.1 FDP_ETC.2 Export of user data with security attributes

Hierarchical to: No other components.

FDP_ETC.2.1 The TSF shall enforce the **P.DECLASSIFICATION_POLICY** when exporting user data *O.Data_Audit and O.Data_Unclass*, controlled under the SFP(s), outside of the TSC.

FDP_ETC.2.2 The TSF shall export the user data with the user data's associated security attributes.

FDP_ETC.2.3 The TSF shall ensure that the security attributes, when exported outside the TSC, are unambiguously associated with the exported user data.

FDP_ETC.2.4 The TSF shall enforce the following rules when user data is exported from the TSC:

- **In case of O.Data_Audit**
  - **Data is only exported to Secure_IT_Platform**
  - **SA.OS_MAC_Level = 'Admin high, Classified'**
  - **SA.OS_Priv_Level = 'Unprivileged'**
  - **SA.Security_Label = 'CLASSIFIED'**
- **In case of O.Data_Unclass**
  - **SA.OS_MAC_Level = 'Unclassified'**
  - **SA.OS_Priv_Level = Unprivileged'**
  - **SA.Security_Label = 'NATO UNCLASSIFIED / PN RELEASEABLE**

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
(hierarchical component FDP_IFC.2 (1) included)

5.1.2.2 FDP_IFC.2 (1) Complete information flow control

Hierarchical to: FDP_IFC.1

FDP_IFC.2.1 The TSF shall enforce the **P.DECLASSIFICATION_POLICY** on **O.Data_Class, O.Data_Unclass, L1-Provider and LIFOS** and all operations *(these are the operations R.Sanitize, R.Downgrade and their sequence)* that cause that information to flow to and from subjects covered by the SFP[13].

FDP_IFC.2.2 The TSF shall ensure that all operations that cause any information in the TSC to flow to and from any subject in the TSC are covered by an information flow control SFP.

Dependencies: FDP_IFF.1 (1) Simple security attributes (included)

5.1.2.3 FDP_IFC.2 (2) Complete information flow control

Hierarchical to: FDP_IFC.1

FDP_IFC.2.1 The TSF shall enforce the **P.KEEP-ALIVE-POLICY** on **O.Command and the two parts of the TOE** and all operations that cause that information to flow to and from subjects covered by the SFP.

FDP_IFC.2.2 The TSF shall ensure that all operations that cause any information in the TSC to flow to and from any subject in the TSC are covered by an information flow control SFP.

Dependencies: FDP_IFF.1 (2) Simple security attributes (included)

---

[13] This requirement is rephrased to list explicitly all operations that cause the information to flow to and from subjects covered by the P.DECLASSIFICATION_POLICY.

The appropriate information flow control policy will be defined in the context of the Security Policy Model as part of the requirements of ADV_SPM.1. Figure 4 on page 52 shows the principles. The rule set described in "[Rules]" defines the sanitization rules and their sequence. Due to the fact that this rule set has no rule for "downgrading" or a premature exit, the complete sanitization process must be finished before a message can be downgraded.

5.1.2.4 FDP_IFC.2 (3) Complete information flow control

Hierarchical to: FDP_IFC.1

FDP_IFC.2.1 The TSF shall enforce the **P.INTER-TOE-COMMUNICATION** on **O.Command, O.Output_Message and the two parts of the TOE** and all operations that cause that information to flow to and from subjects covered by the SFP.

FDP_IFC.2.2 The TSF shall ensure that all operations that cause any information in the TSC to flow to and from any subject in the TSC are covered by an information flow control SFP.

Dependencies: FDP_IFF.1 (3) Simple security attributes (included)

5.1.2.5 FDP_IFF.1 (1) Simple security attributes

Hierarchical to: No other components.

FDP_IFF.1.1 The TSF shall enforce the **P.DECLASSIFICATION_POLICY** based on the following types of subject and information security attributes: **L1-Provider, LIFOS, O.Data_Class and SA.Oper_Mode, SA.Security_Label, time since last O.Command.**

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

- **O.Data_Class flows from L1-Provider to LIFOS,**
- **SA.OS_MAC_Level = 'Admin high, Classified' for all operations until and including R.Downgrade,**
- **SA.OS_MAC_Level = 'Unclassified' for all operations after R.Downgrade,**
- **SA.OS_Priv_Level = 'Privileged' for the operation R.Downgrade**

FDP_IFF.1.3 The TSF shall enforce the

- **removal of all data that does not pass R.Verify_Output, R.Sanitize, R.CRC-Pack and R.CRC_Check while flowing from L1-Provider to LIFOS.**

FDP_IFF.1.4 The TSF shall provide the following *additional SFP capabilities:* **none**[14]

FDP_IFF.1.5 The TSF shall explicitly authorise an information flow based on the following rules: **none** [15]

FDP_IFF.1.6 The TSF shall explicitly deny an information flow based on the following rules: **The testframe part of the TOE did not receive any O.Command from the operator console since 3 minutes.**

Dependencies: FDP_IFC.1 Subset information flow control
(hierarchical component FDP_IFC.2 (1) included)
FMT_MSA.3 (1) Static attribute initialisation (included)

[14] FDP_IFF.1.4 does not add information relevant for the TSF. The wording was adapted to this meaning.

[15] FDP_IFF.1.5 does not add information relevant for the TSF. The wording was adapted to this meaning.

5.1.2.6 FDP_IFF.1 (2) Simple security attributes

Hierarchical to: No other components.

FDP_IFF.1.1 The TSF shall enforce the **P.KEEP-ALIVE-POLICY** based on the following types of subject and information security attributes: **Operator Console, Testframe, time since the last O.Command was sent.**

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

- **O.Commands flows from the Operator Console to the Testframe**

FDP_IFF.1.3 The TSF shall enforce the *additional information flow control SFP rules:* **none**[16]

FDP_IFF.1.4 The TSF shall provide the following *additional SFP capabilities:* **If there is no other O.Command communication, the operator console sends every 10 seconds an O.Ping to the testframe part of the TOE** [17]

FDP_IFF.1.5 The TSF shall explicitly authorise an information flow based on the following rules: **none** [18]

FDP_IFF.1.6 The TSF shall explicitly deny an information flow based on the following rules: **none** [19]

Dependencies: FDP_IFC.1 Subset information flow control
(hierarchical component FDP_IFC.2 (2) included)
FMT_MSA.3 (2) Static attribute initialisation (included)

---

16 FDP_IFF.1.3 does not add information relevant for the TSF. The wording was adapted to this meaning.

17 The wording was adapted to this meaning.

18 FDP_IFF.1.5 does not add information relevant for the TSF. The wording was adapted to this meaning.

19 FDP_IFF.1.6 does not add information relevant for the TSF. The wording was adapted to this meaning.

5.1.2.7 FDP_IFF.1 (3) Simple security attributes

Hierarchical to: No other components.

FDP_IFF.1.1 The TSF shall enforce the **P.INTER-TOE-COMMUNICATION** *policy* based on the following types of subject and information security attributes: **the two parts of the TOE, the network interface, the O.Command port and the O.Output_Message port.**

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

- **The network interface is the loopback interface,**
- **The O.Output_Message port is**
  - **from the range of 8181 to 8187**
  - **not used by any other application than the TOE**
- **The O.Command port is**
  - **the number of the O.Output_Message port increased by 1**
  - **from the range of 8182 to 8188**
  - **not used by any other application than the TOE**

FDP_IFF.1.3 The TSF shall enforce the *additional information flow control SFP rules:* **none**[20]

FDP_IFF.1.4 The TSF shall provide the following *additional SFP capabilities:* **none**[21]

FDP_IFF.1.5 The TSF shall explicitly authorise an information flow based on the following rules: **none** [22]

FDP_IFF.1.6 The TSF shall explicitly deny an information flow based on the following rules: **none** [23]

Dependencies: FDP_IFC.1 Subset information flow control
(hierarchical component FDP_IFC.2 (3) included)
FMT_MSA.3 (3) Static attribute initialisation (included)

[20] FDP_IFF.1.3 does not add information relevant for the TSF. The wording was adapted to this meaning.

[21] FDP_IFF.1.4 does not add information relevant for the TSF. The wording was adapted to this meaning.

[22] FDP_IFF.1.5 does not add information relevant for the TSF. The wording was adapted to this meaning.

[23] FDP_IFF.1.6 does not add information relevant for the TSF. The wording was adapted to this meaning.

5.1.2.8 FDP_ITC.2 Import of user data with security attributes

Hierarchical to: No other components.

FDP_ITC.2.1 The TSF shall enforce the **P.DECLASSIFICATION_POLICY** when importing user data *O.Data_Class*, controlled under the SFP, from outside of the TSC.

FDP_ITC.2.2 The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3 The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

FDP_ITC.2.4 The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC:

- **SA.OS_MAC_Level = 'Admin high, Classified'**
- **SA.OS_Priv_Level = 'Unprivileged'**
- **SA.Security_Label = 'CLASSIFIED'**

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
(hierarchical component FDP_IFC.2 (1) included)]
[FTP_ITC.1 Inter-TSF trusted channel, or
FTP_TRP.1 Trusted path]
(FTP_ITC.1 included)]
FPT_TDC.1 Inter-TSF basic TSF data consistency (included)

5.1.2.9 FDP_RIP.1 Subset residual information protection

Hierarchical to: No other components.

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the deallocation[24] of the resource from the following objects: **O.Data_Class and O.Data_Unclass**.

Dependencies: No dependencies

[24] "Deallocation" includes releasing of the main memory upon stop of the TOE.

5.1.3 FMT Security management

5.1.3.1 FMT_MSA.1 (1) Management of security attributes

Hierarchical to: No other components.

FMT_MSA.1.1 The TSF shall enforce the **P.DECLASSIFICATION_ POLICY** to restrict the ability to **change** the security attribute[25] **SA.Oper_Mode** to **S.SysOper**.

Dependencies: [FDP_ACC.1 Subset access control or
FDP_IFC.1 Subset information flow control]
(FDP_IFC.2 (1) included)
FMT_SMF.1 Specification of management functions (included)
FMT_SMR.1 Security roles (included (environment))

5.1.3.2 FMT_MSA.1 (2) Management of security attributes

Hierarchical to: No other components.

FMT_MSA.1.1 The TSF shall enforce the **P.KEEP-ALIVE-POLICY** to restrict the ability to change default the security attributes **maximum time between two O.Commands and time-out threshold** to **nobody**.

Dependencies: [FDP_ACC.1 Subset access control or
FDP_IFC.1 Subset information flow control]
(FDP_IFC.2 (2) included)
FMT_SMF.1 Specification of management functions (included)
FMT_SMR.1 Security roles (included (environment))

[25] The original text was changed to improve grammar as there is only a single security attribute.

5.1.3.3 FMT_MSA.1 (3) Management of security attributes

Hierarchical to: No other components.

FMT_MSA.1.1 (1) The TSF shall enforce the **P.INTER-TOE-COMMUNICATION** *policy* to restrict the ability to change_default the security attributes **O.Command port and O.Output_Message port** to **S.SysAdmin**.

