TNOR Guard 1.1.3

Security Target

Includes:

STANAG 4406 Ed 2 Guard

Email (SMTP) Guard

Chat (XMPP) Guard

XML/SOAP Guard

Copyright © THALES Norway AS

DOCUMENT CHANGE HISTORY

Revision	Date	Description
10.4-public	09.05.2022	Initial public release.

	–	10.4-public					
Written by	SE Team	**CHTE**					
Checked by	QA Manager	**TBO**					
Approved by	PDA	**ØYJ**					

Table of Contents

TABLE OF CONTENTS 3
LIST OF FIGURES 4
LIST OF TABLES 4
1. SECURITY TARGET INTRODUCTION (ASE_INT) 6
1.1 SECURITY TARGET REFERENCE 6
1.2 TOE REFERENCE 6
1.3 REFERENCED DOCUMENTS 6
1.4 TOE OVERVIEW 7
1.5 TOE DESCRIPTION 14
2. CONFORMANCE CLAIMS (ASE_CCL) 25
2.1 CC CONFORMANCE CLAIM 25
2.2 PP CONFORMANCE CLAIMS 25
3. SECURITY PROBLEM DEFINITION (ASE_SPD) 26
3.1 ASSUMPTIONS 26
3.2 THREATS 27
3.3 ORGANISATIONAL SECURITY POLICIES 31
4. SECURITY OBJECTIVES (ASE_OBJ) 32
4.1 TOE IT SECURITY OBJECTIVES 32
4.2 TOE ENVIRONMENT SECURITY OBJECTIVES 34
5. EXTENDED COMPONENTS DEFINITION (ASE_ECD) 37
5.1 EXTENDED COMPONENTS DEFINITION (SFRS) 37
6. SECURITY REQUIREMENTS (ASE_REQ) 38
6.1 SECURITY FUNCTIONAL REQUIREMENTS (SFRS) 38
6.2 SECURITY ASSURANCE REQUIREMENTS (SARS) 53
7. TOE SUMMARY SPECIFICATION (ASE_TSS) 54
7.1 TOE SECURITY FUNCTIONS 54
8. RATIONALE 58
8.1 SECURITY OBJECTIVES RATIONALE 58
8.2 SECURITY REQUIREMENTS RATIONALE 65
8.3 TOE SUMMARY SPECIFICATION RATIONALE 73
8.4 PP RATIONALE 75
9. NOTES 76
9.1 NOTATION 76
9.2 ABBREVIATION AND ACRONYMS 76
9.3 TERMINOLOGY 77

Copyright © THALES Norway AS

List of Figures

Figure 1-1 Overview of TOE Environment ... 8
Figure 1-2 – Scenario with two separate guards ... 9
Figure 1-3 – Simplified view of Guard between the security domains ... 12
Figure 1-4 - Decomposing Guard into HW and SW units ... 15
Figure 1-5: Guard Reference Monitor ... 18
Figure 1-6 - Multi-level security messaging guard, both networks have assets that need protection... 19
Figure 1-7 - Original security label marking ... 20
Figure 1-8 - Separation between strategic and tactical networks ... 20
Figure 1-9 XMPP/Chat Guard example ... 21
Figure 1-10 TOE deployed to protect database service ... 22

List of Tables

Table 4-1 TOE Security Objectives ... 34
Table 4-2 Security objectives for the TOE Environment ... 36
Table 6-1 Auditable Events ... 41
Table 6-2 Management functions and operations on TSF data ... 50
Table 8-1: TOE threats coverage ... 59
Table 8-2: Assumptions coverage ... 60
Table 8-3: Policies coverage ... 61
Table 8-4: Security objectives satisfaction (part 1/2) ... 67
Table 8-5 Security objectives satisfaction (part 2/2) ... 69
Table 8-6: Functional requirements dependency check ... 73
Table 8-7: Functional requirements satisfaction ... 75
Table 9-1 Acronyms ... 77
Table 9-2 Terminology ... 78

Copyright © THALES Norway AS

Foreword

This document is the Security Target for Thales Guard. The document describes the operating environment and IT product requirements, as well as security functionality implemented in the IT product.

The document was prepared on behalf of

Forsvarsmateriell IKT-kapasiteter
Postboks 800 Postmottak,
2617 Lillehammer
Norway

By:

THALES Norway AS
Postboks 744 Sentrum
0106 Oslo
Norway

Telephone: *(+47) 22 63 83 00*
E-Mail: *mhs@thales.no*
Internet: *http://www.thales.no*

Copyright notice:

The information contained in this document is proprietary to Thales Norway AS.
You are free to copy and redistribute the document as long as it is unmodified.

THALES Norway AS has made every attempt to ensure that the information in this document is correct and complete. However, THALES Norway AS assumes no liability for errors, or for any damage that may result from the use of this document or any product that it accompanies.

1. SECURITY TARGET INTRODUCTION (ASE_INT)

1.1 SECURITY TARGET REFERENCE

Title:	Security Target for the TNOR Guard
Version:	See front page
Date:	See front page
Document id:	739 20726 AAAA SC

1.2 TOE REFERENCE

TOE name:	TNOR Guard
TOE version:	1.1.3
Product id:	

Product name	Product ID	Supported platforms
STANAG 4406 Message Guard	3AQ 28150	Kontron (B) Generic PC (C)
SMTP Message Guard	3AQ 28151	
XMPP Chat Guard	3AQ 28152	
SOAP XML Guard	3AQ 28153	

The TOE is identified with Product ID, Platform Code and version. The platform code is embedded in a 4-letter qualifier code shown in documentation and the installation media.

Example: "3AQ 28150 AAAB version 1.1.3" identifies the STANAG 4406 Message Guard version 1.1.3 for a Kontron HW.

Product ID	Qualifier (e.g. AAAA)				Version
	Reserved	HW version (TOE Environment)		Platform	
Unique ID (letters or numbers)	A	A..Z	A .. Z	B or C	x.x.x

Assurance level:	EAL4 augmented with ALC_FLR.3 and AVA_VAN.4
CC Identification:	Version 3.1 Revision 5

1.3 REFERENCED DOCUMENTS

[CCPART1] Common Criteria for Information Technology Security Evaluation, April 2017, Version 3.1 revision 5, Part 1 (also known as part 1 of the ISO/IEC 15408 Evaluation Criteria).

[CCPART2] Common Criteria for Information Technology Security Evaluation, April 2017, Version 3.1 revision 5, Part 2 (also known as part 2 of the ISO/IEC 15408 Evaluation Criteria).

[CCPART3] Common Criteria for Information Technology Security Evaluation, April 2017, Version 3.1 revision 5, Part 3 (also known as part 3 of the ISO/IEC 15408 Evaluation Criteria).

[FIPS-180] FIPS PUB 180-4 Secure Hash Standard, NIST, August 2015

[LAPADULA] Bell & La Padula: Secure Computer Systems: *Unified Exposition and Multics Interpretation*

[ST4406] Military Message Handling System Edition 2, NATO C3 Board, 2005

[MILS-PP] Protection Profile Multiple Independent Levels of Security: Operating System, version 2.03, EURO-MILS project, 31 March 2016

[PKCS11] PKCS #11 Cryptographic Token Interface Base Specification Version 2.40, OASIS standard, 14 April 2015.

[RFC3507] Internet Content Adaptation Protocol (ICAP), IETF, April 2003

[RFC6120] Extensible Messaging and Presence Protocol (XMPP): Core, IETF, March 2011

[RFC6121] Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence, IETF, March 2011

[RFC 7622] Extensible Messaging and Presence Protocol (XMPP): Address Format, IETF, September 2015

[SOAP] SOAP Version 1.2 Messaging Framework, Second edition, W3C, 27 April 2007

1.4 TOE OVERVIEW

The TOE is the TNOR Guard, a high-assurance guard providing an automatic and controlled flow of information between two domains that may operate under different security policies. No information is allowed to pass from one of the domains to the other unless the Security Policy of the TOE explicitly allows it to pass.

The TOE covers the following four Guard products:

- STANAG 4406 Ed 2 Message Guard
  For connectivity towards the NATO standard Military Message Handling System (MMHS)

- SMTP Message Guard (E-mail)
  For connectivity towards standard e-mail systems such as Microsoft Exchange.
  Supports RFC 6477 for Military Message Handling attributes within the SMTP domain.

- XMPP Chat Guard

- SOAP XML Guard

A simplified deployment overview for the Guard is shown in Figure 1-1.

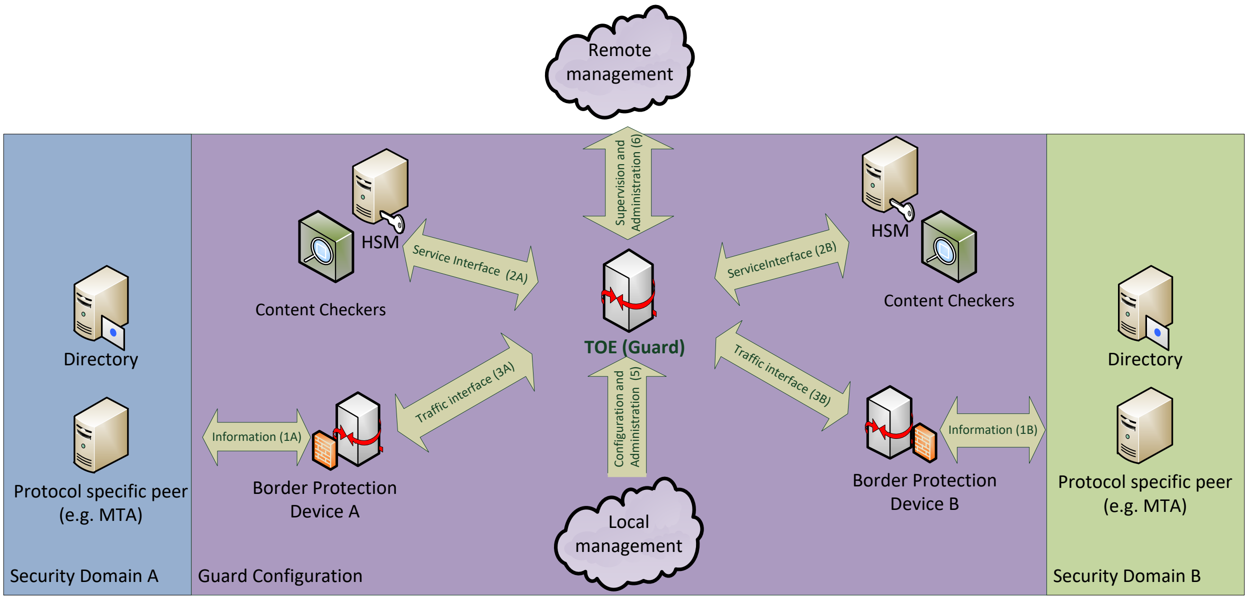

Figure 1-1 Overview of TOE Environment

The Guard communicates with one or more peers in each security domain which acts as a proxy for other services within each of the two security domains. The Guard also uses directory services to access certificates and certificate revocation lists in each of the security domains.

The Guard is installed in a protected environment, with border protection devices mitigating as shown in the figure.

During processing of the information objects (messages) the Guard uses external services, such as directory servers and content checker services to provide necessary information to perform a release decision, and it uses a hardware security modules when signing released information objects. Even though the Guard uses external services it ensures that no parts of, or traces of, the information object is released into the destination domain before a positive release decision has been made.

The Guard provides online tools for management of the run-state, logs and configuration data of the Guard.

The Guard Configuration Tool (TOE Environment) is provided for creating configuration vectors for the Guard (TOE). This software runs on separate computers, and the resulting configuration vectors are loaded via the “Local management” interface as shown in Figure 1-1. The “Remote management” interface is used for management of the Guard (TOE).

The TNOR Guard is transparent to the users of its services in the security domains.

The Guard is not based on store-and-forward principles. The Guard acts as a proxy and important mechanisms in a Military Messaging System such as queuing and recover / retry must be implemented by adjacent MTAs. The Guard does not provide a routing service, it considers all received information objects from one domain to be requests for information release to the other domain.

The Guard receives the information (message) via the traffic interface, and converts the information to a protocol independent format. The Guard processes the information as described in 1.4.1, and decides whether the security policy allows the information to be released, or whether it must be rejected.