FMT_MSA.1.1 (2) The TSF shall enforce the **P.INTER-TOE-COMMUNICATION** *policy* to restrict the ability to change_default the security attributes **network interface** to **nobody**.

Dependencies: [FDP_ACC.1 Subset access control or
FDP_IFC.1 Subset information flow control]
(FDP_IFC.2 (3) included)
FMT_SMF.1 Specification of management functions (included)
FMT_SMR.1 Security roles (included (environment))

5.1.3.4 FMT_MSA.3 (1) Static attribute initialization

Hierarchical to: No other components.

FMT_MSA.3.1 The TSF shall enforce the **P.DECLASSIFICATION_ POLICY** to provide restrictive default values for security attributes[26]

- *SA.Oper_Mode = Peace Operational Mode*
- *SA.Security_Label = up to and including CLASSIFIED*

that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow **nobody**[27] to specify alternative initial values to override the default values when an object or information is created.

Dependencies: FMT_MSA.1 (1) Management of security attributes (included)
FMT_SMR.1 Security roles (not included)[28]

[26] The original text has been changed to accommodate a list of default values.

[27] The original text was modified to make the sentence grammatically correct after defining the assignment.

[28] This dependency is not applied, because the only security role involved is ‘nobody’.

5.1.3.5 FMT_MSA.3 (2) Static attribute initialization

Hierarchical to: No other components.

FMT_MSA.3.1 The TSF shall enforce the **P.KEEP-ALIVE-POLICY** to provide restrictive default values for security attributes[29]

- *Maximum time between two O.Commands =10 seconds*
- *Time-out threshold =180 seconds*

that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow **nobody**[30] to specify alternative initial values to override the default values when an object or information is created.

Dependencies: FMT_MSA.1 (2) Management of security attributes (included)
FMT_SMR.1 Security roles (not included)[31]

5.1.3.6 FMT_MSA.3 (3) Static attribute initialization

Hierarchical to: No other components.

FMT_MSA.3.1 The TSF shall enforce the **P.INTER-TOE-COMMUNICATION** *policy* to provide restrictive default values for security attributes[32]

- *IP address of the network interface = 127.0.0.1*
- *O.Output_Message port = 8181*
- *O.Command port = 8182*

that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow **nobody**[33] to specify alternative initial values to override the default values when an object or information is created.

Dependencies: FMT_MSA.1 (3) Management of security attributes (included)
FMT_SMR.1 Security roles (not included)[34]

---

[29] The original text has been changed to accommodate a list of default values.

[30] The original text was modified to make the sentence grammatically correct after defining the assignment.

[31] This dependency is not applied, because the only security role involved is ‘nobody’.

[32] The original text has been changed to accommodate a list of default values.

[33] The original text was modified to make the sentence grammatically correct after defining the assignment.

[34] This dependency is not applied, because the only security role involved is ‘nobody’.

5.1.3.7 FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions**:**

- **Set SA.Oper_Mode**
- **Monitor SA.Oper_Mode**

Dependencies: No Dependencies

5.1.4 FPT Protection of the TSF

5.1.4.1 FPT_AMT.1 Abstract machine testing

Hierarchical to: No other components.

FPT_AMT.1.1 The TSF shall run a suite of tests during initial start-up to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

Dependencies: No dependencies

5.1.4.2 FPT_FLS.1 (1) Failure with preservation of secure state

Hierarchical to: No other components.

FPT_FLS.1.1 The TSF shall preserve *that the operation R.Downgrade is not performed* when the following types of failures occur:

- **a failure of R.Sanitize**.
- **a failure of R.Test**.

Dependencies: ADV_SPM.1 Informal TOE security policy model (included)

5.1.4.3 FPT_FLS.1 (2) Failure with preservation of secure state

Hierarchical to: No other components.

FPT_FLS.1.1 The *operator console* shall *automatically exit* when the following types of failures occur:

- **unexpected log out of the user without exit the operator console.**
- **unexpected close of the window the operator console runs within without exiting the operator console.**
- **receiving a SIGTERM signal from the operating system due to a manual kill of the process.**

Dependencies: ADV_SPM.1 Informal TOE security policy model (included)

5.1.4.4 FPT_ITT.1 Basic internal TSF data transfer protection

Hierarchical to: No other components.

FPT_ITT.1.1 The TSF shall protect TSF data from modification when it is transmitted between separate parts of the TOE.

Dependencies: No dependencies

5.1.4.5 FPT_RVM.1 Non-bypassability of the TSP

Hierarchical to: No other components.

FPT_RVM.1.1 The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

Dependencies: No dependencies

5.1.4.6 FPT_TDC.1 Inter-TSF basic TSF data consistency

Hierarchical to: No other components.

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret

- **O.Data_Class**

when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 The TSF shall use **[STANAG5501] message decoding** when interpreting the TSF data from another trusted IT product.

Dependencies: No dependencies

5.1.4.7 FPT_TST.1 TSF testing

Hierarchical to: No other components.

FPT_TST.1.1 The TSF shall run a suite of self tests during initial start-up to demonstrate the correct operation of the **sanitization function**.

FPT_TST.1.2 The TSF shall provide authorized users with the capability to verify the integrity of **SA.Oper_Mode and O.Filter_Rule_Set**.

FPT_TST.1.3 The TSF shall provide authorized users with the capability to verify the integrity of stored TSF executable code.

Dependencies: FPT_AMT.1 Abstract machine testing (included)

5.1.5 Fault tolerance

5.1.5.1 FRU_FLT.1 Degraded fault tolerance

Hierarchical to: No other components.

FRU_FLT.1.1 The TSF shall ensure the operation of **all functions of the testframe part of the TOE** when the following failures occur: **the operator console does not run**.

Dependencies: FPT_FLS.1 (2) Failure with preservation of secure state (included)

5.1.6 FTP Trusted path/channels

5.1.6.1 FTP_ITC.1 Inter-TSF trusted channel

Hierarchical to: No other components.

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and a remote trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2 The TSF shall permit the TSF to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for **R.Downgrade**

Dependencies: No dependencies

5.2 Strength-of-function claim

No strength-of-function claim.

5.3 TOE Security Assurance Requirements

The assurance level of the TOE is EAL4.

Components for Configuration management (**Class ACM**)

ACM_AUT.1 Partial CM automation
ACM_CAP.4 Generation support and acceptance procedures
ACM_SCP.2 Problem tracking CM coverage

Components for Delivery and operation (**Class ADO**)

ADO_DEL.2 Detection of modification
ADO_IGS.1 Installation, generation, and start-up procedures

Components for Development (**Class ADV**)

ADV_FSP.2 Fully defined external interfaces
ADV_HLD.2 Security enforcing high-level design
ADV_LLD.1 Descriptive low-level design
ADV_IMP.1 Subset of the implementation of the TSF
ADV_RCR.1 Informal correspondence demonstration
ADV_SPM.1 Informal TOE security policy model

Components for Guidance documents (**Class AGD**)

AGD_ADM.1 Administrator guidance
AGD_USR.1 User guidance

Components for Life cycle support (**Class ALC**)

ALC_DVS.1 Identification of security measures
ALC_LCD.1 Developer defined life-cycle model
ALC_TAT.1 Well-defined development tools

Components for Tests (**Class ATE**)

ATE_COV.2 Analysis of coverage
ATE_DPT.1 Testing: high-level design
ATE_FUN.1 Functional testing
ATE_IND.2 Independent testing – sample

Components for Vulnerability assessment (**Class AVA**)

AVA_MSU.2 Validation of analysis
AVA_SOF.1 Strength of TOE security function evaluation
AVA_VLA.2 Independent vulnerability analysis

Table 1, Assurance requirements for the TOE.

5.4 Security Requirements for the IT Environment

All security functional requirements for the IT environment are implemented by Secure_IT_Platform, see [ST-Solaris]. This includes the following SFRs:

FAU_STG.2	Guarantees of audit data availability
FAU_STG.4	Prevention of audit data loss
FDP_ACC.1	Subset access control
FDP_ACF.1	Security attribute based access control
FDP_IFC.1	Subset information flow control
FDP_IFF.2	Hierarchical security attributes
FDP_ITC.1	Import of user data without security attributes
FDP_ITC.2	Import of user data with security attributes
FDP_RIP.1	Subset residual information protection
FIA_UAU.2	User authentication before any action
FIA_UID.2	User identification before any action
FMT_SMR.1	Security roles
FPT_FLS.1	Failure with preservation of secure state
FPT_SEP.1	TSF domain separation
FPT_STM.1	Reliable time stamps

6. TOE Summary Specification

6.1 TOE Security Functions

The security functions that implements the TOE are divided into primary and supporting security functions.

The primary Security Function is:
- SF.Downgrade

The supporting Security Functions regarding the filter functionality are:
- SF.Audit_Export
- SF.Check_Integrity
- SF.Check_Sanitization
- SF.Disregard
- SF.Pack
- SF.Sanitize
- SF.Set_Mode
- SF.StartStop
- SF.Test
- SF.Verify_Outbound

The supporting Security Functions regarding the Operator Console – Testframe communication are:
- SF.Consider_Logout
- SF.Operator_Input
- SF.Keep_Alive
- SF.Sec_Com_Op
- SF.Sec_Com_Testframe
- SF.Keep_Alive_Check

These functions are described below.
Along with the function description the security attributes are indicated. A number of the functions shall be performed sequentially and the sub sequential function is mentioned in the function description.

For some security functions the security attribute SA.Security_Label defines the classification of the processed data and the process itself. The TOE shall be able to work in a CLASSIFIED environment. Therefore, the security attribute has the value CLASSIFIED for the appropriate security functions.

The actual function of the TOE will not be affected by the value of this security attribute because this security attribute is part of (or considered by) the underlying operating system only.

In Figure 3 an overview of the intended solution is provided. This picture shows especially the security functions responsible for the communication between operator console and testframe.

Figure 4 shows the security functions of the actual filter part of the TOE in detail. All security functions relevant for the communication only are not included in this picture.

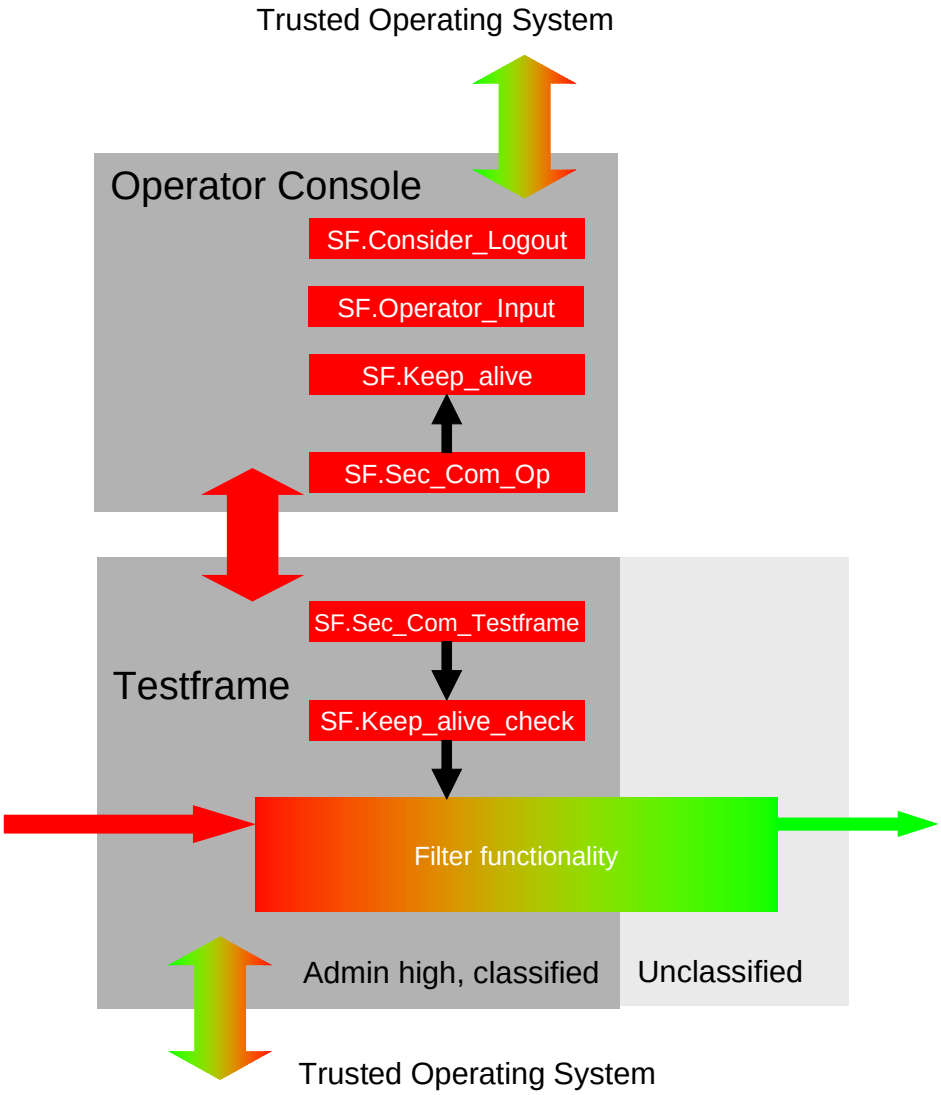

Figure 3: Overview of the Security Functions of the communication functionality and their relation.

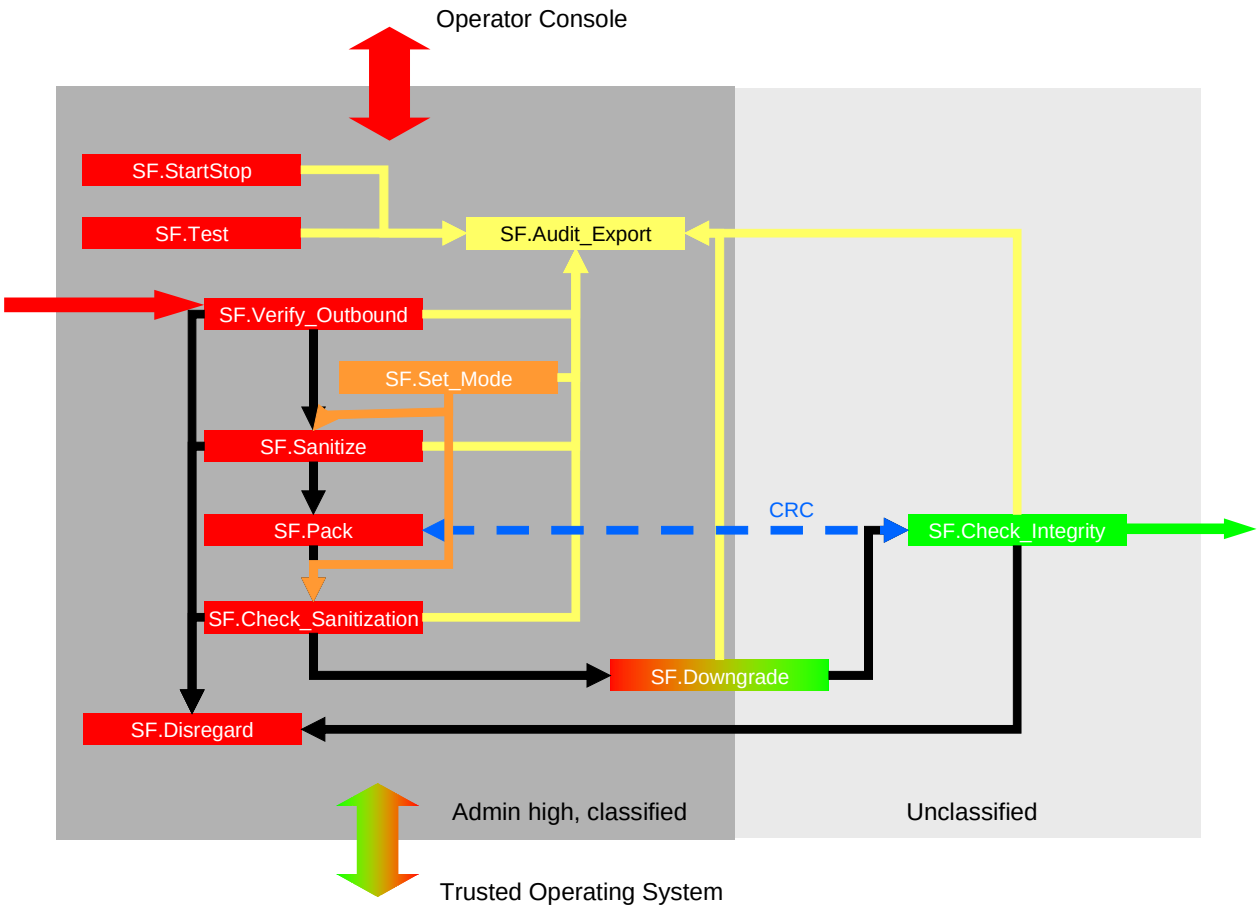

Figure 4: Overview of the Security Functions of the filter functionality and their relation.

6.1.1 Primary Security Functions

6.1.1.1 SF.Downgrade

This function aims at downgrading sanitized O.Data_Class from the classified to unclassified partition on the Secure Operating System.

SF.Downgrade performs the operation R.Downgrade on sanitized O.Data_Class, provided by SF.Check_Sanitization only. The SF.Downgrade passes the sanitized O.Data_Class as a 'new' O.Data_Unclass on to SF.Check_Integrity.

After the operation R.Downgrade SF.Downgrade generates O.Data_Audit. Date, Time and the frame content (including the sequence number) will be recorded. Due to the fact that SF.Downgrade does not modify the content of a frame, a blank frame will be recorded if the originally received frame must not be send out. Otherwise, the frame content of the sanitized frame (which is O.Data_Unclass after SF.Downgrade) will be recorded.

Security Attribute	Value
SA.Oper_Mode	Not applicable for this function
SA.OS_MAC_Level	Admin high, Classified
SA.OS_Priv_Level	Privileged
SA.Security_Label of the processed data	Starting with CLASSIFIED, Resulting in NATO UNCLASSIFIED/PN RELEASABLE

6.1.2 Supporting Security Functions for the actual filter

6.1.2.1 SF.Audit_Export

This function aims at recording audit logs of all operations done by the security functions in order to trace all changes made on the Link-1 data.

The security function SF.Audit_Export implements the operation R.Audit_Trail. SF.Audit_Export receives O.Data_Audit from other security functions. The security function writes an audit trail on the Secure_IT_Platform.

This security function does not generate O.Data_Audit.

Security Attribute	Value
SA.Oper_Mode	Not applicable for this function
SA.OS_MAC_Level	Admin high, Classified
SA.OS_Priv_Level	Unprivileged
SA.Security_Label of the processed data	CLASSIFIED

6.1.2.2 SF.Check_Integrity

This function aims at checking the integrity of the downgraded O.Data_Unclass by recalculating its cyclic redundancy check.

The security function SF.Check_Integrity performs the R.CRC_Check operation on the O.Data_Unclass provided by the SF.Downgrade only.
The operation confirms or denies the check done on the data in the O.Data_Unclass.

When the check is confirmed O.Data_Unclass is ready for transmission. This will take place with SA.OS_MAC_Level = Unclassified, SA.OS_Priv_Level = Unprivileged and SA.Security_Label of the processed data = NATO UNCLASSIFIED/PN RELEASABLE.
When the verification is denied O.Data_Unclass is passed on to the security function SF.Disregard.

After the operation R.CRC_Check SF.Check_Integrity generates O.Data_Audit. Date, Time and the frame content (including the sequence number) be sent out will be recorded.

Security Attribute	Value
SA.Oper_Mode	Not applicable for this function
SA.OS_MAC_Level	Unclassified
SA.OS_Priv_Level	Unprivileged
SA.Security_Label of the processed data	NATO UNCLASSIFIED/PN RELEASABLE

6.1.2.3 SF.Check_Sanitization

This function aims at verification and sanitization of O.Data_Class mandated by the O.Filter_Rule_Set appropriate for the current SA.Oper_Mode.

The security function SF.Check_Sanitization performs the operation R.Sanitize on the sanitized O.Data_Class, provided by SF.Pack only.
The SF.Check_Sanitization passes the sanitized O.Data_Class on to SF.Downgrade, when there is no data in the object that is rejected by R.Sanitize. When sanitized O.Data_Class contains rejected data, this O.Data_Class is passed on to SF.Disregard.[35]

After the operation R.Sanitize SF.Check_Sanitization generates O.Data_Audit. Date, Time, the rule numbers applied and the frame content (including the sequence number) after the sanitization will be recorded.

Security Attribute	Value
SA.Oper_Mode	All, different mode equals different rule set
SA.OS_MAC_Level	Admin high, Classified
SA.OS_Priv_Level	Unprivileged
SA.Security_Label of the processed data	CLASSIFIED

[35] Basically, this function is identical to SF.Sanitize but acts as control function. In this security function the implementation of the R.Sanitize is based on a rule-based mechanism and this is a different mechanism than the implementation of the R.Sanitize in SF.Sanitize.

6.1.2.4 SF.Disregard

This function aims at disregarding and deleting invalid outcomes of other security functions in a controlled manner to prevent unelaborated distribution of O.Data_Class or O.Data_Unclass.

The security function SF.Disregard implements the R.Disregard operation.
This operation is used to remove the data of all main memory objects passed to this security function. The function assures that this information is deleted and that no residual information of this data is stored or reused in the main memory of the TOE or the underlying Operating System.

This security function does not generate O.Data_Audit.