Copyright © THALES Norway AS

While processing the information release requests, the Guard has not yet accepted the message from the adjacent MTA. If the information release request is accepted the Guard will send the message to the other adjacent MTA, and once that MTA has accepted responsibility for the message the Guard will free any resources allocated for that messages and acknowledge the message from the MTA in the source domain

If the information release request is rejected the Guard will free any resources allocated for the message, and signal the rejection of the message to the MTA in the source domain. Further handling is the determined by the connected MTA. The Guard does not generate non-delivery reports or provide traffic operator functions.

The Guard runs on top of a separation kernel hypervisor, further described in ch 1.5.1.1 Separation kernel hypervisor and ch 1.5.9.5 PikeOS. The separation kernel hypervisor is used to separate different parts of the TOE using partitions. Separate partitions are used to isolate specific security function implementations from other functions, to separate processing of different messages, and to separate the information flow directions. The IPC mechanism features IPC communication between processes in the same partition, between processes in different partition on the same processing unit, and between processes in different partitions on different processing units. The IPC mechanism also provides strict control on inter-process communication, and denies all IPC communication that is not explicitly allowed on both process-level and on partition-level.

1.4.1 DETAILED PROCESSING STEPS

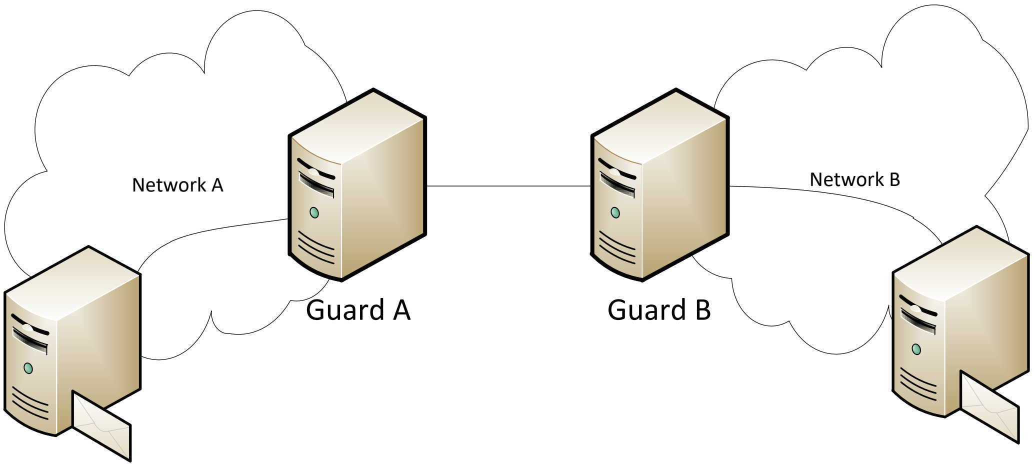

Figure 1-2 – Scenario with two separate guards

Given the scenario in Figure 1-2, Guard A performs the following when processing information that is sent from Network A to Network B. Validation, content checking and filtering is performed in parallel.

#	Step	Description
1	Receive Information Object	Receive Information Object from MTA in Network A, decode and transform to an internal protocol independent representation. Verify addresses against Directory
2	Validate attributes and signature	Validate addresses and possibly perform address translation. The Directory Server in Network A is used to

Copyright © THALES Norway AS

		validate addresses. Validate digital signatures and certificates. The Directory Server in Network A is used if additional certificates are required for validation. This step corresponds to *Digital Signature Validation* and *Discretionary Access Control (DAC)* as described in 1.4.2.
	Content checking	Apply content checking using external content checkers for Network A. This step corresponds to Content Checking as described in 1.4.2.
	Apply filter	Apply the filter in two steps: 1) Check that the Information Object is allowed to flow from security domain A to security domain B according to the configured security policy. Note: The configured security policy must be based on agreements between the owners of the two security domains / networks. This step corresponds to Mandatory Access Control (MAC) as described in 1.4.2. 2) Modify the Information Object according to the security policy of the destination domain (the security policy of Network B), and protocol elements that may be used for covert channels. This step corresponds to the Filter functions as described in 1.4.2.
3	Perform release decision	Based on the previous steps, a release decision is made. If the information is allowed into the destination domain (Network B), the Guard proceeds to the next step. A negative release decision results in an error response to the originating peer.
4	Translate addresses	Possibly perform address translation according to the current configuration vector. Note: Double address translation is not necessary unless both security domains need to protect their address structure.

5	Prepare object for release with new signature	Convert released information from internal representation back to the traffic interface protocol, and sign the data (if required by the Network B security policy). Note: The original digital signature cannot be reused because the information may be changed in steps 2 and 4.
6	Send the information object	Send the information to the MTA in Network B, and notify the sending MTA in Network A that the information has been released.

1.4.2 MAJOR SECURITY FEATURES OF THE TOE

The main security feature of the Guard is to mediate a one-way or bidirectional information flow between two security domains. The Guard inspects every information object that is requested sent between the security domains, and makes an automated release decision according to configured policy.

To support the main security feature as stated above, the Guard performs the following security checks:

- Mandatory Access Control (MAC)
  The Guard configuration specifies rules for which security labels are allowed to flow in and out of each security domain. The rules are typically defined in a bilateral agreement. The Guard is responsible for checking that all information that flows between the security domains conforms to the security policy definitions and the rules for exchanging information between them.

- Discretionary Access Control (DAC)
  The Guard can ensure that only authorized originators and recipients are allowed to exchange information through the Guard. The integrity of the originator and recipient designators may be strengthened using digital signatures.

- Filter
  The Guard can strip, replace or remove parts of the information before it is allowed to flow through the Guard. Note that the Guard must conform to the supported protocol in which the information is transferred, and this limits the ability to modify the information. This function also minimizes the covert channels inherent for the protocols.

- Content Checking
  The TOE offloads the information object payload, label and subject to third-party content checkers. The content checker must return a positive or negative response to the analysis. This allows for detailed checking of information, such as antivirus and malware inspection, text analysis, comparing data types to object content and so on.

- Digital signature validation
  The TOE validates digital signatures (supported by external HSM) that are bound to the information objects, verifying consistency, the certificate chain and any CRL’s.

The security checks above checks different aspects of the information object release request. All these checks and their results combined implements the ABAC (Attribute Based Access Control) feature of the Guard.

Copyright © THALES Norway AS

The security checks are supported also by security functions that ensure that the security checks are performed in a controlled and predictable way. Self-testing is performed regularly to ensure that the software and configuration integrity is maintained. Audit is performed to allow operators of the Guard to observe the security performance of the Guard, and to track any security relevant events. The Guard is furthermore designed to always maintain a secure state, meaning that it shall never perform/allow an uncontrolled information flow between the security domains.

The design of the Guard ensures that the information that belongs to one security domain is not allowed to enter the other security domain until the Guard completes its release decision and the message conforms to the configured information flow policy. See figure below.

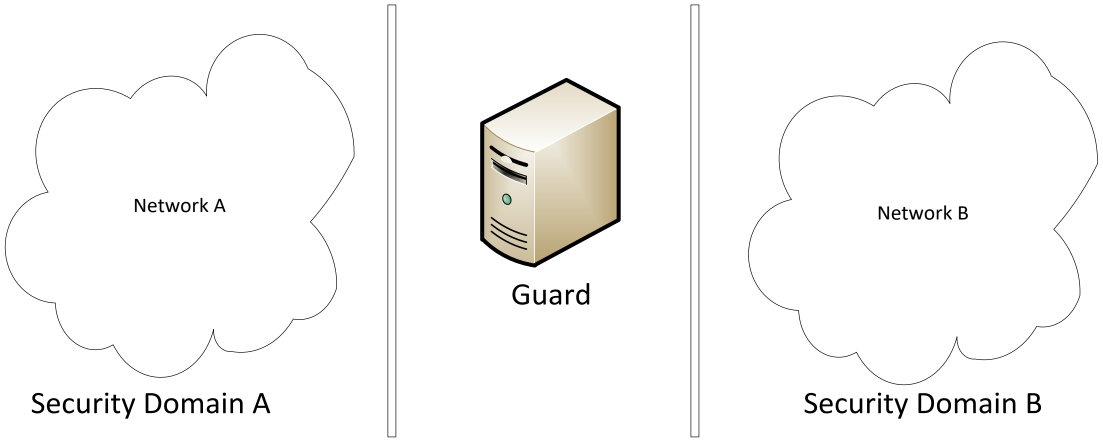

Figure 1-3 – Simplified view of Guard between the security domains

The Guard is located between the security domains, with its traffic interfaces protected by border protection devices towards each of the security domains. Information must be allowed to flow to the Guard in order for the Guard to evaluate whether the information can flow into the other security domain.

The Guard ensures that the defined information flow policy is enforced, only releasing messages that are in conformance with the policy, considering a diverse set of attributes such as originator, recipient, security label, attachment types, content checker results and digital signature validity.

The Guard can hide the address structure of a security domain from the other security domain, and rebuilds each information object with only the attributes that are allowed to flow through the Guard.

The Guard will handle delivery reports and receipt notifications for relevant protocols even when address mapping has been activated. This applies to the STANAG 4406 and E-mail Guards, where delivery of reports and notifications require a correct mapping of identifiers and addresses to ensure delivery of these objects, to ensure they can be associated with previously transmitted messages.

The TOE provides priority handling for STANAG 4406 and SMTP messaging traffic, where priority attributes are available, ensuring that high priority messages are delivered with minimal delay even during periods of heavy traffic or network congestion.

The TOE ensures all administrators are authenticated, and provides user management functions such as automated logout and lockout.

1.4.3 REQUIRED NON-TOE HARDWARE, SOFTWARE AND FIRMWARE

1.4.3.1 Third party services

The Guard may be configured to use the following non-TOE components:

Connected to the TOE service interfaces A and B (Ethernet):

- Content checker, interfacing the Guard through the ICAP protocol [RFC3507].
- Hardware Security Module with a PKCS #11 interface.

Connected to the traffic interfaces A or B (Ethernet):

- An LDAP-compliant Directory Server for retrieval of addresses and intermediate certificates.
- An HTTP/HTTPS web server or LDAP compliant Directory Server for retrieval of CRLs.
- NTP Server

1.4.3.2 Guard Configuration Tool (non-TOE)

The configuration vectors for the TOE are prepared using the Guard Configuration Tool, provided as part of the Guard software media.

The Configuration Tool allows configuration vectors to be created and digitally signed, before being uploaded through the local management interface on the Guard.

The Guard Configuration Tool does not require network access, but does require access to a security token to sign the configuration vectors.

The Configuration Tool requires 64-bit Windows 7/10, or Windows Server 2012 R2.

1.4.3.3 Hardware

The TOE is qualified to run on two hardware configurations:

Variant	**Kontron platform (B)** 2u hardware unit in enclosed chassis.	**Generic PC x86_64 hardware platform (C)** 3 x 1U COTS Intel Xeon based servers.
Hardware vendor	RECAB	Generic PC
Construction	Custom chassis: RECAB 19” 2U “VPX Secure Message Gateway” R235333MLSG1 Standard 2u rack mountable form factor.	3 servers Form factor according to deployment requirements. Usually 1U standard rack mountable servers.
Processing units	Kontron Vx3052-SA	Intel Xeon family CPU

Copyright © THALES Norway AS

	Memory: 8GB	Memory: At least 8GB
Storage	1 x S-ATA III SSD hard drive	1 x S-ATA III SSD hard disk or 1 x NVMe SSD hard disk, capacity at least 500GB Core unit: One additional disk controller and one separate disk, with capacity of at least 100GB.
Network interfaces	1 x Intel 10GbE integrated network interface card 2 x 1Gbit Intel Ethernet NIC	4 x Ethernet ports. Each of the ports needs to have separate PCI unit numbers (e.g. separate NICs)
Internal connectivity	Custom VPX backplane	1 or 10Gbit Ethernet.
Other interfaces	1 x USB 3.0 port 1 x Serial port	1 x USB 3.0 port 1 x Serial port

Hardware Security Modules (HSM) stores the private keys, and are used for some cryptographic functions. The Guard requires either one HSM for each security domain, or one HSM with two network interfaces that is capable of secure separation between two different PKI and ensure that no information is allowed to flow between the two partitions. The HSMs must support PKCS#11.

To allow changes to the hardware without modifications to the TOE, the TOE relies on hardware abstraction layers (Architecture Support Package and Board Support Package) for PikeOS. This allows introducing hardware changes without altering the TOE or how the TOE interfaces its environment, as all hardware access is handled through the PikeOS platform. When introducing new hardware, the PikeOS configuration requires minor updates to correctly adjust for memory size and hardware IDs.

Each supported hardware model requires a separate image to be built, which embeds the TOE (unchanged), and the specific PikeOS configuration.