Security Attribute	Value
SA.Oper_Mode	Not applicable for this function
SA.OS_MAC_Level	Admin high, Classified
SA.OS_Priv_Level	Unprivileged
SA.Security_Label of the processed data	CLASSIFIED

6.1.2.5 SF.Pack

This function shall add a cyclic redundancy check to the sanitized O.Data_Class. The added cyclic redundancy check is used to verify after the downgrade the resulting O.Data_Unclass is not altered.

The security function SF.Pack performs the operation R.CRC_Pack on the sanitized O.Data_Class, provided by SF.Sanitize only.
The SF.Pack passes the sanitized O.Data_Class on to the SF.Check_Sanitization after the operation R.CRC_Pack has been performed.

This security function does not generate O.Data_Audit.

Security Attribute	Value
SA.Oper_Mode	Not applicable for this function
SA.OS_MAC_Level	Admin high, Classified
SA.OS_Priv_Level	Unprivileged
SA.Security_Label of the processed data	CLASSIFIED

6.1.2.6 SF.Sanitize

This function aims at the sanitization of O.Data_Class as mandated by the O.Filter_Rule_Set appropriate for the current SA.Oper_Mode.

The security function SF.Sanitize performs the R.Sanitize operation on the O.Data_Class provided by SF.Verify_Outbound only. The SF.Sanitize completely blocks O.Data_Class by generating a “sanitized” blank message (also considered as O.Data_Class) or the function generates a new sanitized O.Data_Class with the data that is not rejected by the operation R.Sanitize. The new sanitized O.Data_Class is passed on to the SF.Pack and the O.Data_Class provided by the SF.Verify_Outbound is passed on to the SF.Disregard.
In this security function the implementation of the R.Sanitize is based on a case based mechanism, which is a different mechanism than the implementation of the R.Sanitize in SF.Check_Sanitization.

After the operation R.Sanitize SF.Sanitize generates O.Data_Audit. Date, Time, the rule numbers applied and the frame content (including the sequence number) after the sanitization will be recorded.

Security Attribute	Value
SA.Oper_Mode	All, different mode equals different rule set
SA.OS_MAC_Level	Admin high, Classified
SA.OS_Priv_Level	Unprivileged
SA.Security_Label of the processed data	CLASSIFIED

6.1.2.7 SF.Set_Mode

This function will set the appropriate set of filter rules that will be enforced by the operation R.Sanitize.

The security function SF.Set_Mode implements the R.Set_Mode operation. The R.Set_Mode operation sets the O.Filter_Rule_Set to one of the SA.Oper_Mode. The operation is allowed to be performed by the S.SysOper. Default value for SA.Oper_Mode = “Peace Operational Mode”.

After the operation R.Set_Mode SF.Set_Mode generates O.Data_Audit. Date/time, the old mode and the new mode of operation will be recorded. If the old and the new mode of operation are identical, it will be recorded that the mode remains unchanged.

Security Attribute	Value
SA.Oper_Mode	Peace Operational Mode
SA.OS_MAC_Level	Admin high, Classified
SA.OS_Priv_Level	Unprivileged
SA.Security_Label of the processed data	CLASSIFIED

6.1.2.8 SF.StartStop

This function records the date and time of the testframe start-up and shutdown.

The security function SF.StartStop generates an audit record at start-up and at controlled shutdown of the TOE.

SF.StartStop generates O.Data_Audit. Date/time and the event will be recorded.

Security Attribute	Value
SA.Oper_Mode	Not applicable for this function
SA.OS_MAC_Level	Admin high, Classified
SA.OS_Priv_Level	Unprivileged
SA.Security_Label of the processed data	CLASSIFIED

6.1.2.9 SF.Test

This function will test the correct operation of the filter and the Secure_IT_Platform.

The security function SF.Test implements a suite of tests during the start of the TOE. This suite tests at least:

- Correct operation of the TSF,
- Integrity verification of the TSF and TSF data for the S.SysOper,
- Check whether Secure_IT_Platform is running.

SF.Test is executed before any other function of the TOE. If SF.Test detects an error, this will be recorded and the TOE stops.

After each test SF.Test generates O.Data_Audit. Date/time and the operator input will be recorded. If the operator approves the test results by entering 'Y', this is a record the successful self-tests. All other input (record content) states unsuccessful self-tests.

Security Attribute	Value
SA.Oper_Mode	Not applicable for this function
SA.OS_MAC_Level	Admin high, Classified
SA.OS_Priv_Level	Unprivileged
SA.Security_Label of the processed data	CLASSIFIED

6.1.2.10 SF.Verify_Outbound

This function aims at verification of syntactical [STANAG5501] compliance of the O.Data_Class received from the L1-Provider.

The security function SF.Verify_Outbound performs the operation R.Verify_Outbound on the O.Data_Class provided by the L1-Provider only. This operation confirms or denies the verification performed on O.Data_Class:

- When the verification is confirmed O.Data_Class is passed on to SF.Sanitize.
- When the verification is denied O.Data_Class is passed on to SF.Disregard.

After the operation R.Verify_Outbound SF.Verify_Outbound generates O.Data_Audit. Date, Time, the rule numbers applied and the frame content (including the sequence number) after the sanitization will be recorded.

Security Attribute	Value
SA.Oper_Mode	Not applicable for this function
SA.OS_MAC_Level	Admin high, Classified
SA.OS_Priv_Level	Unprivileged
SA.Security_Label of the processed data	CLASSIFIED

6.1.3 Supporting Security Functions regarding the Operator Console – Testframe communication

All these security functions do have the same security attribute values. Therefore, the respective values are stated here for all security functions.

Security Attribute	Value
SA.Oper_Mode	Not applicable for these functions
SA.OS_MAC_Level	Admin high, Classified
SA.OS_Priv_Level	Unprivileged
SA.Security_Label of the processed data	CLASSIFIED

6.1.3.1 SF.Consider_Logout

This function aims at recognition of an (unexpected) end of the operator console process.

The security function SF.Consider_Logout implements the Unix standard behaviour to handle and consider signals send from the operating system to processes. If this function receives an SIGTERM from the operating system, the

process will be ended like the user initiated exit. This signal will usually be sent in case of a user logout, close of the window the application runs within or when the user explicitly sends this signal in order to kill the application.
Furthermore, the security function handles user input which exits the application (the operator inputs ‘x’).
This means especially that the network connected to the testframe will be terminated correctly and the process can be removed from the main memory.

This security function generates O.Data_Audit (type of the event) for SF.Operator_Input and a special command will be sent to the testframe part which records the exit of the operator console, too.

6.1.3.2 SF.Keep_Alive

This function aims at sending O.Ping to the testframe every 10 seconds when no other O.Command will be sent.

This security function implements the first part of the keep-alive system of the TOE. The operator console verifies every 10 seconds whether an O.Command was send to the testframe. If no O.Command was sent, an O.Ping will be sent in order to signal the testframe that the operator console is still running.

This security function does not generate O.Data_Audit.

6.1.3.3 SF.Keep_Alive_Check

This function aims at receiving O.Command (including O.Ping) from the operator console and controlling the information flow between the L1-providing System and LIFOS.

This security function implements the second part of the keep-alive system of the TOE.
Every time an O.Command was received, a timer will be initialised with the current time. The precision of this timer is at least the second.
Every time a frame is received testframe verifies the value of this timer and compares it with the current system time.[36]

- If the difference is lower than 3 minutes, the testframe and especially the filter part of the testframe work within normal parameters.
- If the difference is greater than or equal to 3 minutes, the testframe will be stopped (recorded by SF.StartStop). This result in a complete blocking of

[36] It should be noted that the TOE, the underlying operating system and the hardware do not have and do not need real-time properties. Therefore, the time frame “10 seconds” is only an approximate value. The precision depends from the precision of the hardware system clock. The timer check per 10 seconds of the testframe depends from whether the testframe actually has to filter messages or not.

all messages received from the L1-providing system.
In order to avoid undefined states, message processing must not be performed when the timer is validated.

All O.Commands received from the operator console will be considered by this security function. This security function does not interpret or modify O.Commands. All O.Command except O.Ping will be forwarded to the filter functionality.

This security function does generate O.Data_Audit. Date/time of the stop of the filter and the event itself will be recorded.

6.1.3.4 SF.Operator_Input

This security function records start and stop of the operator console as well as all user input.

This security function maintains separate audit files for the operator console. These files will contain a complete protocol of all actions an operator has initialised by entering commands. All key presses will be recorded. Furthermore, start and stop of the operator console will be recorded in these files. All the records include the date/time and the Unix user ID of the operator.

This security function does generate O.Data_Audit but does not forward these information to SF.Audit_Export.

6.1.3.5 SF.Sec_Com_Op

This function aims at building up a secure network connection to the testframe part in order to be able to exchange O.Command and O.Output_Message in a secure way.

This security function implements the network interface of the operator console. This interface is able to build up a connection to the testframe part by using the loopback interface of the operating system. The connection consists of two separate IP connections (sockets), one for outgoing and one for incoming traffic. Each IP connection will be handled by a specified port number on the loopback interface. The port number for the outgoing IP connection must be between 8182 and 8188. The port number for the incoming IP connection must be between 8181 and 8187 and must be the number of the outgoing port decreased by one (1).
The operator console connects to the outgoing port and provides the incoming port for the testframe process.[37]
During start-up of the operator console this security function checks whether the loopback interface and valid port numbers are configured. Otherwise, the operator console starts up with the default values for this port configured erroneously (8181, 8182).

The operating system assures that no other process can use these ports. Therefore, the information sent/received over these ports is protected.
The parameters of the two IP connections are stored in the configuration file of the TOE. This means, only S.SysAdmin is able to maintain the parameters due to access restriction to this file. The access restrictions will be enforced by the environment (the operating system).
The operator console will receive all O.Output_Messages from the testframe, when the operator console is running.

This security function does not generate O.Data_Audit.

[37] Please remark that “incoming” and “outgoing” is from the operator console’s point of view.

6.1.3.6 SF.Sec_Com_Testframe

This function aims at building up a secure network connection to the operator console in order to be able to exchange O.Command and O.Output_Message in a secure way.

This security function implements the network interface of the testframe part of the TOE. This interface is able to build up a connection to the operator console by using the loopback interface of the operating system. The connection consists of two separate IP connections (sockets), one for incoming and one for outgoing traffic. Each IP connection will be handled by a specified port number on the loopback interface.
The port number for the incoming IP connection must be between 8182 and 8188.
The port number for the outgoing IP connection must be between 8181 and 8187 and must be the number of the incoming port decreased by one (1).
The testframe connects to the outgoing port and provides the incoming port for the operator console.[38]
During start-up of the testframe this security function checks whether the loopback interface and valid port numbers are configured. Otherwise, the testframe starts up with the default values for this port configured erroneously (8181, 8182).

The operating system assures that no other process can use these ports. Therefore, the information sent/received over these ports is protected.
The parameters of the two IP connections are stored in the configuration file of the TOE. This means, only S.SysAdmin is able to maintain the parameters due to access restriction to this file. The access restrictions will be enforced by the environment (the operating system).
The testframe will receive all O.Command from the operator console, when the testframe is running.

This security function does not generate O.Data_Audit.

6.1.4 Probabilistic and permutational functions and mechanisms

None.

6.2 Assurance Measures

Appropriate assurance measures are employed to satisfy the security assurance requirements. The following list gives a mapping between the assurance requirements and the documents containing the information needed for the fulfilment of the respective requirement.

[38] Please remark that “incoming” and “outgoing” is from the testframe point of view.

Configuration Management (ACM) assurance measures
It will be described and documented which configuration management system is in use, how it works and how it is used. Furthermore, the following documents will be created: the configuration management plan, the acceptance plan and the configuration list which lists all configuration items.
The TOE will be uniquely identified and labelled by a version number which is also used in all other documents.

Delivery and Operation (ADO) assurance measures
All delivery procedures for the TOE will be described and documented. The developer will use these procedures.
All steps necessary for the secure installation, generation and start-up of the TOE will be described and documented.

Development (ADV) assurance measures
For each the Functional Specification, the High Level Design, the Low Level Design, the Information TOE Security Policies and the Analysis of the Correspondence, a document will be created which contains all necessary information and covers all requirements to content and style.
The complete implementation of the TSF will be provided as source code files.

Guidance (AGD) assurance measures
An user guidance for S.SysOper will be provided.
An administrator guidance will not be provided because the TOE does not differ between user and administrator. All roles defined in chapter 3.1.2 are implemented by the underlying operating system but not by the TOE.

Life Cycle (ALC) assurance measures
The physical, procedural, personnel and other security measures applied by the developer are implemented in accordance with [MIL498]. The developer produces evidence that these procedures are followed.
The development and maintenance life cycle model is implemented in accordance with [MIL498]. All tools used for development will be listed and shortly described in a document. All documentation about the tools will be provided. Only such tools, programming languages, code generators, compilers, etc. will be used which are well-defined and work according an accepted standard.

Test (ATE) assurance measures

The test documentation will contain a coverage analysis which shows the complete coverage of all TSF by the tests. A depth of test analysis in this document will show that the TSF operates in accordance with the High Level Design.
Furthermore, all tests will be defined (test plan), described (test procedure description) and the actual results of the test performance be documented. Some of these tests will be performed by using scripts, which will be provided as part of the test documentation.
For the independent evaluator tests, the security target, all design and development documents and the source code as well as a working TOE in an appropriate environment will be provided to the evaluator.

Vulnerability Assessment (AVA) assurance measures

The user guidance provided for fulfilling AGD will be analysed for completeness regarding the AVA_MSU (misuse) requirements. This analysis will be documented.
A SOF analysis will not be performed because the security target does not contain a SOF statement nor contains the TOE a probabilistic or permutational function.
A vulnerability analysis will be performed, documented and provided for evaluation.

7. PP Claims

This Security Target TOE does not claim any conformance to a Protection Profile:

8. Rationale

8.1 Security Objectives Rationale

For each assumption, threat and OSP it will be demonstrated that it is met by the security objectives. The tracings are provided in the following table.

	A.E.INSIDE	A.E.INTER-TOE-COMMUNICATION	A.E.NATO_SECURITY_POLICY	A.E.OUTSIDE	A.E.RECORDING	A.E.TOE_ACCESS_POLICY	A.U.ONLY_WAY	P.DECLASSIFICATION_POLICY	P.INTER-TOE-COMMUNICATION	P.KEEP-ALIVE-POLICY	P.TOE_DATA_INPUT	P.TOE_FAIL_INSECURE	T.BYPASS	T.MODE_SYNC	T.NEGLIGENCE	T.OPERATOR_DOES_NOT_EXIT	T.TOE_REPROGRAM
SOT.CONSIDER_LOGOUT										X						X	
SOT.DATA_AUDIT								X							X		
SOT.DATA_EXPORT															X		
SOT.DOWNGRADE								X									
SOT.FAIL_SECURE												X					
SOT.FILTER_RULE								X						X			
SOT.KEEP_ALIVE										X							
SOT.NO_BYPASS													X				
SOT.NO_REPROGRAM																	X
SOT.NO_RESIDUAL												X	X				
SOT.SANITIZE								X			X						
SOT.SECURE_COMMUNICATION									X								
SOE.AUDIT_REVIEW															X		
SOE.DATA_AUDIT															X		
SOE.MODE_SYNC								X						X			
SOE.SECURE_ENVIRONMENT		X	X			X			X								
SOE.SECURE_IT_PLATFORM					X	X			X			X	X				X
SOE.SECURE_USAGE		X	X						X				X		X	X	
SOE.SECURE_COMMUNICATION									X								
SOE.TOE_LOCATION	X			X			X						X				

Table 2, Environment to Objectives.

A.E.INSIDE
The assumption is directly met by the objective SOE.TOE_LOCATION indicating that the inner side of the TOE is connected to a L1-Provider.

A.E.NATO_SECURITY_POLICY
The assumption is met by the following objectives:
- SOE.SECURE_ENVIRONMENT directly implements the mandated policy for a secure facility in which the TOE is located.
- SOE.SECURE_USAGE defines the procedures to install, use and maintain the TOE.

A.E.OUTSIDE
The assumption is directly met by the objective SOE.TOE_LOCATION indicating that the outer side of the TOE is only connected appropriately to a LIFOS.

A.E.RECORDING
The assumption is directly met by the objective SOE.SECURE_IT_PLATFORM indicating that all security events will be recorded (logged) on the level of the operating system.

A.E.TOE_ACCESS_POLICY
The assumption is directly met by SOE.SECURE_IT_PLATFORM, which enables the operating system to restrict the access to the TOE, and SOE_SECURE_ENVIRONMENT which directly implements the mandated policy and restricts the access to the TOE

A.E.INTER-TOE-COMMUNICATION
The assumption is directly met by SOE.SECURE_USAGE which ensures that the operating system is configured properly. SOE_SECURE_ENVIRONMENT supports this by restricting the access to the system and the configuration.

A.U.ONLY_WAY
The assumption is directly met by the objective SOE.TOE_LOCATION indicating that the TOE is the only communication path between the L1-Provider and LIFOS.

P.DECLASSIFICATION_POLICY

The policy P.DECLASSIFICATION_POLICY is met by a series of objectives:

- SOT.DOWNGRADE providing that only sanitized data are downgraded
- SOT.SANITIZE providing the rules for sanitization
- SOT.FILTER_RULE and SOE.MODE_SYNC providing that the appropriate O.Filter_Rule_Set is used according to SA.Oper_Mode set by S.SysOper.
- SOT.DATA_AUDIT that generates O.Data_Audit. This enables the subject S.Audit to check whether O.Data_Class has been transmitted.

P.INTERNAL-TOE-COMMUNICATION

The policy P.INTERNAL-TOE-COMMUNICATION is met by a series of objectives:

- SOT.SECURE_COMMUNICATION ensures that the TOE accepts and builds up connections from/to the local machine only.
- SOE.SECURE_COMMUNICATION ensures that the local system and the processes running on it do not interfere with the inter-TOE communication.
- SOE.SECURE_USAGE ensures that the system and the TOE are configured properly.
- SOE.SECURE_ENVIRONMENT ensures that only authorized personnel has access to the system.
- SOE.SECURE_IT_PLATFORM ensures that the system configuration will be enforced and cannot be circumvented.

These objectives ensure that no other process can read, manipulate, deny, replay or spoof the communication between the two parts of the TOE.
SOT.SECURE_COMMUNICATION ensures that the two parts of the TOE are configured correctly so that a communication is possible. Together with the four objectives for the environment listed above, it ensures that each part of the TOE will receive all commands/messages intended for it.

P.KEEP-ALIVE-POLICY

- SOT.KEEP_ALIVE ensures that the testframe part is able to work without a running operator console and that the testframe recognises nearly immediately that the operator console does not run. After three (3) minutes without an O.Ping message from the operator console, the testframe part stops.
- SOT.CONSIDER_LOGOUT ensures that the operator console exits even in the case of an (unexpected) user logout or equivalent event. This enables the testframe in all cases to recognise that the operator console does not run.

P.TOE_DATA_INPUT

The policy P.TOE_DATA_INPUT is met by the SOT.SANITIZE providing that the TOE is able to

- *Outbound*: handle different types of outbound bit streams, received from the Link-1 provider, installed in accordance with A.E.NATO_SECURITY_POLICY requirements and resulting in a sanitized [STANAG5501] Link-1 Message.

P.TOE_FAIL_INSECURE

The policy P.TOE_FAIL_INSECURE is met by the objective SOT.FAIL_SECURE countering a failure in the operation R.Sanitize before operation R.Downgrade is performed. This means that a not sanitized message will not be downgraded (and sent out), regardless of errors or failures of the hardware, the operating system or the TOE software.

In addition, this policy is met by SOT.NO_RESIDUAL ensuring that no classified information may remain in memory after disregarding of messages or stop of the TOE. This means, the operating system or another software is not able to access these information.

The assigned objective SOE.SECURE_IT_PLATFORM ensures that the TOE runs on Secure_IT_Platform, which is a dependable platform regarding the hardware and the operating system.

T.BYPASS

The objective is met by SOE.TOE_LOCATION which ensures that the TOE is not physically bypassed.

The objective SOT.NO_BYPASS assures that all incoming data will be filtered and cannot bypass the TOE.

SOT.NO_RESIDUAL assures that no O.Data_Class can be accessed from memory or other resources after the TOE is stopped.

SOE.SECURE_IT_PLATFORM assures that all permanently stored classified information in the audit trail cannot be accessed by unauthorized people.

SOE.SECURE_USAGE ensures that no misconfiguration may lead to a bypass of classified information.

T.MODE_SYNC

The threat is countered by SOE.MODE_SYNC and SOT.FILTER_RULE.

SOE.MODE_SYNC provides the procedures to keep SA.Oper_Mode of the TOE synchronised with the SA.Oper_Mode of the L1-Provider.

SOT.FILTER_RULE provides the means to change SA.Oper_Mode of the TOE by S.SysOper.

T.NEGLIGENCE
The threat is countered by the following objectives

- SOT.DATA_AUDIT assuring that the TOE generates O.Data_Audit,
- SOT.DATA_EXPORT assuring that the TOE provides O.Data_Audit to S.Audit,
- SOE.DATA_AUDIT assuring that O.Data_Audit are available on the short and long term for S.Audit,
- SOE.AUDIT_REVIEW assuring that S.Audit reviews O.Data_Audit.
- SOE.SECURE_USAGE assures that all authorized users are trained.

T.OPERATOR_DOES_NOT_EXIT
This threat is directly countered by SOT.CONSIDER_LOGOUT. The operator console recognises its termination and exits the normal way.
SOE.SECURE_USAGE also counters the threat because all users are educated that they must not log out without exit the operator console.

T.TOE_REPROGRAM
T.TOE_REPROGRAM is countered by the SOT.NO_REPROGRAM providing that changes of the integrity of the TOE, the SA.Oper_Mode and the O.Filter_Rule_Set are detected at start-up of the TOE. Furthermore, SOE.SECURE_IT_PLATFORM prevents the files of the TOE (including configuration files) from modification by unauthorized user.