1.4.3.4 Firmware

The hardware for running the TOE, and any computers running services that are external to the Guard, all come with firmware. The Guard uses the firmware indirectly, as in the units would not boot without the firmware, and expects the firmware to ensure boot integrity, but the Guard does not rely on specific functions in the firmware.

1.5 TOE DESCRIPTION

The Guard (TOE) is part of the XOmail product family for messaging and information exchange in mission-critical military and civilian networks. The Guard implements high-assurance information flow control for the trusted exchange of information across security domain boundaries.

The TOE covers the following four Guard products:

- STANAG 4406 Ed 2Guard
  Military Messaging according to NATO STANAG 4406 Edition 2

- E-mail (SMTP) Guard
  Civilian email and special purpose messaging over SMTP

- Chat (XMPP) Guard
  Instant Messaging service between security domains.

- XML/SOAP Guard
  Exchange of XML/SOAP data between security domains.

Sample operational scenarios are described in 1.5.7.

1.5.1 THE BUILDING BLOCKS

The TOE is comprised by software distributed on three separate hardware instances, as shown in the figure below. The Guard is composed of three hardware units, each with a set of PikeOS separation kernel and software

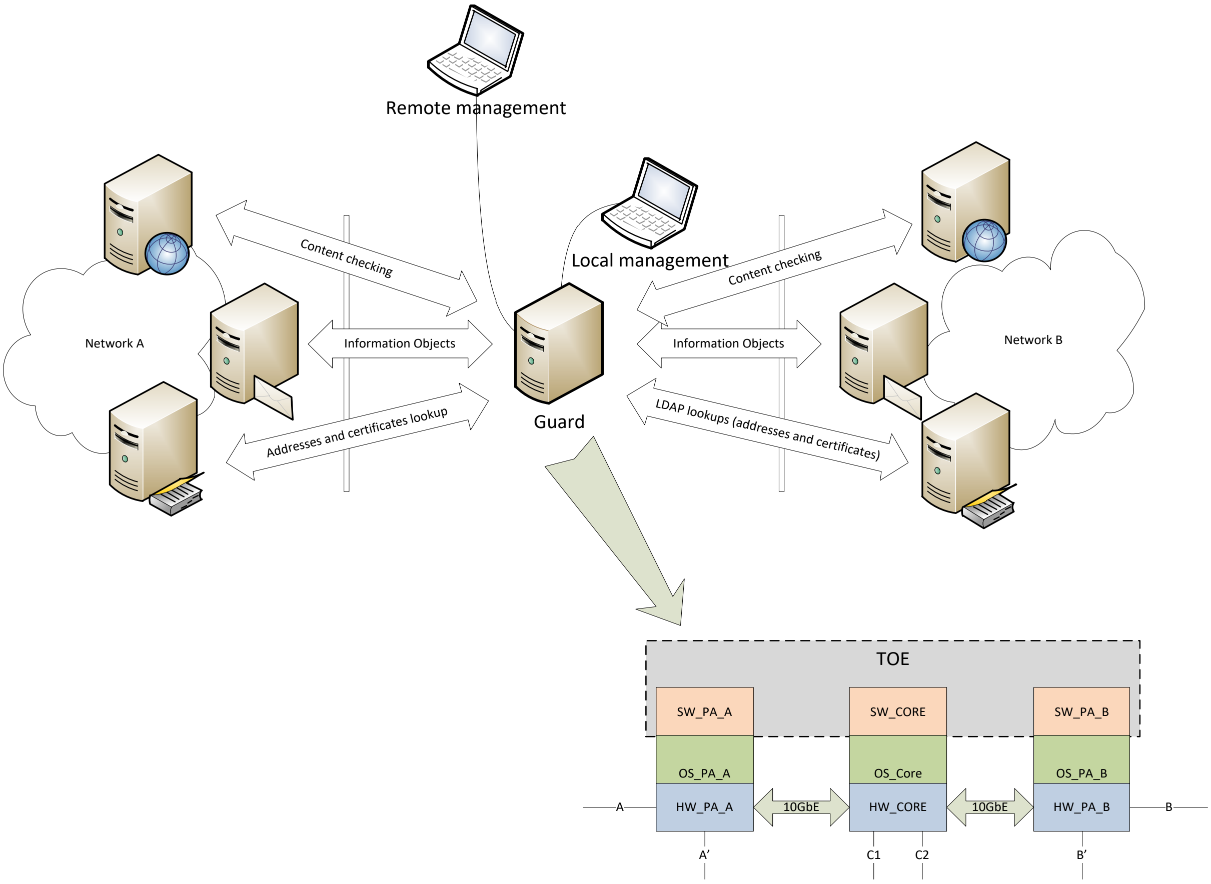

Figure 1-4 - Decomposing Guard into HW and SW units

Each of the processing units shown in Figure 1-4 consists of the following:

- Independent computer (e.g. CPU, memory, controllers, ..)

- Guard software, containing the PikeOS separation kernel platform.

Copyright © THALES Norway AS

The TOE is the Guard application software, which runs within partitions provided by the PikeOS secure hypervisor on three separate processing units. The three processing units are connected as shown on the figure above. The separation kernel and hardware, such as processing units and interconnections, are not part of the TOE.

The Guard product is delivered with an offline management tool that is used for creating configuration data for the Guard. This management tool is not part of the TOE.

1.5.1.1 Separation kernel hypervisor

The TOE runs on top of the SysGo PikeOS separation kernel hypervisor. PikeOS is a MILS type platform and provides strict compartmentalization between different TOE subsystems and modules.

PikeOS provides a set of functions on which the Guard bases its security, for instance separation between partitions, reset of memory before re-use, and device access.

1.5.2 NETWORK INTERFACES

Each of the hardware units has two external Ethernet ports with dedicated controllers, shown as A, A', C1, C2, B and B' in Figure 1-4. The HW_PA_A hardware controls the Ethernet ports for services in Network A and HW_PA_B controls the Ethernet ports for services in Network B:

- Traffic interfaces (interfaces marked A and B in Figure 1-4):
  - Receiving and sending information
    Note that the Guard supports MTA-MTA traffic only. Submit and delivery functions are not supported.
  - Address lookup
  - CRL and intermediate certificates retrieval
  - NTP (one side only)
- Service interfaces (interfaces marked A' and B'):
  - Content checkers
  - HSM interface for signature generation

The HW_CORE controls the Ethernet ports for local and remote management, C1 and C2:

- Management interfaces
  - C1: Local management
  - C2: Remote management

1.5.2.1 Traffic interfaces

The following protocols are supported:

Four versions of the TOE is available, each with support for one of the following protocols for sending and receiving information objects:

- Military Messaging according to NATO STANAG 4406 Ed. 2

- Email (SMTP)
  RFC 5321, RFC 5322
- Chat/XMPP
  RFC 6120, RFC 6121, RFC 7622
- SOAP/XML
  SOAP v1.2

Supporting services, shared by all Guards:

- LDAP for retrieving address information, intermediate certificates and CRLs
- HTTP/HTTPS for retrieving CRLs
- Date and time synchronization using NTP

1.5.2.2 Service interfaces

The following protocols are used:

- Content checkers run on separate computers and interfaces the TOE using the ICAP protocol.
- All cryptographic operations towards the HSMs use PKCS#11.

1.5.2.3 Management interface

The Guard offers two management interfaces, one local and one remote. Both interfaces are provided using SSH, and provide a command line where administrative commands can be executed.

The local interface provides full administrative capabilities, which includes configuration vector management, backup and audit export.

The remote interface provides a limited set of commands, such as changing the active configuration vector and monitoring the TOE interfaces

1.5.2.4 Internal interfaces

10Gbit Ethernet interfaces are used for communication across the hardware units on which the TOE runs, as shown in Figure 1-4.

1.5.3 MANAGEMENT OF THE GUARD

The Guard provides two management interfaces, a local management interface and a remote management interface. All management tasks are potentially available via the local management interface, while only a subset of the management tasks are available via the remote management interface. In addition command access configuration for each administrator may further limit the set of available management tasks.

The management interfaces are provided as SSH-based connections. The SSH tunnel gives access to a command line based set of management tools. Management of the Guard is only possible for authenticated operators.

Management commands include, but are not limited to:

- Loading signed configuration vectors, up to ten configuration vectors can be stored in the Guard
- Change active configuration vector
  (while in management mode, change is implemented on return to active state)
- Manage run-state of the Guard

Copyright © THALES Norway AS

- Manage logs, including export
- Manage operators
- Initiate emergency erase

Audit data may also be automatically exported to a syslog-server, and automatic deletion strategies may be configured to limit disk usage.

1.5.4 PARTITIONS AND OPERATING SYSTEMS

PikeOS offers a trusted partitioning mechanism for strict compartmentalization (partitions). The isolation provided by the separation kernel allows security critical modules to be protected from interference by other critical modules or by less critical or security irrelevant modules running on shared hardware. TOE components and non-TOE components are separated into a set of partitions, each containing a set of processes

1.5.5 REFERENCE MONITOR

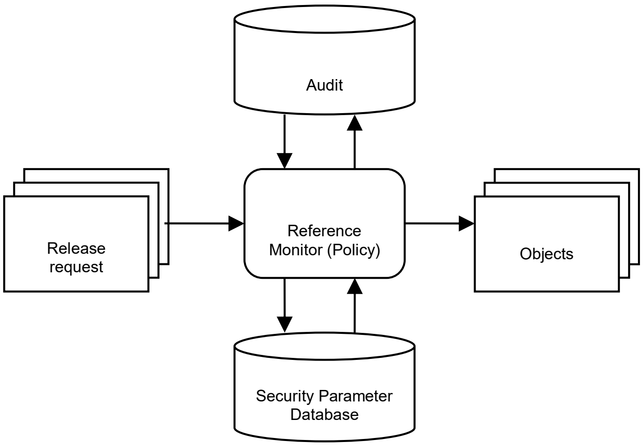

Figure 1-5: Guard Reference Monitor

The TOE implements a reference monitor architecture to enforce a controlled and auditable information flow between the connected domains. The reference monitor is tamperproof, protected by the TSF and TOE Environment, and it is always invoked as part of both internal information flow and object release. A schematic overview of the reference monitor is shown in Figure 1-5.

All information conveyed between the two connected security domains are labelled with a security label (used in MAC) and a priority. The security label is used for access control by the reference monitor, based on the Bell & LaPadula multilevel security policy model [LAPADULA]. The priority attribute ensures that higher priority Information Objects are always given precedence in queues or assigned reserved resources. The security label and priority attributes are part of both the external interfaces and, when available, TOE internal information flow.

1.5.6 STANDALONE MANAGEMENT TOOLS

- The Standalone Management Tools (TOE Environment) are provided to creae configuration data for the Guard, and perform log analysis.

Copyright © THALES Norway AS

1.5.7 TOE USAGE SCENARIOS

As indicated in 1.4 the TOE may be deployed in different scenarios and configurations. Common usage scenarios are shown below.

1.5.7.1 Multi-level security, networks belongs to separate organizations

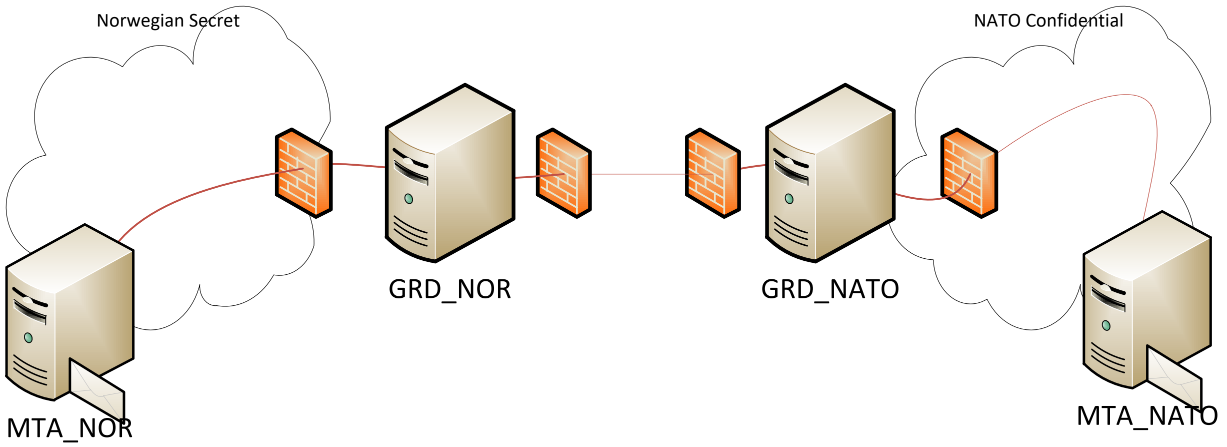

Figure 1-6 - Multi-level security messaging guard, both networks have assets that need protection

When connecting networks from different organizations, both networks have assets that require protection. This configuration is typically implemented with separate guards maintained by each connected organization. In Figure 1-6 the example shows messaging guards connecting two domains. This scenario applies to all Guards covered by the TOE.