8.2 Security Requirements Rationale

The purpose of the Security Requirements Rationale is to demonstrate that the security requirements are suitable to meet the Security Objectives.

8.2.1 The SFRs for the TOE meet the Security Objectives for the TOE

For each Security Objective for the TOE we demonstrate that it is met by the SFRs. The tracings are provided implicitly by the rationales and explicitly by this table.

	SOT.CONSIDER_LOGOUT	SOT.DATA_AUDIT	SOT.DATA_EXPORT	SOT.DOWNGRADE	SOT.FAIL_SECURE	SOT.FILTER_RULE	SOT.KEEP_ALIVE	SOT.NO_BYPASS	SOT.NO_REPROGRAM	SOT.NO_RESIDUAL	SOT.SANITIZE	SOT.SECURE_COMMUNICATION
FAU_GEN.1		X							X			
FDP_ETC.2			X	X								
FDP_IFC.2 (1)				X		X		X			X	
FDP_IFC.2 (2)							X					
FDP_IFC.2 (3)												X
FDP_IFF.1 (1)				X			X	X				
FDP_IFF.1 (2)							X					
FDP_IFF.1 (3)												X
FDP_ITC.2											X	
FDP_RIP.1										X		
FMT_MSA.1 (1)						X		X				
FMT_MSA.1 (2)							X					
FMT_MSA.1 (3)												X
FMT_MSA.3 (1)						X						
FMT_MSA.3 (2)							X					
FMT_MSA.3 (3)												X
FMT_SMF.1						X						
FPT_AMT.1									X			
FPT_FLS.1 (1)					X				X			
FPT_FLS.1 (2)	X											
FPT_ITT.1				X							X	
FPT_RVM.1								X				
FPT_TDC.1											X	
FPT_TST.1									X			
FRU_FLT.1							X					
FTP_ITC.1				X								

Table 3, Objectives to SFR.

SOT.CONSIDER_LOGOUT
The objective SOT.CONSIDER_LOGOUT is directly implemented by FPT_FLS.1 (2). An unexpected termination of the operator console will be determined and the operator console exits the normal way.

SOT.DATA_AUDIT
The objective SOT.DATA_AUDIT is directly implemented by FAU_GEN.1 for the generation of O.Data_Audit.

SOT.DATA_EXPORT
The objective SOT.DATA_EXPORT is directly implemented by FDP_ETC.2 preserving that userdata O.Data_Audit is exported outside the TOE.

SOT.DOWNGRADE
The secure handling of labelled information will be assured by FDP_IFF.1 (1). FDP_IFC.2 (1) enforces an information flow control according the P.DECLASSIFICATION_POLICY inside the TOE, which defines among others the sequence of operations. FTP_ITC.1 enables the TOE to transfer unclassified information from a privileged and classified in an unprivileged and unclassified part of the TOE. FPT_ITT.1 assures that O.Data_Unclass will not be modified after R.Downgrade and is therefore integer regarding the respective O.Data_Class. FDP_ETC.2 assures that the generated O.Data_Unclass can be exported outside the TOE.

SOT.FAIL_SECURE
The objective SOT.FAIL_SECURE is directly implemented by FPT_FLS.1 (1) that indicates that R.Downgrade is not performed when R.Sanitize fails.

SOT.FILTER_RULE
The objective SOT.FILTER_RULE is directly implemented by FDP_IFC.2 (1) assuring that the appropriate O.Filter_Rule_Set is used according to the current SA.Oper_Mode. In addition the default values for the P.DECLASSIFICATION_POLICY are restricted by FMT_MSA.3 (1). FMT_SMF.1 enables the TOE to change the filter rules by changing the mode of operation. The roles that are allowed to change the SA.Oper_Mode are restricted by FMT_MSA.1 (1) to S.SysOper.

SOT.KEEP_ALIVE
The objective SOT.KEEP_ALIVE is directly implemented by FDP_IFC.2 (2) and FDP_IFF.1 (2) which enables that the operator console sends O.Commands to testframe. FDP_IFF.1 (1) ensures that the testframe does not filter O.Data_Class after 3 minutes without O.Command communication. FRU_FLT.1 enables the testframe part of the TOE to run without an operator console. FMT_MSA.3 (2) sets the default values “10 seconds” and “180 seconds” and FMT_MSA.1 (2) denies the modification of these values..

SOT.NO_BYPASS
The objective SOT.NO_BYPASS is directly implemented by FPT_RVM.1 assuring that security functions are not invoked and succeed before this is allowed. The objective is supported by FDP_IFF.1 (1) and FDP_IFC.2 (1) to define which policy shall not be bypassed when this policy applies. In addition FMT_MSA.1 (1) restricts the possibilities available to change the mode of operation to authorized users only.

SOT.NO_REPROGRAM
The objective SOT.NO_REPROGRAM is implemented by FPT_TST.1 that checks the integrity of the TSF and TSF data on start up. In addition, FPT_AMT.1 checks the security assumptions on the underlying virtual machine, which is here the trusted operating system. In all cases FAU_GEN.1 ensures that the result of the tests will be recorded in the audit trail. In case of an error, FPT_FLS.1 (1) ensures that the TOE fails into a secure state and does not forward unsanitized frames.

SOT.NO_RESIDUAL
The objective SOT.NO_RESIDUAL is directly implemented by FDP_RIP.1, by ensuring that neither any O.Data_Class nor any rejected parts of O.Data_Class remain available, even the TOE does not run.

SOT.SANITIZE
The objective SOT.SANITIZE is directly enforced by FDP_IFC.2 (1) defining the rules for filtering and the sequence of operations as defined by P.DECLASSIFICATION_POLICY. FDP_ITC.2 assures a dependable import of classified information from outside the TOE. FPT_TDC.1 requires the correct interpretation of the received messages. FPT_ITT.1 ensures that the O.Data_Unclass will not be modified after R.Downgrade is performed.

SOT.SECURE_COMMUNICATION
The objective SOT.SECURE_COMMUNICATION is directly implemented by FDP_IFC.2 (3) and FDP_IFF.1 (3) which enforces that the communication between the two parts of the TOE does not run across an external network and that exactly these two programs communicate.
Furthermore, the objective is enforced by FMT_MSA.3 (3) defining secure default values for the connection and FMT_MSA.1 (3) which ensures that only S.SysAdmin has the ability to change the values of some of the communication parameters but nobody may chose a physical network interface for the connection.

8.2.2 The SFR for the IT environment meet the security objectives for the IT environment

In this section it is shown how all IT security objectives for the IT environment are addressed by security requirements for the IT environment. The security objectives that are purely non-IT are not addressed.

	SOE.SECURE_IT_PLATFORM	SOE.MODE_SYNC	SOE.SECURE_COMMUNICATION
FAU_STG.2	X		
FAU_STG.4	X		
FDP_ACC.1	X	X	X
FDP_ACF.1	X	X	X
FDP_IFC.1	X		
FDP_IFF.2	X		
FDP_ITC.1	X		
FDP_ITC.2	X		
FDP_RIP.1	X		
FIA_UAU.2	X	X	X
FIA_UID.2	X	X	X
FMT_SMR.1	X	X	X
FPT_FLS.1	X		
FPT_SEP.1	X		
FPT_STM.1	X		

Table 4, Objectives for the IT Environment to SFR for the IT Environment.

SOE.SECURE_IT_PLATFORM

The objective SOE.SECURE_IT_PLATFORM is implemented by a series of SFRs; see for a precise description [ST-Solaris]:

- FAU_STG.2 and FAU_STG.4 to make storage of audit event generated by the TOE possible,
- FDP_ACF.1 and FDP_ACC.1 to restrict access to the Secure_IT_Platform and the TOE to authorised users only,
- FDP_IFC.1 and FDP_IFF.2 to define the operation levels of the Secure_IT_Platform
- FDP_ITC.1 and FDP_ITC.2 to import data from the TOE
- FDP_RIP.1 to prevent the existence of residual information after termination of the TOE operating system process.
- FIA_UAU.2 and FIA_UID.2 to authenticate and identify users in the IT environment.
- FMT_SMR.1 to maintain security roles for the Secure_IT_Platform,
- FPT_FLS.1 to preserve the secure state of the Secure_IT_Platform,
- FPT_SEP.1: to separate the logical execution of the TOE from any other programs running on the Secure_IT_Platform,
- FPT_STM.1 to provide a reliable time stamp for correct audit file records.

SOE.MODE_SYNC

The objective SOE.MODE_SYNC is implemented by a series of SFRs that are provided by the Secure_IT_Platform:

- FDP_ACF.1 and FDP_ACC.1 to restrict access to the Secure_IT_Platform and the TOE to S.SysOper
- FIA_UAU.2 and FIA_UID.2 to authenticate and identify S.SysOper.
- FMT_SMR.1 to maintain S.SysOper as a role for the Secure_IT_Platform

Note that SOE.MODE_SYNC is realized by IT, but also by non-IT.

SOE.SECURE_COMMUNICATION

The objective SOE.SECURE_COMMUNICATION is implemented by a series of SFRs that are provided by the Secure_IT_Platform:

- FDP_ACF.1 and FDP_ACC.1 to restrict access to the Secure_IT_Platform
- FIA_UAU.2 and FIA_UID.2 to authenticate and identify S.SysAdmin.
- FMT_SMR.1 to maintain S.SysAdmin as a role for the Secure_IT_Platform

Note that SOE.SECURE_COMMUNICATION is realized by IT, but also by non-IT.

8.2.3 The security objectives for the non-IT environment needs not to be met by the TOE or its IT environment

The following security objectives for the environment are met by non-IT measures, and therefore not elaborated in this Security Target:

- SOE.AUDIT_REVIEW
- SOE.DATA_AUDIT
- SOE.MODE_SYNC
- SOE.SECURE_COMMUNICATION
- SOE.SECURE_ENVIRONMENT
- SOE.SECURE_USAGE
- SOE.TOE_LOCATION

- Note that SOE.MODE_SYNC and SOE.SECURE_COMMUNICATION are realized by IT, but also by non-IT.

8.2.4 Justification for the Assurance level

Adequate protection of CLASSIFIED information is the driver for this evaluation. The protection is merely transformed to assurance in ‘good security design’, because:

1. Due to NATO policies, the location and environmental personnel, physical and organisation security measures are on the level of CLASSIFIED and thus the TOE is constantly under control of physical and personal security measures and the persons that deal with the TOE are familiar with this kind of security measures.
2. The capabilities required by potential threat agents are considered to be high. However due to the security measures mentioned under point 1 and limited possibilities for untrusted interaction with the TOE an attack profile “low” is sufficient.

EAL4 provides the requirements to provide good commercial practice in security design and aids the evaluators at all design abstraction layers. In addition EAL4 provides additional assurance with the development of the TOE, the testing and the deployment.