Each guard is protected by Border Protection Devices on the traffic interfaces. Border Protection Devices (BPDs) filter and monitor network traffic to detect and prevent low-level network-based attacks.

In this example the guards GRD_NOR and GRD_NATO are transparent to MTA_NOR and MTA_NATO, and logically the MTA_NOR and the MTA_NATO communicates directly with each other.

When connecting a sensitive or classified network to a less sensitive or unclassified network, it may be sufficient with one Guard, as there is only one domain that requires the level of assurance provided.

One security domain may be connected to multiple other security domains, and hence receive information objects that originates from more than two security domains. When receiving information objects that originates from another security domain when receiving security domain must normally be able to guarantee that the received information objects are not accidentally released into a third security domain. This problem is illustrated in Figure 1-7 below where a mission domain and a Swedish domain are connected to a Norwegian domain. The Guards needs to evaluate if the current security domain is the owner of the information objects.

Copyright © THALES Norway AS

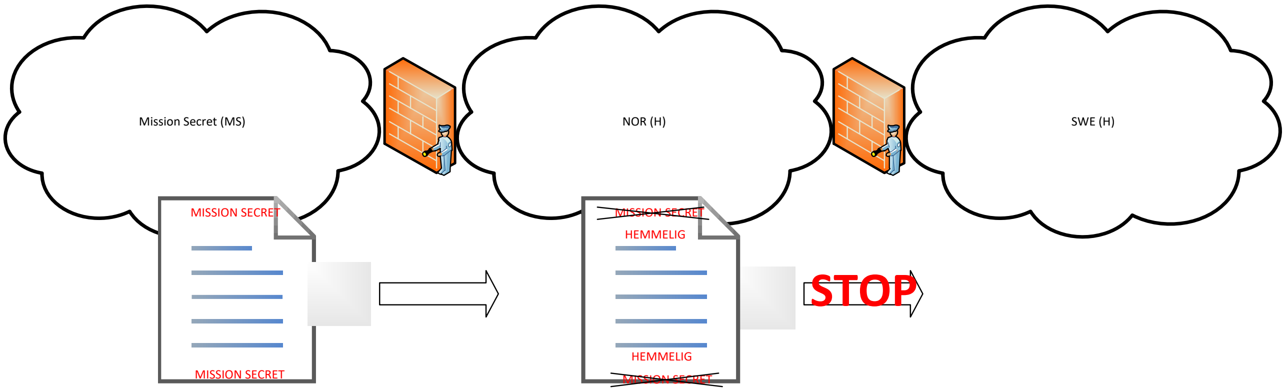

Figure 1-7 - Original security label marking

In this case the information system in the Norwegian domain does not support the mission policy. Therefore the Guard will translate the security labels from mission policy to Norwegian policy, based on the agreement for exchanging information between the security domains. The Guards implements the original security label marking, in which the principle is illustrated in Figure 1-7, and therefore supports connections to multiple security domains also in the case where the security domain does not have knowledge of the security policies in the interconnected security domains.

1.5.7.2 Single guard between two networks within a single organization

The network overview in Figure 1-8 shows a typical configuration where two security domains under the control of the same organization is connected using a single guard. The example below shows a messaging guard, but the same principle applies for XMPP and XML guards.

As both domains belong to the same organization, the Guard is trusted to control information flowing from either domain.

In the example below the TOE enforces information flow controls between a strategic and tactical network.

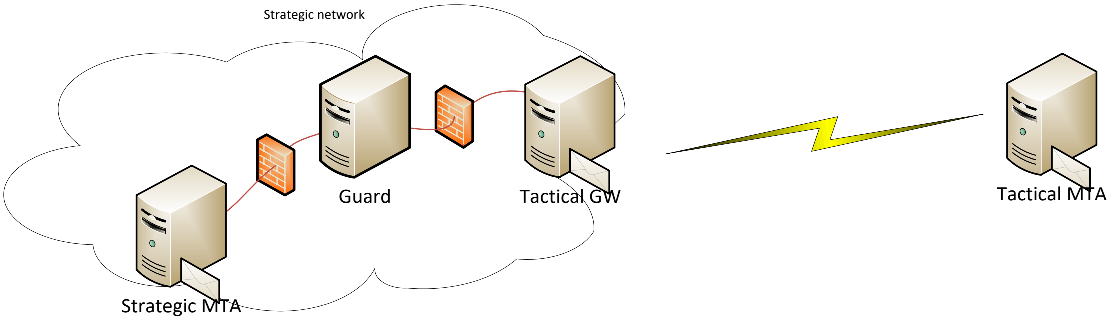

Figure 1-8 - Separation between strategic and tactical networks

Strategic and tactical networks may have different properties that result in requirements for a guard, even if the two domains operate on the same security level defined by the same security policy:

- Shorter life cycle and higher change rate on information.
- Tactical nodes may be more vulnerable to enemy attacks.
- Lower trust attributable to information within the tactical domain.

Additional environments where this network configuration would apply:

- National Restricted network connected to a National Secret network
- “Office” network connected to a high-value industrial production environment

These are both examples where the two security domains are owned by the same organization, but trust and sensitivity levels require a strong separation and information flow control mechanism.

1.5.7.3 Chat Guard and group communications

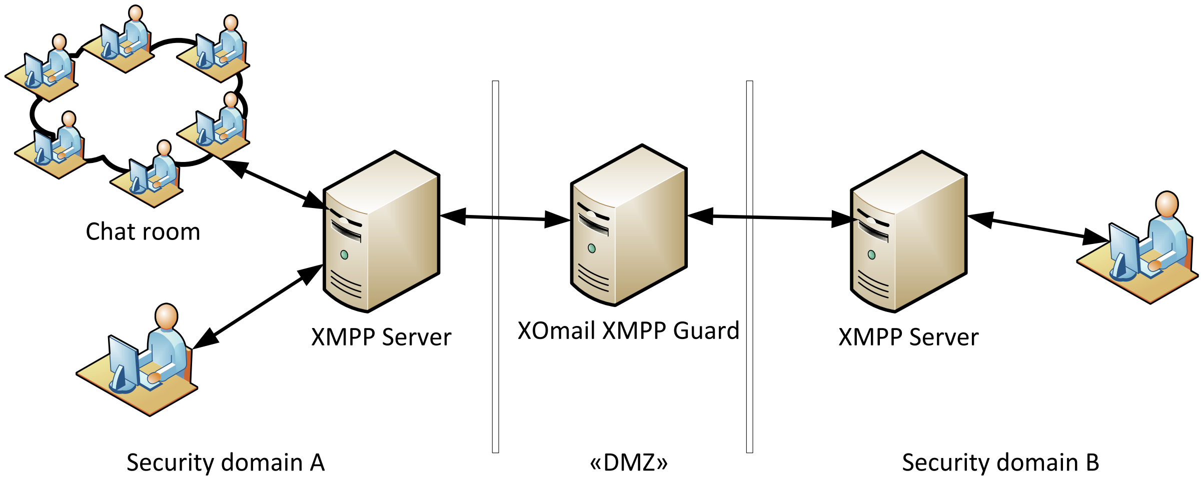

Figure 1-9 XMPP/Chat Guard example

As shown in Figure 1-9 the XMPP/Chat Guard allows direct messaging between XMPP addresses, as well as participating in chat rooms or groups across security domains. The Guard allows ease of use in common deployments, allowing use of standard chat clients and servers. Where higher assurance is required, the Guard may be configured to require that one or both security domains use chat clients with digital signature or multi-level security support.

1.5.7.4 Resource Guard

The XML/SOAP Guard of the TOE may be set up in a configuration to mediate access to one or more specific resources or services. In this scenario, the TOE protects a sensitive system, such as a database query service, from unintended information release or deliberate attacks. The Guard provides controlled access to the service to internal or external authorized users.

By delegating access control to the Guard, it is also possible to use COTS software as a storage or service backend, while maintaining high assurance access control requirements.

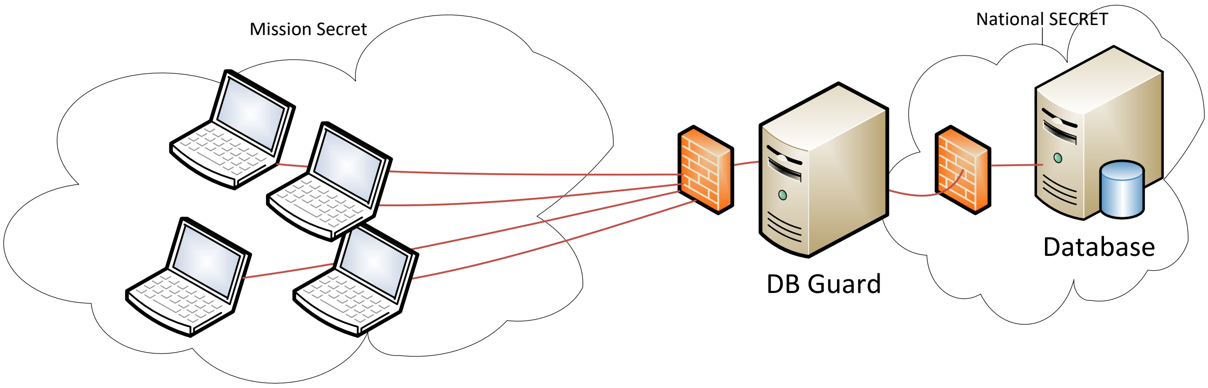

Figure 1-10 TOE deployed to protect database service

1.5.8 PROTECTING INTEGRITY OF INFORMATION OBJECTS

The TOE offers a configurable information flow control policy. The release decision for information objects is based on a range of attributes, including security labels, addresses or analysis by external content checkers.

The TOE supports controlling the information flow of both signed and unsigned information objects. This allows the TOE to be deployed in a wide range of network environments, also those where PKI systems have not been made available.

The configuration of the TOE must conform to the security requirements of the connected security domains. If digital signatures are disabled for a network, the TOE environment and associated network is responsible for providing an appropriate level of integrity protection to the information objects and flows.

Each connected security domain can be independently configured to require information objects to be signed or not. This will apply to all information objects within the given security domain. The infrastructure (certificate trust anchors, HSMs, CRL lookup, etc) is configured independently for each domain.

1.5.9 TOE ENVIRONMENT

1.5.9.1 Physical protection

The TOE environment is required to provide physical protection of the TOE, ensuring that only authorized personnel are allowed physical access to the TOE.

1.5.9.2 Network protection

The TOE environment is required to protect the network, reducing or eliminating the effect of low level network attacks and DoS attacks, complementing the protection mechanisms offered by the Guard.

The TOE provides three groups of interfaces, which must be protected by the TOE environment as specified:

Interface	**Proc. unit**	**Usage**	**Requirements for TOE Environment**
Traffic interfaces	A, B	Information objects to be transferred by the Guard. LDAP directory lookup Certificate Revoction List retrieval NTP time updates	Border protection devices monitoring for and preventing low level network attacks, including DoS, and ensuring only intended hosts may communicate with the TOE.
Service interfaces	A, B	Access to content inspection servers via	Network access to the TOE is limited to the specified services only.

Copyright © THALES Norway AS

		ICAP. Access to HSM via PKCS#11	
Local and remote management interfaces	Core	Administrative interface via ssh Syslog audit export	The TOE Environment ensures access to the interface is limited to TOE administrators only.

1.5.9.3 Hardware

The TOE runs on the hardware configuration presented in ch. 1.4.3.3.

The A and B units are connected to the Core unit, ensuring the TOE Security Policy cannot be physically circumvented.

The processing units are based on the Intel processor architecture. Each processing unit provides a processor with support for hardware virtualization, separate network controllers, and disk storage on each unit.

In addition, the Core unit has two 10Gbit Ethernet interfaces, providing dedicated connections towards unit A and B.

For the Kontron hardware platform the following additional constraints are required:

The internal Ethernet interface connections are physically realized using a custom VPX backplane. The customization of the backplane ensures that unit A cannot communicate directly with unit B and vice versa.

The hardware shall provide an ability to enable write protection for the NVRAM used for the BIOS executable and configuration data.

Prior to BIOS startup, the BIOS executable code and configuration data is validated against a checksum stored in protected memory.

1.5.9.4 UEFI BIOS

An UEFI BIOS is installed on the hardware, and is required to provide the following security functionality:

- Secure boot: Keys for secure boot provided by the hardware manufacturer are removed. Secure boot is enabled, and configured with keys for validating the TOE prior to boot.
- Password: The BIOS is password protected.
- Boot order: S-ATA disks only.

1.5.9.5 PikeOS

The TOE is built and tightly integrated with PikeOS. PikeOS is configured with a specific set of partitions, communication paths, and hardware resource access controls. PikeOS is required to protect these.

1.5.9.6 Border Protection Devices

External Border Protection Devices (BPDs) are required to counter more advanced or low level network attacks via the traffic interfaces. It is assumed that the Guard is protected using a BPD for each traffic interface. It may be possible to omit the BPD if a traffic interface connects to another guard (such as the case in Figure 1-6).

1.5.9.7 LDAP server

The LDAP server serves the Guard with address-information for address validation, and it serves the Guard with intermediate certificates and CRLs. If the TOE is configured to use LDAP, the LDAP server(s) must be available for the TOE to operate normally.

Note that CRLs may alternatively be retrieved using HTTP/HTTPS.

1.5.9.8 HTTP/HTTPS server (optional)

The HTTP/HTTPS server serves the Guard with CRLs. If CRL lookup through HTTPS is configured, the HTTP/HTTPS server must be available during normal operations.

CRLs may alternatively be retrieved using LDAP.

1.5.9.9 Hardware Security Module (HSM) (optional)

The Hardware Security Modules holds private keys used by the Guard. The Guard depends on the HSMs for proper operation when information objects are digitally signed, and uses PKCS#11 for all operations towards the HSMs [PKCS11].

2. CONFORMANCE CLAIMS (ASE_CCL)

2.1 CC CONFORMANCE CLAIM

Conformance	Common Criteria for Information Technology Security Evaluation Part 2 conformant: Security Functional Components [CCPART2]. Part 3 conformant: Security Assurance Components [CCPART3]
Assurance level	EAL4 augmented with: • ALC_FLR.3 (Systematic flaw remediation) • AVA_VAN.4 (Methodical vulnerability analysis)

2.2 PP CONFORMANCE CLAIMS

The Security Target has no Protection Profile claims.

We would like to acknowledge the Protection Profile for the NATO High Assurance ABAC Guard as an important reference in the development of the Guard.

3. SECURITY PROBLEM DEFINITION (ASE_SPD)

3.1 ASSUMPTIONS

The following conditions are assumed to exist in the operational environment.

A.APPROVED_CRYPTO

The TOE uses an approved cryptographic module (HSM) with approved methods for key management (i.e.; generation, access, distribution, destruction, handling, and storage of private keys), and for cryptographic operations (i.e.; signature and random number generation services).

A.APPROVED_PKI

The PKI is approved for use by the TOE in its intended environment and is approved for use with the TOE.

The TOE may use a single PKI for signing information objects released to both security domains, or a separate PKI for each security domain. The TOE may use one HSM for both domains, or separate HSMs for each domain. The TOE Environment ensures an appropriate configuration is used which complies with the requirements in each security domain.

A.CORRECT_CONFIGURATION

It is assumed that a properly trained and trusted individual will create configuration vectors that correctly represents the information flow policy intended for the TOE.

A.NETWORK_PROTECTED

The TOE environment ensures the networks or computers connected to the TOE is provided with appropriate security measures commensurate with the value of the IT assets protected by the TOE. The TOE environment prevents attackers from directly accessing the service and management interfaces of the TOE.

A.PHYSICAL_ACCESS_MANAGED

The TOE is located in a restricted or monitored environment that provides protection from unmanaged access to the physical components and data interfaces of the TOE. The non-IT environment provides the TOE with appropriate physical security commensurate with the value of the IT assets protected by the TOE.

A.TRUSTED_AND_TRAINED_ADMIN

System Administrators are authorized for access to the information to be handled by the Guard and the network where the Guard is placed. System Administrators are aware of the security policies and procedures of their organization, are trained and competent to follow the manufacturer’s guidance and documentation, and correctly configure and operate the TOE in accordance with those policies and procedures.

A.TRUSTED_LABELLER

A labeller is trusted to only create content labels in accordance with the organisation’s policy commensurate with the value of the information that he can create labels for.

3.2 THREATS

3.2.1 GENERAL

This section identifies the assets, threat agents and threats.

3.2.2 IDENTIFICATION OF ASSETS

The following assets are to be protected:

ASSET.COMMUN_OBJ_CONT

IPC object content is the content of an Inter-Process Communication object and is exchanged (received/read and sent/written) between TOE components.

This is a subset of ASSET.NON_EXPORTED_RESOURCE.

ASSET.INFO_OBJECT

Information Objects to be considered for mediation by the TOE, including its payload, set of headers, attributes and optional signature. If present, a signature ensures the integrity of the object’s payload and provides a strong binding between the Information Object and associated attributes, including originator.

The set of attributes that are required to be present depends on the active configuration vector. Some attributes may be optional, but are assigned a configurable value by the TOE when the Information Object is received.

ASSET.RELEASABLE_INFORMATION

Information stored in one of the connected security domains and labelled with content properties, based on which its release through the TOE is allowed.

ASSET.NON_RELEASABLE_INFORMATION

Information that belongs to one of the connected security domains that has an associated set of attributes such that it cannot be released through the TOE.

ASSET.EXPORTED_RESOURCE

Consumable data (containing ASSET.RELEASABLE_INFORMATION) or a service in one security domain that, according to the Guard security policy, can be accessed by a subject in the other security domain.

ASSET.NON_EXPORTED_RESOURCE

Consumable data or a service in on security domain that cannot be accessed by a subject in the other security domain.

ASSET.RELEASE_EVIDENCE

Audit records that are generated by the TOE as evidence of an information release process of the TOE.

ASSET.SYSTEM_RESOURCES

Resources that can be consumed by subjects in one of the interconnected security domains through use of the TOE external interfaces, e.g. system memory, persistent storage, processing time.

ASSET.TSF_INTERNALS — Comprises TSF data (data for the operation of the TOE upon which the enforcement of the SFR relies) and executable code.

3.2.3 IDENTIFICATION OF THREAT AGENTS

The threats and threat agents met by the TOE are diverse and depend on the scenario where it is deployed. The TOE is designed to mediate traffic between two security domains, and protects itself from attacks equally from both domains. Management traffic and supporting services are partitioned into own network segments, effectively reducing available attack vectors.

TA.INTERNAL — An attacker connected to the remote management interface or one of the service interfaces.

TA.USER — An attacker with the ability to interact with one of the traffic interfaces, i.e an authorized user in one of the connected security domains. These attackers are diverse and may be further classified into the following groups:

- Authenticated users on a controlled network, with user attributes that are compatible with the release policy of the TOE. These users are trusted to assign the correct label to all information objects to be released, but may still attempt other attacks towards the TOE or attempt to disclose information through covert channels.

- Authenticated users on a controlled network, with user attributes that are incompatible with the release policy of the TOE. These users are not permitted to provide information object for release by the TOE, and may attempt attacks on the TOE itself or try to have information objects released.

TA.EXTERNAL — Attackers attempting to reach the TOE through a traffic interface. These attackers have the intent to divulge classified information, from the TOE itself or its connected networks, or prevent operation of the TOE. The attacker may be an authorized user in one of the connected security domains. These attackers may have unlimited resources.

TA.ATTACKER — An attacker that may interact with any of the traffic interfaces, service interfaces or remote management interface. (TA.INTERNAL, TA.USER or TA.EXTERNAL)

TA.SYSTEM_ERROR — Hardware or software errors may cause faults during operation. Administrators may accidentally introduce errors when installing or updating the TOE improperly.

3.2.4 THREATS

The specific threats to the Guard are: existence of a *covert channel*, *information leakage*, *network attack*, *network reconnaissance*, *tampering with content labels, tampering with user, terminal and environment attributes, and unauthenticated access.*

T.ADMIN_MASQUERADE	TA.ATTACKER may masquerade as an administrator on the remote management interface in order to gain unauthorized access to

Copyright © THALES Norway AS

	ASSET.NON_EXPORTED_RESOURCE or ASSET.EXPORTED_RESOURCE via the traffic interfaces.
T.AUDIT_COMPROMISE	TA.ATTACKER may view, delete or modify ASSET.RELEASE_EVIDENCE, or prevent future ASSET.RELEASE_EVIDENCE from being recorded, thus masking an authorized subject's action, disclosing sensitive information or masking an attacker's activities.
T.OBJECT_TAMPERING	TA.USER may modify ASSET.INFO_OBJECT to make the TOE release ASSET.NON_RELEASABLE_INFORMATION. Example: A TA.USER that is not allowed to release information through the TOE may append a different TA.USER's address or certificate to an information object to circumvent the TOE release policy.
T.COVERT_CHANNEL	TA.USER may initiate an illicit flow of ASSET.NON_RELEASABLE_INFORMATION from the internal security domain to the external security domain as a result of exploiting a covert channel in the TOE.
T.DOS	TA.ATTACKER may block others from ASSET.SYSTEM_RESOURCES via a resource exhaustion attack.
T.INFORMATION_LEAK	TA.EXTERNAL may carry out a network-based attack against a traffic interface or released objects in order to obtain ASSET.NON_RELEASABLE_INFORMATION.
T.INSECURE_STATE	The TOE may be placed in an insecure state as a result of an administrative error during installation or configuration, a fault during installation, (re-)configuration, initialization or during change of mode of operation or as a result of an unsuccessful recovery from a system failure or discontinuity (TA.SYSTEM_ERROR affecting ASSET.TSF_INTERNALS).
T.MALWARE_INJECTION	A malicious agent in one of the connected security domains (TA.USER, TA.EXTERNAL) may attempt to introduce active content to a network through the TOE, whereby the active content can carry out or trigger actions automatically without an authorized subject in the internal domain directly or knowingly invoking the actions thereby compromising ASSET.RELEASABLE_INFORMATION or ASSET.NON_RELEASABLE_INFORMATION.

T.METADATA_LEAK	TA.EXTERNAL may carry out a network-based attack against a traffic inerface or released objects in order to obtain ASSET.NON_RELEASABLE_INFORMATION as metadata attached to a released information object. The metadata may disclose potentially compromising information regarding a security domain, such as network and organizational structure, security policies, addresses and directory structure, and which IT systems are in use.
T.NETWORK_ATTACK	TA.EXTERNAL may carry out a network-based attack against ASSET.EXPORTED_RESOURCE (e.g., by sending invalid messages to circumvent the TOEs security policy or by introducing viruses or active content into a security domain through the TOE) thereby compromising or affecting the availability of ASSET.RELEASABLE_INFORMATION or ASSET.NON_RELEASABLE_INFORMATION.
T.RECONNAISSANCE	TA.EXTERNAL may obtain ASSET.NON_RELEASABLE_INFORMATION about resources (e.g. IP addresses, port numbers, system names, system date/time, products, versions) from a security domain e.g. by using network scanning techniques, network traffic monitoring, etc.
T.RESIDUAL_DATA	TA.ATTACKER may gain unauthorized access to ASSET.RELEASABLE_INFORMATION, ASSET.NON_RELEASABLE_INFORMATION or ASSET.NON_EXPORTED_RESOURCE through reallocation of TOE resources between processes or partitions on the TOE. Errors in external services may also cause data from one information object to leak into other data structures.
T. TSF_COMPROMISE	TA.ATTACKER may cause ASSET.TSF_INTERNALS to be inappropriately accessed (viewed, modified, executed or deleted). The attack may be performed before the TOE is made operational (during delivery), during configuration, normal operation or maintenance (patches).
T.UNATTENDED_ADMIN_SESSION	TA.INTERNAL may gain unauthorized access to an unattended administrator session, thereby potentially gaining access to ASSET.RELEASE_EVIDENCE, change configuration (ASSET.TSF_INTERNALS) or disable the TOE.
T.UNAUTHORIZED_ACCESS	A TA.EXTERNAL may gain access to ASSET.NON_EXPORTED_RESOURCE or ASSET.EXPORTED_RESOURCE.

T.UNNOTICED_ATTACK	Due to an insufficient audit configuration, the administrator may not have ability to notice potential security violations by TA.ATTACKER that could compromise any asset, thus limiting the administrator's ability to identify and take action against a possible security breach.