8.2.5 Strength of Function Claim is appropriate

The TOE does not use any probabilistic or permutational mechanisms, and thus a Strength of Function claim is not appropriate. Therefore, no Strength of Function Claim is defined (see section 5.2)

8.2.6 All dependencies have been met

The dependencies of the SFRs are not completely fulfilled.

FMT_MSA.3 (all iterations) has a dependency to FMT_SMR.1 which is not applied, because the set of security roles is always restricted to “nobody”. Therefore, no rules have to be managed by FMT_SMR.1. This means, all dependencies of the SFRs are implicitly fulfilled.

For every dependency chapter 5 indicates whether this is fulfilled by SFRs for the TOE or for the IT environment. The dependencies of the two FPT_FLS.1 iterations to ADV_SPM are fulfilled by the EAL-level chosen.

The dependencies of the SAR are fulfilled per definition because an EAL level without any augmentations was selected.

8.2.7 The requirements are internally consistent

The SARs are internally consistent, because they are an EAL and therefore cannot cause inconsistencies.

The two FPT_FLS.1 have dependencies on ADV_SPM.1. These dependencies are already covered by EAL4 and will therefore not introduce an inconsistency. All other SARs and SFRs are completely independent of each other, so there are no inconsistencies between them.

The SFRs are internally consistent because

a) The Security Objectives do not conflict each other (see section 8.1)
b) The justifications in sections 8.2.1 show that each Security Objective for the TOE is met by the assigned SFR and these SFR do not conflict each other
c) The justifications in sections 8.2.2 show that each Security Objective for the environment is met by the assigned SFR and these SFR do not conflict each other

Therefore, the requirements assigned to one objective will not conflict with requirements assigned to another objective because

a) the requirements do not affect the same events, operations, data or test, or
b) the requirements are assigned to both security objectives

The security requirements for the IT environment are all derived from the Secure_IT_Platform, which is a Common Criteria certified platform. The security requirements for the IT environment are independent of the other requirements and are internally consistent. All dependencies are resolved and no conflicting requirements are included.

8.2.8 The requirements are mutually supportive

As stated and explained in chapter 8.2.1 the tracing from SFR to security objectives is complete and the SFR are suitable to meet the security objectives.

Chapter 5.1 lists the SFR and their dependencies. All dependencies are resolved (partly by SFR to the environment) respectively the dependency needs not to be resolved for this TOE.
Furthermore, the SFR support each other

a) FPT_RVM.1 prevents bypassing of the TOE and therefore bypassing of the TSP and TSF
   FDP_IFC.2 (1) assures that the TSF will be used in the intended sequence and therefore prevents bypassing of single or several TSF.
   FDP_RIP.1 and FPT_FLS.1 (1) assure that all confidential information will be made unavailable. This prevents bypassing of not sanitized data and therefore bypassing of the TSF.
   FDP_IFF.1 (1) and FDP_ITC.2 prevent bypassing of classified information to unauthorized person by enforcing the correct labelling and therefore the correct handling of these information. FDP_ETC.2 supports this by enforcing the correct labelling even by exporting data outside the TOE. Furthermore, FDP_IFF.1 (1) and FDP_ITC.2 enforce a stop of the information flow when obviously no O.SysOper monitors the TOE. FDP_IFF.1 (2) ensures that a running operator console will be recognised by the testframe.
b) FPT_SEP.1 as part of the environment requirements prevents tampering of the TOE by other software components running on the same hardware.
   FDP_IFC.2 (2), FDP_IFC.2 (3), FDP_IFF.1 (2), FDP_IFF.1 (3), FDP_ITC.2, FMT_MSA.1 (2), FMT_MSA.1 (3), FRU_FLT.1 and FPT_FLS.1 (2) ensure that a modification of the communication settings is not possible and that the testframe part is able to work properly even in the case of a not running operator console.
c) The requirements FMT_MSA.1 (1), FMT_MSA.1 (2), FMT_MSA.1 (3), FMT_MSA.3 (1), FMT_MSA.3 (2), FMT_MSA.3 (3) and FMT_SMF.1 define the default settings for the mode of operation, the communication parameter and restrict the access to modify this mode and values. Therefore, these requirements prevent de-activation or unauthorized modification of TSF as well as spoofing, tampering and information disclosure of O.Command and O.Output_Message due to an illegal network connection.
d) FAU_GEN.1 enables the TOE to generate audit information and FDP_ETC.2 enables the TOE to export (store) these information persistently on the system.
   FPT_STM.1 (environment) assures that the audit log information contain always the correct time and date.
   FPT_AMT.1 and FPT_TST.1 test the TSF and the underlying operating system and will therefore detect modifications.
   FPT_TDC.1 ensures that only [STANAG5501]-compliant messages will be processed. FPT_ITT.1 and FTP_ITC.2 require a TSF internal integrity check which detects unintended modifications on sanitized O.Data_Class and enforces this integrity check by using a trusted channel.

8.3 TOE Summary Specification Rationale

8.3.1 The functions meet the SFRs

For each SFR we demonstrate that it is met by the Security Functions. The tracings are provided implicitly by the rationales and explicitly by this table.

	SF.Audit_Export	SF.Check_Integrity	SF.Check_Sanitization	SF.Consider_Logout	SF.Disregard	SF.Downgrade	SF.Keep_Alive	SF.Keep_Alive_check	SF.Operator_Input	SF.Pack	SF.Sanitize	SF.Sec_Com_Op	SF.Sec_Com_Testframe	SF.Set_Mode	SF.StartStop	SF.Test	SF.Verify_Outbound
FAU_GEN.1		X	X	X		X		X	X		X			X	X	X	X
FDP_ETC.2	X	X							X								
FDP_IFC.2 (1)		X	X			X				X	X			X			X
FDP_IFC.2 (2)							X	X									
FDP_IFC.2 (3)												X	X				
FDP_IFF.1 (1)		X	X			X		X		X	X			X			X
FDP_IFF.1 (2)							X										
FDP_IFF.1 (3)												X	X				
FDP_ITC.2																	X
FDP_RIP.1					X												
FMT_MSA.1 (1)														X			
FMT_MSA.1 (2)							X	X									
FMT_MSA.1 (3)												X	X				
FMT_MSA.3 (1)														X			
FMT_MSA.3 (2)							X	X									
FMT_MSA.3 (3)												X	X				
FMT_SMF.1														X			
FPT_AMT.1																X	
FPT_FLS.1 (1)			X								X					X	
FPT_FLS.1 (2)				X													
FPT_ITT.1		X								X							
FPT_RVM.1		X	X			X				X	X						X
FPT_TDC.1																	X
FPT_TST.1																X	
FRU_FLT.1				X				X									
FTP_ITC.1						X											

Table 5, SFR to TSF.

FAU_GEN.1

FAU_GEN.1 is implemented by SF.Check_Integrity , SF.Check_Sanitization, SF.Consider_Logout, SF.Downgrade, SF.Keep_Alive_check, SF.Operator_Input, SF.Sanitize, SF.Set_Mode, SF.StartStop, SF.Test and SF.Verify_Outbound by generating O.Data_Audit.

- SF.Check_Integrity performs R.CRC_Check and records this.
- SF.Check_Sanitization performs R.Sanitize and records the respective result.
- SF.Consider_Logout recognises the unintentional stop of the operator console and generates respective audit information.
- SF.Downgrade performs R.Downgrade and records this.
- SF.Keep_Alive_check records when the time-out threshold is reached and testframe exits itself.
- SF.Operator_Input records all keys the operator presses. Furthermore, start and stop of the operator console will be recorded.
- SF.Sanitize performs R.Sanitize and records the respective result.
- SF.Set_Mode performs R.Set_Mode and records this.
- SF.StartStop records the start-up and the controlled shutdown of the TOE. In case of a crash, the TOE is not able to record, but this is not possible in any way.
- SF.Test performs R.Test and records the results
- SF.Verify_Outbound performs R.Verify_Outbound and records this.

FDP_ETC.2

The SFR FDP_ETC.2 is implemented by the SF.Check_Integrity, SF.Audit_Export and SF.Operator_Input by exporting O.Data_Unclass respectively O.Data_Audit outside the TSF to the Secure_IT_Platform conform the defined security attributes. After this, O.Data_Audit can be further processed by S.Audit.

FDP_IFC.2 (1)

The SFR FDP_IFC.2 (1) is implemented directly by the sequence of SF.Verify_Outbound, SF.Sanitize, SF.Check_Sanitization, SF.Pack, SF.Downgrade and SF.Check_Integrity enforcing P.DECLASSIFICATION_POLICY and all other supporting checks. The requirement FPT_RVM.1 ensures, that these sequence will be called in all cases. In addition, SF.Set_Mode realizes that all modes of P.DECLASSIFICATION_POLICY can be used.

FDP_IFC.2 (2)

FDP_IFC.2 (2) is implemented directly by SF.Keep_Alive and SF.Keep_Alive_check. When running, the operator console sends an O.Command to the testframe every 10 seconds. (FDP_IFF.1 (2)) The testframe verifies whether

an O.Command was received during the last three minutes. If no O.Command was received, all O.Data_Class from the L1-Provider will be blocked (FDP_IFF.1 (1)).

FDP_IFC.2 (3)
FDP_IFC.2 (3) is implemented directly by SF.Sec_Com_Testframe and SF.Sec_Com_Op. These two security functions ensure that on both sides the network connection is configured properly. Especially the loopback interface and appropriate ports are used. Together with SOE.SECURE_COMMUNICATION satisfied by the environment requirements FDP_ACC.1 and FDP_ACF.1, the network interface and the port numbers can be considered as basic authentication.

FDP_IFF.1 (1)
The SFR FDP_IFF.1 (1) is implemented by the security functions that realize P.DECLASSIFICATION_POLICY (see FDP_IFC.2 (1)).
Furthermore, the SFR is implemented by SF.Keep_Alive_check. If the testframe did not receive any O.Command from the operator console within the last 3 minutes, all messages from the L1-Provider will be blocked.

FDP_IFF.1 (2)
The SFR FDP_IFF.1 (2) is implemented by SF.Keep_Alive. The operator console ensures that every 10 seconds an O.Command will be sent to the testframe. If no regular O.Command needs to be send, an O.Ping will be sent as keep-alive message.

FDP_IFF.1 (3)
The SFR FDP_IFF.1 (3) is implemented by the security functions that realize P.INTER-TOE-COMMUNICATION. SF.Sec_Com_Testframe and SF.Sec_Com_Op establish a one-to-one communication between the operator console and the testframe. The “one-to-one” property is realized by a very basic kind of authentication based on the network interface and the ports used. The S.SysAdmin has to ensure that no other applications on this system will use the respective ports on this network interface. This very simple authentication is considered as sufficient due to the fact that security baseline of the system is very high (organisational, personnel, physical and network security as well as the usage of a high secure operating system)
This connection between the two parts of the TOE ensures that the testframe (and only the testframe) receives all O.Command from the operator console. Also, the operator console (and only the operator console) receives O.Output_Messages from the testframe.
Naturally, a part of the TOE does not receive a message if this part does not run. This behaviour is not considered as error.