3.3 ORGANISATIONAL SECURITY POLICIES

Organisational security policies (OSPs) are security rules, procedures, or guidelines imposed on the operational environment by the organisation running the TOE.

P.ACCOUNTABILITY	The TOE shall provide the capability to make available information regarding the occurrence of security relevant events. The authorized subjects of the TOE shall be held accountable for their actions within the TOE.
P.CLASSIFICATION	The system must limit the access to information based on CPR attributes included in a label and the information flow policy as defined in the TOE. The access rules enforced shall prevent a subject from accessing information which is of higher sensitivity than it is operating at and prevent a subject from causing information from being downgraded to a lower sensitivity.
P.CRYPTOGRAPHY	The TOE shall use approved and validated methods for cryptographic operations, (i.e. signature validation, and hashing.
P.MINIMAL_POSTURE	The Administrator shall ensure that only strictly required services and applications are running on the TOE and in the external services connected to the TOE (i.e. HSM, content inspector).

Copyright © THALES Norway AS

4. SECURITY OBJECTIVES (ASE_OBJ)

4.1 TOE IT SECURITY OBJECTIVES

O.ACCESS	The TOE will ensure that external subjects (services connected to the TOE service interfaces, i.e. HSM and content inspection service) only gain access to those ASSET.EXPORTED_RESOURCES for which they are authorized.
O.AUDIT	The TOE will provide the capability to detect, generate, export and review the audit trail (ASSET.RELEASE_EVIDENCE) for security relevant auditable events.The TOE provides secure storage for the audit trail.
O.CMD_ACL	The TOE provides means for restricting access to administrative commands for each System Administrator. Administrator access is further restricted according to role (determined by local or remote interface).
O.CONFIGURATION_CHANGE	The TOE will support the capability to perform a static configuration change. Reconfiguration is performed while the TOE is in a non-operational management mode, and takes effect once the TOE re-enters operational state. The TOE validates configuration vectors prior to use.
O.CORRECT_TSF_OPERATION	The TOE will provide a capability to test the TSF to ensure the correct operation of the TSF in its operational environment. The TOE will provide a runtime self-test capability. The TOE will provide the means for an administrator to invoke and obtain the results of the self-test. The TOE will take action in response to any failure of a runtime self-test capability.
O.INTERNAL_LEAST_PRIVILEGE	The entire TSF will be structured to achieve the principle of least privilege among TSF modules.
O.LABEL_MAPPING	The TOE supports mapping of security labels from one security policy to another, as specified by the configured Security Policy definitions. The TOE supports to bind both the translated and original versions of the originator-issued security label to the information object. This objective supports the IT implementation of bilateral agreements on protection of shared classified information.

O.MAC

The TOE enforces Mandatory Access Control according to the Bell LaPadula security policy.

O.MGMT_MODE

The TOE shall provide a mode from which the configuration vector can be changed, recovery or initial start-up procedures can be performed.

O.CONTROLLED_INFORMATION_FLOW

The TOE shall control the flow of information between the security domains by only relaying Information Objects and their associated metadata that are allowed as part of the information flow policy.

Example: The Information Flow Policy may combine a number of rules, such as allowed originator and recipient addresses, required attribute filters and mapping, content inspection (external services), and digital signature requirements.

O.MINIMAL_PROXY

The TOE shall provide mechanisms that can be used to limit the amount of information which is transmitted between the security domains in the headers or attributes of released Information Objects.

Examples are: Map addresses and identifiers, discard trace information, and rebuild information objects to a normalized standard.

O.OBJECT_INTEGRITY

The TOE ensures digital signatures and certificates associated with information objects are validated, and ensures released information objects are signed, as specified by configuration.

O.RESIDUAL_INFORMATION

The TOE will ensure that any information contained in an ASSET.EXPORTED_RESOURCE is not released when the resource is reallocated; this includes that no residual information from a previously relayed message is transmitted.

O.RESOURCE_SHARING

The TOE shall provide mechanisms that enforce constraints on the allocation of ASSET.EXPORTED_RESOURCES and mitigate attempts to exhaust these resources (e.g., resulting in denying access to ASSET.EXPORTED_RESOURCE).

O.SECURE_STATE

The TOE will preserve secure state during an execution session.

The TOE will provide startup mechanisms to transition the TSF from offline state to an initial secure state without protection compromise.

The TOE will provide procedures and/or mechanisms, which can be used in the event of failure, faults, or discontinuity, to preserve secure state and to transition the TSF back to a secure state without protection compromise.

O.SUBJECT_ISOLATION

The TOE will provide mechanisms to protect each subject within the TOE from unauthorized interference by other subjects.

O.TRANSITION

The TOE will provide the capabilities for an authorized subject to restart the TOE, halt the TOE and transition the TOE into management mode or active mode.

O.TSF_INTEGRITY

The TOE will verify the integrity of ASSET.TSF_INTERNALS.

O.VALID_LABEL

The TOE shall validate any recognized security labels in the information objects. The labels must be valid and consistent with the Security Policy of the TOE. Data objects that arrive without a label from a system high domain may be assigned a configurable fixed label during import.

Table 4-1 TOE Security Objectives

4.2 TOE ENVIRONMENT SECURITY OBJECTIVES

OE.BORDER_PROTECTION

The OE provides Border Protection Devices in front of the TOE. These shall inspect traffic towards the TOE, countering attacks from the network. The OE must manage and update the Border Protection Devices to counter current threats.

Note: The OE must consider how to implement the protection services, such as malware protection and antivirus scanners. Services operating on information contained in objects relayed by the TOE may either be managed by the Border Protection Devices or be attached as to the TOE as a content inspection service.

OE.CONTENT_INSPECTION_SE
RVICE

The OE provides one or more content inspection services for use by the TOE if configured. The content inspection service will detect and give feedback to the TOE to support a release decision for the object being inspected.

Additionally, authorized users in the OE shall ensure the content inspection service configuration is kept current. If the service includes antivirus and malware protection, the OE shall ensure updates for virus definitions, malware signatures etc are kept current.

OE.DIRECTORY_SERVICE

The OE provides directories for completing and translating addresses, X.509 intermediate certificates, and CRLs.

OE.APPROVED_CRYPTO

The OE provides an approved cryptographic module (HSM) with approved methods for key management (i.e.; generation, access, destruction, handling, and storage of keys), and for cryptographic operations (i.e.; encryption, decryption, signature creation and validation, and random number generation services).

OE.APPROVED_PKI

The OE provides an approved PKI service, with a trusted root anchor, certificates for users and systems, with valid certificate chains. The TOE maintains the PKI, ensures that the PKI components are kept updated, particularly CRLs. The OE provides private keys for storage in a HSM for use by the TOE.

OE.MINIMAL_POSTURE

The Administrator shall ensure that only strictly required services and applications are running in the environments connected to the administrator interfaces and service interfaces.

OE.NETWORK

The TOE Environment will ensure the network used for the security domains are protected according to the sensitivity and integrity protection required for the information contained within the domains.

OE.PHYSICAL_ACCESS_MANAGED

The TOE is located in a restricted or monitored environment that provides protection from unmanaged access to the physical components and data interfaces of the TOE.

Physical security will be provided for the TOE by the non-IT environment commensurate with the value of the IT assets protected by the TOE.

OE.CONFIGURATION

A properly trained and trusted individual will create configuration vectors for the TOE that correctly reflects the environment's requirements for controlling the flow of information between the connected security requirements.

OE.TRUSTED_AND_TRAINED_ADMIN

Sites using the TOE will ensure that administrators are trusted and aware of the security policies and procedures of their organization, are trained and competent to follow the manufacturer's guidance and documentation, and correctly configure and operate the TOE in accordance with those policies and procedures.

OE.TRUSTED_LABELLER

A labeller is trusted to only assign attributes to Information Objects (ASSET.INFO_OBJECT), including the Security Label, in accordance with applicable security policies and their respective guidelines.

The assurance of the label creation process must be commensurate with the value of the information that the labels are created for.

Copyright © THALES Norway AS

OE.PLATFORM	The OE provides the required separation kernel and hardware platform, consisting of three instances of PikeOS, running on individual processing units, connected internally. The processing units provide the fundamental tools for isolating internal components using the Intel processor’s hardware virtualization support.
OE.TIME_SOURCE	The OE provides a NTP time service.

Table 4-2 Security objectives for the TOE Environment

5. EXTENDED COMPONENTS DEFINITION (ASE_ECD)

5.1 EXTENDED COMPONENTS DEFINITION (SFRS)

N/A

6. SECURITY REQUIREMENTS (ASE_REQ)

6.1 SECURITY FUNCTIONAL REQUIREMENTS (SFRS)

6.1.1 CLASS FAU: SECURITY AUDIT

6.1.1.1 Security audit automatic response (FAU_ARP)

6.1.1.1.1 FAU_ARP.1 Security alarms

Hierarchical to: No other components.

Dependencies: FAU_SAA.1 Potential violation Analysis (not satisfied, see 8.2.2).

FAU_ARP.1.1: The TSF shall take **the following actions** upon detection of a potential security violation **or any failure to the tests defined in FPT_TST.1.1**:

- *immediately raise an alarm for the potential security violations,*
- *make accessible the audit record contents associated with the auditable event(s) that generated the alarm, for:*

  a) local administrators;

  b) remote administrators;

  c) exported audit configuration

  d) [no other methods],

6.1.1.2 Security audit data generation (FAU_GEN)

6.1.1.2.1 FAU_GEN.1 Audit data generation

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps (not satisfied, see 8.2.2).

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 *basic* level of audit;

(c) All auditable events listed in Table 6-1;

FAU_GEN.1.2: The TSF shall record within each audit record at least the following information:

(a) Date and time of the event, type of event, subject identity (if applicable), 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 PP/ST, *information specified in column three of* Table 6-1.

Copyright © THALES Norway AS

Requirement	Auditable Events	Additional Audit Record Contents
FAU_ARP.1	Actions taken due to potential security violations..	Identification of what caused the generation of the alarm.
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_SAA.1	Enabling and disabling of any of the analysis mechanisms Automated responses performed by the tool.	None.
FAU_SAR.1	Reading of information from the audit records.	None.
FAU_SAR.2	Unsuccessful attempts to read information from the audit records.	None.
FAU_STG.1	None.	None.
FAU_STG.3	Actions taken due to exceeding of a threshold.	None.
FAU_STG.4	Actions taken due to the audit storage failure.	None.
FCO_NRO.1	The invocation of the non-repudiation service. Identification of the information, the destination and a copy of the evidence provided. *Note: The following auditable event from CC Part 2 is not applicable, as the non-repudiation service is invoked by system configuration:* *The identity of the user who requested that evidence of origin would be generated.*	None.
FCS_COP.1	Success and failure, and the type of cryptographic operation. Any applicable cryptographic mode(s) of operation, subject attributes and object attributes.	None.
FDP_ACF.1	All requests to perform an operation on an object covered by the SFP.	None.

Copyright © THALES Norway AS

Requirement	Auditable Events	Additional Audit Record Contents
FDP_ETC.2	Successful export of information.	None.
FDP_IFC.2	None.	None.
FDP_IFF.1	All decisions on requests for information flow.	None.
FDP_IFF.2	All decisions on requests for information flow.	None.
FDP_ITC.2	All attempts to import user data, including any security attributes.	None
FDP_RIP.2	None.	None.
FDP_UIT.1	The identity of any user or subject attempting to use the user data exchange mechanisms, but who is unauthorised to do so.	None.
FIA_ATD.1	None.	None.
FIA_UID.2	All use of the user identification mechanism, including the user identity provided.	None.
FMT_MOF.1	All modifications in the behaviour of the functions in the TSF.	None.
FMT_MSA.1	All modifications of the values of security attributes.	None.
FMT_MSA.3	Modifications of the default setting of permissive or restrictive rules. All modifications of the initial values of security attributes.	None.
FMT_MTD.1	All modifications to the values of TSF data.	None.
FMT_MTD.3	Rejection of specified values for TSF data.	All rejected values of TSF data.
FMT_REV.1	All attempts to revoke security attributes.	None.
FMT_SMF.1	Use of the management functions.	
FMT_SMR.2	The role associated with each administrator session. Note: Tailored from CC Part 2 definition.	None.
FPT_FLS.1	Failures detected by the FMT_TST.1 tests. Other TSF failures specified in the assignment	None.

Copyright © THALES Norway AS

Requirement	Auditable Events	Additional Audit Record Contents
	statement of FPT_FLS.1.1-b.	
FPT_RCV.4	The inability to return to a secure state after a failure of the TSF. The detection of a failure of a security function.	None.
FPT_TST.1	Execution of the TSF self-tests and the results of the tests.	None.
FRU_PRS.1	None. Note: Minimal and Basic audit requirements exist for this SFR, but are considered not relevant for the TOE, as the service cannot be rejected or fail. However, the TOE audits the priority level of each handled Information Object.	None.
FTP_TRP.1	All attempted uses of the trusted path functions. Identification of the user associated with all trusted path invocations, if available.	Object attributes: Originator, recipient, security label.

Table 6-1 Auditable Events

6.1.1.2.2 FAU_GEN.2 User identity association

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation,
FIA_UID.1 Timing of identification.

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

Application note: Auditing of processing information objects: “Users” may be identified as originators or recipients of information objects.

6.1.1.3 Security audit analysis (FAU_SAA)

6.1.1.3.1 FAU_SAA.1 Potential violation analysis

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation.

FAU_SAA.1.1: The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the enforcement of the SFRs.

FAU_SAA.1.2: The TSF shall enforce the following rules for monitoring audited events:

Copyright © THALES Norway AS

(a) Accumulation or combination of the following events known to indicate a potential security violation:

- *Multiple failed administrator logon events.*
- *Configurable number of Information Flow policy violations from an individual originator address within an administrator-specified time period;*

(b) [none].

6.1.1.4 Security audit review (FAU_SAR)

6.1.1.4.1 FAU_SAR.1 Audit review

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation.

FAU_SAR.1.1: The TSF shall provide *the System Administrators* with the capability to read *all audit data* from the audit records.

FAU_SAR.1.2: The TSF shall provide the audit records in a manner suitable for **the *System Administrators*** to interpret the information.

6.1.1.4.2 FAU_SAR.2 Restricted audit review

Hierarchical to: No other components.

Dependencies: FAU_SAR.1 Audit review.

FAU_SAR.2.1: The TSF shall prohibit all users read access to the audit records in the audit trail, *except* ***the System Administrators***.

6.1.1.5 Security audit event storage (FAU_STG)

6.1.1.5.1 FAU_STG.1 Protected audit trail storage

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation.

FAU_STG.1.1: The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

FAU_STG.1.2: The TSF shall be able to *prevent* unauthorised modifications to the stored audit records in the audit trail.

6.1.1.5.2 FAU_STG.3 Action in case of possible audit data loss

Hierarchical to: No other components.

Dependencies: FAU_STG.1 Protected audit trail storage.

FAU_STG.3.1: The TSF shall *generate an alarm* if the audit trail exceeds *a configurable percentage of storage capacity*.

6.1.1.5.3 FAU_STG.4 Prevention of audit data loss

Hierarchical to: FAU_STG.3 Action in case of possible audit data loss.

Dependencies: FAU_STG.1 Protected audit trail storage.

FAU_STG.4.1: The TSF shall *prevent audited events, except those taken by the authorised user with special rights* and *transition to management mode* if the audit trail is full.

6.1.2 CLASS FCO: COMMUNICATION

6.1.2.1 Non-repudiation of origin (FCO_NRO)

6.1.2.1.1 FCO_NRO.1 Selective proof of origin

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification.

FCO_NRO.1.1: The TSF shall be able to generate evidence of origin for transmitted *Information Objects* ~~at the request of~~ **when enabled by configuration.**

FCS_NRO.1.2: The TSF shall be able to relate the *subject name and address* of the originator of the information, and the *object's digital signatures, including S/MIME SignedAttributes and the information object payload* of the information to which the evidence applies.

FCO_NRO.1.3: The TSF shall provide a capability to verify the evidence of origin of information to *the message releaser and recipient* given *that a message signature is present*.

6.1.3 CLASS FCS: CRYPTOGRAPHIC SUPPORT

6.1.3.1 Cryptographic operation (FCS_COP)

6.1.3.1.1 FCS_COP.1 Cryptographic operation (cryptographic hashing)

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 or FDP_ITC.2, or FCS_CKM.1]
FCS_CKM.4.

FCS_COP.1.1: The TSF shall perform *cryptographic hashing services* in accordance with a specified cryptographic algorithm *SHA-2*~~and cryptographic key sizes [assignment: cryptographic key sizes]~~ and **message digest sizes 256, 384 and 512** bits that meet the following: *FIPS PUB 180-4 [FIPS-180]*.

6.1.4 CLASS FDP: USER DATA PROTECTION

6.1.4.1 Access control policy (FDP_ACC)

6.1.4.1.1 FDP_ACC.2 Complete Access Control

Hierarchical to: FDP_ACC.1 Subset access control

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.2.1: The TSF shall enforce the *Management Function SFP* on *local and remote administrators, and configurable system parameters including configuration vectors,* and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2: The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

6.1.4.2 Access control functions (FDP_ACF)

6.1.4.2.1 FDP_ACF.1 Security Attribute Based Access Control (Command Access)

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control

Copyright © THALES Norway AS

FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1: The TSF shall enforce the *Management Function SFP* to objects based on the following: *subject identifier, subject attribute command access rights, object identifier (management function)*.

FDP_ACF.1.2: The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *The System Administrator has a command access attribute that allows the specific management operation*.

FDP_ACF.1.3: The TSF shall explicitly authorize access of subjects to objects based on the following additional rules:

a) Access to management functions for Local and Remote System Administrators are restricted according to Table 6-2*.*

FDP_ACF.1.4: The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *none*.

6.1.4.3 Export from the TOE (FDP_ETC)

6.1.4.3.1 FDP_ETC.2 Export of user data with security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control,
or FDP_IFC.1 Subset information flow control]

FDP_ETC.2.1: The TSF shall enforce the *Object Release SFP* when exporting user data, controlled under the SFP(s), outside of the TOE.

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 TOE, 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 TOE:

a) Convert security label from internal representation into the representation required by the export medium.

6.1.4.4 Information flow control policy (FDP_IFC)

6.1.4.4.1 FDP_IFC.2/r Complete information flow control (Object Release SFP)

Hierarchical to: FDP_IFC.1 Subset information flow control

Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.2.1/r: The TSF shall enforce the *Object Release SFP based on content-based protection requirements and release conditions* on *all information objects mediated by the TOE between the two connected domains* and all operations that cause that information to flow to and from subjects covered by the SFP.

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

6.1.4.4.2 FDP_IFC.2/i Complete information flow control (Internal Flow Control SFP)

Hierarchical to: FDP_IFC.1 Subset information flow control

Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.2.1/i: The TSF shall enforce the *Internal Flow Control SFP* on *all inter-process communication* and all operations that cause that information to flow to and from subjects covered by the SFP.

Copyright © THALES Norway AS

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

6.1.4.5 Information flow control functions (FDP_IFF)

6.1.4.5.1 FDP_IFF.1/r Simple security attributes (Object Release SFP)

Hierarchical to: No other components.

Dependencies: FDP_IFC.1 Subset information flow control

FMT_MSA.3 Static attribute initialisation

FDP_IFF.1.1/r: The TSF shall enforce the *Object Release SFP* based on the following types of subject and information security attributes:

- *object release direction,*
- *object originator address and destination addresses,*
- *content inspection service response,*
- *protocol filter response*
- *digital signature validation status*
- *object security label*
- *original security label*

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

- The operation is allowed by the policy defined by the active configuration vector, including MAC validation (FDP_IFF.2).

FDP_IFF.1.3/r: The TSF shall enforce the additional information flow rules: *Objects are processed according to priority*. High priority objects use reserved resources.

FDP_IFF.1.4/r: The TSF shall explicitly authorise an information flow based on the following rules: *none*

FDP_IFF.1.5/r: The TSF shall explicitly deny an information flow based on the following rules: *none*

6.1.4.5.2 FDP_IFF.1/i Simple security attributes (Internal Flow Control Policy)

Hierarchical to: No other components.

Dependencies: FDP_IFC.1 Subset information flow control

FMT_MSA.3 Static attribute initialisation

FDP_IFF.1.1/i: The TSF shall enforce the *Internal Flow Control SFP* based on the following types of subject and information security attributes:

- Subject security attributes 'subject identity' (process id, partition id)
- Information security attributes: destination process, destination partition, priority level

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

- *The operation is allowed by the communication flows defined between TOE processes.*

FDP_IFF.1.3/i: The TSF shall enforce the additional information flow rules: *Information is processed according to priority*.

FDP_IFF.1.4/i: The TSF shall explicitly authorise an information flow based on the following rules: *none*

FDP_IFF.1.5/i: The TSF shall explicitly deny an information flow based on the following rules: *none*

6.1.4.5.3 FDP_IFF.2 Hierarchical security attributes (Object Release SFP)

Hierarchical to: FDP_IFF.1 Simple Security Attributes

Dependencies: FDP_IFC.1 Subset information flow control
FMT_MSA.3 Static attribute initialisation

FDP_IFF.2.1: The TSF shall enforce the *Object Release SFP* based on the following types of subject and information security attributes: *subject security clearance (max HCL, NHC, SP), object hierarchical classification level (HCL), object non-hierarchical categories (NHC) and object security policy (SP).*

FDP_IFF.2.2: The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules, based on the ordering relationships between security attributes hold:

a) Read operation: Subject clearance must dominate object security label (S >= O).

b) Write operation: Object security label must dominate subject clearance (O >= S).

c) RW operation: Subject clearance must be equal to object security label (S == O).

FDP_IFF.2.3: The TSF shall enforce the *following additional flow control SFP rules: if configured, validate in addition the original security label according to the rules defined in FDP_IFF.2.2.*

FDP_IFF.2.4: The TSF shall explicitly authorise an information flow based on the following rules: *none*

FDP_IFF.2.5: The TSF shall explicitly deny information flow based on the following rules: *none*.

FDP_IFF.2.6: The TSF shall enforce the following relationships for any two valid information flow control security attributes:

a) There exists an ordering function that, given two valid security attributes, determines if the security attributes are equal, if one security attribute is greater than the other, or if the two security attributes are incomparable; and

b) There exists a ”least upper bound” in the set of security attributes, such that, given any two security attributes, there is a valid security attribute that is greater than or equal to the two valid security attributes; and

c) There exists a ”greatest lower bound” in the set of security attributes, such that, given any two valid security attributes, there is a valid security attribute that is not greater than the two valid security attributes.

6.1.4.6 Import from outside of the TOE (FDP_ITC)

6.1.4.6.1 FDP_ITC.2 Import of user data with security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or

Copyright © THALES Norway AS

FDP_IFC.1 Subset information flow control]

[FTP_ITC.1 Inter-TSF trusted channel or
FTP_TRP.1 Trusted path]

FPT_TDC.1 Inter-TSF basic TSF data consistency

FDP_ITC.2.1: The TSF shall enforce the *Object Release SFP* when importing user data, controlled under the SFP, from outside of the TOE.

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 TOE:

(a) *Missing security label:*
If the active configuration for the originating security domain requires labels to be present: Reject the release request.
If labels are optional, set the specified default (system high) label.

(b) *Security policy not allowed or security label outside defined label range for the originating security domain: Reject the information object.*

6.1.4.7 Residual information protection (FDP_RIP)

This family addresses the need to ensure that any data contained in a resource is not available when the resource is de-allocated from one object and reallocated to a different object.

6.1.4.7.1 FDP_RIP.2 Full residual information protection

Hierarchical to: FDP_RIP.1 Subset residual information protection.

Dependencies: No dependencies.

FDP_RIP.2.1: The TSF shall ensure that any previous information content of a resource is made unavailable upon the *allocation of the resource to* **all** objects.

6.1.4.8 Inter-TSF user data integrity transfer protection (FDP_UIT)

6.1.4.8.1 FDP_UIT.1 Data exchange integrity

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or

FDP_IFC.1 Subset information flow control]

[FTP_ITC.1 Inter-TSF trusted channel, or

FTP_TRP.1 Trusted path]

FDP_UIT.1.1: The TSF shall enforce the *Object Release SFP* to *transmit and receive* user data in a manner protected from *modification and insertion* errors.

FDP_UIT.1.2: The TSF shall be able to determine on receipt of user data, whether *modification or insertion* has occurred.

6.1.5 CLASS FIA: IDENTIFICATION AND AUTHENTICATION

6.1.5.1 User attribute definition (FIA_ATD)

6.1.5.1.1 FIA_ATD.1 User attribute definition

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_ATD.1.1: The TSF shall maintain the following list of security attributes belonging to individual users:

a) User identifier

b) User command access rights

c) Interface in use when accessing the TOE (local or remote)

6.1.5.2 User Identification (FIA_UID)

6.1.5.2.1 FIA_UID.2 User identification before any action

Hierarchical to: FIA_UID.1 Timing of identification

Dependencies: No dependencies.