FDP_ITC.2
The SFR FDP_ITC.2 is implemented by the SF.Verify_Outbound by importing O.Data_Class received from a L1 provider.

FDP_RIP.1
The SFR FDP_RIP.1 is directly implemented by SF.Disregard, deleting all data that have not passed the other security functions.

FMT_MSA.1 (1)
The SFR FMT_MSA.1 (1) is implemented by SF.Set_Mode, providing only S.SysOper the possibility to change SA.Oper_Mode.

FMT_MSA.1 (2)
The SFR FMT_MSA.1 (2) is implemented by SF.Keep_Alive and SF.Keep_Alive_check which do not permit the modification of these values.

FMT_MSA.1 (3)
The SFR FMT_MSA.1 (3) is implemented by SF.Sec_Com_Op and SF.Sec_Com_Testframe. These two security functions read and check the port numbers for the inter-TOE-connection given in the configuration file. Only S.SysAdmin is able to modify the communication parameter due to access restrictions to the configuration file. The IP address for the communication is not configurable.

FMT_MSA.3 (1)
The SFR FMT_MSA.3 (1) is implemented by SF.Set_Mode providing the initial setting for SA.Oper_Mode.

FMT_MSA.3 (2)
The SFR FMT_MSA.3 (2) is implemented by SF.Keep_Alive and SF.Keep_Alive_check which uses the two default values only.

FMT_MSA.3 (3)
The SFR FMT_MSA.3 (3) is implemented by SF.Sec_Com_Op and SF.Sec_Com_Testframe which provide the three default values.

FMT_SMF.1
The SFR FMT_SMF.1 is implemented by SF.Set_Mode, which enables SA.SysOper to change SA.Oper_Mode. This event generates audit information (FAU_GEN.1) which enables to monitor SA.Oper_Mode.

FPT_AMT.1
The SFR FPT_AMT.1 is directly implemented by SF.Test running a suite of tests during the initial start up.

FPT_FLS.1 (1)
The SFR FPT_FLS.1 (1) is implemented by the SF.Sanitize and SF.Check_Sanitization. Both functions filter all classified data from the data received by the L1-Provider. The way these functions perform the filtering is

implemented differently. SF.Sanitize uses a case based mechanism and SF.Check_Sanitization uses a rule-based mechanism.
Furthermore, the double check of O.Data_Class ensures that an error in one of these functions does not result in downgrading unsanitized data.

- If SF.Sanitize fails, SF_Check_Sanitization will disregard the message.
- If SF_Check_Sanitization fails, the message is already sanitized.

Furthermore, the requirement is implemented by SF.Test. The TOE performs a self-test during start-up and does not filter any messages in case of an not successful test.

FPT_FLS.1 (2)
FPT_FLS.1 (2) is implemented by SF.Consider_Logout. This security function checks whether the operator console receives SIGTERM signals from the operating system. If this signal has been received, the security function exits the operator console in a controlled manner.

FPT_ITT.1
The transfer protection shall cover the protection of O.Data_Unclass after the operation R.Downgrade. This will be enforced by SF.Pack (CRC calculation right before R.Downgrade) and SF.Check_Integrity (CRC verification after R.Downgrade and right before sending the data out).

FPT_RVM.1
The SFR FPT_RVM.1 is implemented by enforcing the execution sequence of the security functions in all cases the TOE receives data. This means, the respective security functions will be called every time a message is received, regardless which form and content this message has.
This requirement is not implemented in a separate security function because the security functionality is that the sequence will be called every time and there are no premature exit points within the called security functions.
The sequence is defined in the requirement FDP_IFC.2 (1) and implemented in the security functions SF.Verify_Outbound, SF.Sanitize, SF.Pack, SF.Check_Sanitization, SF.Downgrade and SF.Check_Integrity.
The requirement FDP_IFC.2 (1) ensures that this sequence will be performed correctly, this means especially without premature exit points.

FPT_TDC.1
The SFR FPT_TDC.1 is implemented by SF.Verify_Outbound. This security function checks the received data package according to defined rules (see Appendix D) whether it is a [STANAG5501]-conformant frame.

FPT_TST.1
The SFR FMT_TST.1 is directly implemented by SF.Test running a suite of tests during the initial start up.

FRU_FLT.1
The SFR FRU_FLT.1 is directly implemented by SF.Keep_Alive_check. This security function ensures that the testframe is able to work without the operator console for three minutes. SF.Consider_Logout supports this security function by ensuring that the inter-TOE connection will be terminated when the operator console exits intentionally or unintentionally.

FTP_ITC.1
The SFR FTP_ITC.1 is implemented by SF.Downgrade by initiating the transport of O.Data_Class from the classified into the unclassified environment (as O.Data_Unclass) and exporting O.Data_Unclass the by calling SF.Check_Integrity (which is the end point of the trusted channel).

8.3.2 The assurance measures meets the SARs
The statement of assurance measures has been presented in the form of a reference to the actions or documents that show that the assurance measures have been met. The documents implement the requirements of EAL4. This statement can be found in section 6.2.

8.4 PP Claims Rationale

This Security Target TOE does not claim conformance to any Protection Profile (see section 7). This section is therefore empty.

9. Appendix A - Abbreviations

A	Assumption
A5OM	Article 5 Operational Mode
ASDE	Air Situation Data Exchange
ASOC	Air Sovereignty Operation Centre
BSI	Bundesamt für Sicherheit in der Informationstechnik
CC	Common Criteria
CEM	Common Evaluation Methodology
CRC	Control and Reporting Centre
CROM	Crisis Response Operational Mode
EAL	Evaluation Assurance Level
EOM	Exercise Operational Mode
F	Functional
IT	Information Technology
ITSEF	IT Security Evaluation Facility
L1FF	Link-1 Forward Filter
LIFOS	Link-1 Fibre Optic Secure System
MAC	Mandatory Access Control
MLS	Multi-Level Secure
NC3A	NATO Consultation, Command and Control Agency
OSP	Organisational Security Policy
P	Policy
PfP	Partnership for Peace
PN	Partner Nation
POM	Peace Operational Mode
PP	Protection Profile
RAP	Recognized Air Picture
SAR	Security Assurance Requirements
SF	Security Function
SFP	Security Function Policy
SFR	Security Functional Requirements
SOE	Security Objective for the Environment
SOF	Strength of Function
SOT	Security Objective for the TOE
SPM	Security Policy Model
ST	Security Target
T	Threat
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSP	TOE Security Policy

10. Appendix B - References

[CC]	Common Criteria for Information Technology Security Evaluation, Parts 1, 2 and 3, version 2.3
[CEM]	Common Methodology for Information Technology Security Evaluation, version 2.3
[MCM140]	MCM-140-00 (N-R) MC Concept for the Air Situation Data Exchange with Partner Nations, 13 September 2000.
[MIL498]	Military Standard – Software Development and Documentation; MIL-STD-498, Department of Defence (DoD), USA, 5 December 1994
[NATO-CIS]	AC/322-D/0030-REV2, Infosec Technical and Implementation Directive for the interconnection of communications and information systems (CIS), 25 October 2002 (NATO UNCLASSIFIED)
[NATO-SP][39]	Security Within The North Atlantic Treaty Organisation (NATO), NATO C-M(2002)49, 17 June 2002 (NATO UNCLASSIFIED).
[SRS]	NC3A, System Requirements Specification, Link 1 Forward Filter (L1FF) for Air Situation Data Exchange (ASDE) with Non-NATO Nations, Draft version 0.3, January 2003.
[STANAG5501]	NATO-MAS, Standardization Agreement Tactical Data Exchange - Link 1 (point-to-point), edition 4 (NATO UNCLASSIFIED)
[ST-Solaris]	SUN Microsystems, Trusted Solaris 8 4/01 Security Target, version 2.0, 14 June 2002
[Rules]	ASDE Link1 Forward Filter and Integrity Filter Rules, NC3A, February 2007, (NATO RESTRICTED)

[39] This policy is the successor of CM(55 15).

11. Appendix C - Glossary of Terms

Security Accreditation Authority: A designated group or section within a NATO headquarters that advise alliance staff as to the conformance and permissibility of the security provisions implemented in their IT systems and network. For NATO C3 Agency (NC3A), the NATO Office of Security (NOS) is the designated SAA. For NATO Programming Centre (NPC), Assistant Chief of Staff on SHAPE Intelligence Division (ACOS/I/SHAPE) is the designated SAA.

Mandatory Access Control: The means whereby unprivileged access to an IT object, e.g. file, process, device, etc. by a subject, e.g. user, process, etc. is protected in such a way that does not require the cooperation of the subject. Subject cannot access MAC protected objects because of the perceived value of the object and not because the subject agrees not to access it.

Multi-Level Secure: A description applied to an IT system that is itself able to securely store and indelibly label items in terms of the true sensitivity of the information. An MLS system is characterized by the use of MAC labels and software that implements a policy of **no read up**, e.g. an uncleared user cannot read a classified item, and **no write down**, e.g. a classified process cannot create an unclassified item without using privileges.

12. Appendix D - Link-1 Forward Filter Sanitization Rules

The filter rules are removed due to classification issues. Details are provided in [Rules]

13. Appendix E - The Need of an Evaluation

Security approval by the Military Committee is required prior to the release of Air Situation Data, or the associated Link 1 documentation, to any PfP nation. Thereafter, each PfP system receiving the Air Situation Data must be approved or accredited by the National Security Authority {as identified in the Security Agreement) and is subject to periodic NATO Office of Security (NOS) inspections under the bi-lateral security agreements. The accreditation must identify the maximum classification to be processed (i.e. during Article 5 operations). NATO policy dictates that a balanced set of security measures (physical, personnel, procedural, computer and communication) shall be identified and implemented to create the secure environment in which an ADP system operates. The system security accreditation process will include the formulation of a System-Specific Security Requirement Statement (SSRS) and Security Operating Procedures (SecOPs) or national equivalents. These documents will be produced by the national ADP System Operational Authority (ADPSOA) or appropriate project staff, approved by the national Accreditation Authority and are subject to NOS review.

The NATO C3 Agency is located in The Hague, The Netherlands. It is a non-profit making element of the North Atlantic Treaty. Staff from this headquarters provides expertise, advice and prototype solutions for the NATO community of users in areas such as command & control, communication, operational research and information technology. This agency will be responsible for designing and implementing the L1FF.

NATO Office of Security is located in Brussels, BE. In the specific context of the NATO C3 Agency, it is responsible to monitor, advise and recommend approval (or otherwise) regarding security measures proposed for or, in the case of prototype equipment, actually deployed on NATO funded computer-based equipment. NOS will be responsible to provide security accreditation and/or approval to operate the L1FF.

The filtering software is a security-enhancing limited functionality product that aims to implement a multi-level secure mode of processing. There is therefore a requirement, under NATO Security Policy, (CM (55) 15 (Final)) for an independent security evaluation of the filtering software by one of the National Evaluation and Certification Authorities, prior to its operational use.