FIA_UID.2.1: The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

6.1.6 CLASS FMT: SECURITY MANAGEMENT

6.1.6.1 Management of functions in TSF (FMT_MOF)

This family allows authorised users control over the management of functions in the TSF. Examples of functions in the TSF include the audit functions and the multiple authentication functions.

6.1.6.1.1 FMT_MOF.1/RSP Management of security functions behaviour (remote security policy change)

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles,
FMT_SMF.1 Specification of Management Functions.

FMT_MOF.1.1/RSP: The TSF shall restrict the ability to *modify the behaviour of* the functions *Information Flow Policy by changing the active configuration vector* to *Remote System Administrator and Local System Administrator*.

6.1.6.1.2 FMT_MOF.1/LSP Management of security functions behaviour (local security policy change)

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles,
FMT_SMF.1 Specification of Management Functions.

FMT_MOF.1.1/LSP: The TSF shall restrict the ability to *modify the behaviour of* the functions *Information Flow Policy by installing a new configuration vector to the Local System Administrator*.

6.1.6.1.3 FMT_MOF.1/AUD Management of security functions behaviour (Audit)

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles,

FMT_SMF.1 Specification of Management Functions.

FMT_MOF.1.1/AUD: The TSF shall restrict the ability to *modify the behaviour of* the functions *Security Alarms (FAU_ARP) and Audit (FAU_GEN)* to *the Local System Administrators*.

6.1.6.2 Management of security attributes (FMT_MSA)

6.1.6.2.1 FMT_MSA.1 Management of security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or

FDP_IFC.1 Subset information flow control]

FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions.

FMT_MSA.1.1: The TSF shall enforce the *Object Release SFP* to restrict the ability to *query, modify, or delete* the security attributes *TOE configuration vectors*, to *the System Administrators*.

6.1.6.2.2 FMT_MSA.3 Static attribute initialisation (restrictive rule set values)

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes,

FMT_SMR.1 Security roles.

FMT_MSA.3.1: The TSF shall enforce the *Object Release SFP* to provide *restrictive* default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2: The TSF shall allow *Local System Administrator (through configuration vectors)* to specify alternative initial values to override the default values when an object or information is created.

Application Note: No Information Objects are allowed to flow through the TOE by default, and a configuration vector must be loaded for the TOE to become operational.

6.1.6.3 Management of TSF data (FMT_MTD)

6.1.6.3.1 FMT_MTD.1 Management of TSF data

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles,

FMT_SMF.1 Specification of Management Functions.

FMT_MTD.1.1: The TSF shall restrict the ability to *perform the operations listed in* Table 6-2 on the *TSF data in the table* to *System Administrators according to the role (remote or local), and the user's command access attributes*.

6.1.6.3.2 FMT_MTD.3 Secure TSF data

Hierarchical to: No other components.

Dependencies: FMT_MTD.1 Management of TSF data.

FMT_MTD.3.1: The TSF shall ensure that only secure values are accepted for TSF data.
Application Note: Secure implies that the values are consistent (e.g. security classifications are consistent with the supplied security policy files), and are valid within the defined range for the TSF data (e.g., an audit enable/disable indicator must be within range of a Boolean type).

6.1.6.4 Revocation (FMT_REV)

6.1.6.4.1 FMT_REV.1 Revocation

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles.

FMT_REV.1.1: The TSF shall restrict the ability to revoke *configuration vectors and certificates* associated with the *TSF configuration*under the control of the TSF to *the Local and Remote System Administrators*.**FMT_REV.1.2**: The TSF shall enforce the rules:

- *apply changes to the configuration vector, immediately or for new objects depending on operator command;*
- *recognize revocation status of digital certificates following a CRL update, revoked trust anchor or expired validity period.*

6.1.6.5 Specification of Management Functions (FMT_SMF)

6.1.6.5.1 FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_SMF.1.1: The TSF shall be capable of performing the following management functions: *Functions specified in column “Function” in the table below.*

TSF data	Function	Operations	Allowed adm. roles	
			Local	Remote
Admin user accounts	Manage users (System Administrators) and command access	Query, Modify, Delete	Yes	
Audit data	Monitor audit trail and TOE status	Query	Yes	Yes
	Manage Audit trail	Configure deletion strategies, configure automated export	Yes	
Configuration vectors	Manage available configuration vectors	Upload, Delete	Yes	
	Set active configuration vector	Query, Set active	Yes	Yes
Certificates	Manage certificates	Query, Upload, Delete	Yes	
TOE run state	Change run state (restart, halt, management mode)	Query, Modify	Yes	Yes
All TSF data	Initiate emergency erase	Initiate	Yes	Yes

Table 6-2 Management functions and operations on TSF data

6.1.6.6 Security Management Roles (FMT_SMR)

6.1.6.6.1 FMT_SMR.2 Restrictions on security roles

Hierarchical to: FMT_SMR.1 Security roles.

Copyright © THALES Norway AS

Dependencies: FIA_UID.1 Timing of identification.

FMT_SMR.2.1: The TSF shall maintain the roles:

- *Remote System Administrator*
- *Local System Administrator*

Note: A System Administrator will be assigned different rights based on whether the local or remote management interface is used.

FMT_SMR.2.2: The TSF shall be able to associate users with roles.

FMT_SMR.2.3: The TSF shall ensure that the conditions:

- *All System Administrators shall be able to administer the TOE locally or remotely;*

are satisfied.

6.1.7 CLASS FPT: PROTECTION OF THE TSF

6.1.7.1 Fail secure (FPT_FLS)

The requirements of this family ensure that the TOE will always enforce its SFRs in the event of identified categories of failures in the TSF.

6.1.7.1.1 FPT_FLS.1 Failure with preservation of secure state

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_FLS.1.1: The TSF shall preserve a secure state when the following types of failures occur:

a) failures from self-tests covered by FPT_TST.1, including validation errors for configuration vectors and security policy files;

b) Non-responsive supporting services (e.g. HSM, directory server, content inspection service).

Application Note: TSF failure modes vary and may include "hard" failures such as those associated with hardware failure or unrecoverable software errors, and "soft" failures such as intermittent hardware errors and recoverable software errors.

Application Note: The TSF is not expected to protect itself against all types of hardware errors. For example, a radiation induced change of a single bit in a memory access control register could result in an incorrect (but valid) memory location being accessed. This would not always be detected by the hardware.

6.1.7.2 Trusted Recovery (FPT_RCV)

The requirements of this family ensure that the TSF can determine that the TOE is started up without protection compromise and can recover without protection compromise after discontinuity of operations. This family is important because the start-up state of the TSF determines the protection of subsequent states.

6.1.7.2.1 FPT_RCV.4 Function recovery

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_RCV.4.1: The TSF shall ensure that *TOE self-test, state transitions (e.g. management mode),* and *object release (success and failure)* have the property that the function either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state.

6.1.7.3 Inter-TSF TSF data consistency (FPT_TDC)

In a distributed environment, a TOE may need to exchange TSF data (e.g. the SFP-attributes associated with data, audit information, identification information) with another trusted IT product, This family defines the requirements for sharing and consistent interpretation of these attributes between the TSF of the TOE and a different trusted IT product.

6.1.7.3.1 FPT_TDC.1 Inter-TSF basic TSF data consistency

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TDC.1.1: The TSF shall provide the capability to consistently interpret *object security labels* when shared between the TSF and another trusted IT product.

FPT_TDC.1.2: The TSF shall use *the rules defined for the communication channel* when interpreting the TSF data from another trusted IT product.

6.1.7.4 TSF self-test (FPT_TST)

The family defines the requirements for the self-testing of the TSF with respect to some expected correct operation. These tests can be carried out at start-up, periodically, at the request of the authorised user, or when other conditions are met. The actions to be taken by the TOE as the result of self-testing are defined in other families.

6.1.7.4.1 FPT_TST.1 TSF testing

Hierarchical to: No other components.

Dependencies: No dependencies.

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

FPT_TST.1.2: The TSF shall provide authorised users with the capability to verify the integrity of *configuration vectors*.

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

6.1.8 CLASS FRU: RESOURCE UTILISATION

6.1.8.1 Priority of service (FRU_PRS)

6.1.8.1.1 FRU_PRS.1 Limited priority of service

Hierarchical to: No other components.

Dependencies: No dependencies.

FRU_PRS.1.1: The TSF shall assign a priority to each subject in the TSF.

FRU_PRS.1.2: The TSF shall ensure that each access to *TOE internal communication channels and reserved resources for flash/override traffic* shall be mediated on the basis of the subject's assigned priority.

Application Note: The SFR applies to the STANAG 4406 Guard and SMTP Guards. The XMPP/Chat and SOAP/XML protocols do not support priority attributes, and hence all subjects are assigned the same priority internally, and there are no reserved resources for Flash/Override.

6.1.9 CLASS FTP: TRUSTED PATH/CHANNELS

6.1.9.1 Trusted path (FTP_TRP)

6.1.9.1.1 FTP_TRP.1 Trusted Path

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_TRP.1.1: The TSF shall provide a communication path between itself and *remote* users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from *modification of Information Objects*.

FTP_TRP.1.2: The TSF shall permit *the TSF or remote users* to initiate communication via the trusted path.

FTP_TRP.1.3: The TSF shall require the use of the trusted path for *protection of Information Objects received by or sent from the TOE*.

Note: The requirement applies to Information Objects when protected by a digital signature, and is limited to the parts of an Information Object covered by the signature.

6.2 SECURITY ASSURANCE REQUIREMENTS (SARS)

The TOE is evaluated to EAL4 augmented with ALC_FLR.3 and AVA_VAN.4.

The security assurance requirements for the TOE are selected according to EAL4 augmented with ALC_FLR.3 (systematic flaw remediation) and AVA_VAN.4 (Methodical vulnerability analysis).

From CC Part 3:

> **EAL4** provides assurance by a full security target and an analysis of the SFRs in that ST, using a functional and **complete** interface specification, guidance documentation, **a** description of the **basic modular** design of the TOE, **and a subset of the implementation,** to understand the security behaviour.
>
> The analysis is supported by independent testing of the TSF, evidence of developer testing based on the functional specification and TOE design, selective independent confirmation of the developer test results, and a vulnerability analysis (based upon the functional specification, TOE design, **implementation representation, security architecture description** and guidance evidence provided) demonstrating resistance to penetration attackers with **a Moderate** attack potential.
>
> **EAL4** also provides assurance through the use of development environment controls **and additional** TOE configuration management **including automation,** and evidence of secure delivery procedures.

EAL4 is considered appropriate for the TOE when placed in an operational environment with the properties and policies described by the security problem definition. The security problem definition has been selected to apply to operational environments for classified networked information systems in military organizations.

The ALC_FLR.3 component has been included to provide assurance for the developer's procedures for handling and patching security flaws discovered in the TOE.

The AVA_VAN.4 component ensures that a methodical vulnerability analysis has been performed.

7. TOE SUMMARY SPECIFICATION (ASE_TSS)

7.1 TOE SECURITY FUNCTIONS

7.1.1 SF.ABAC

The TOE performs an extensive evaluation of all information objects and their associated attributes before release, using the active configuration vector to determine the scope of inspections performed:

- Evaluate originator and recipient
- Digital signature presence and validity
- Content checking
- MAC checking
- Filter response

The SF implements the SFP's defined by FDP_IFC.2/r, FDP_IFF.1/r and FDP_IFF.2.

7.1.2 SF.AUDIT

The TOE is able to generate audit records for the following events:

- All release decisions, for objects transferred by the TOE between the security domains, including direction and release decision
- All administrative operations
- Changes to operational state
- Results from TOE Self-Tests
- Changes to system time

The TOE allows the Audit log to be exported locally by a System Administrator, as well as being continuously exported to a remote service.

This implements FAU_ARP.1, FAU_GEN.1, FAU_GEN.2, FAU_SAA.1, FAU_SAR.1, FAU_SAR.2, FAU_STG.1, FAU_STG.3, FAU_STG.4. The SF uses Information Object originators and System Administrators (FIA_UID.2) for event accountability.

7.1.3 SF.CONFIGURATION

The TOE provides two management interfaces for configuration, local and remote. The Local System Administrator interface provides the full set of administrative tasks. The Remote System Administrator interface provides a reduced subset of management operations.

Local System Administrators are permitted to:

- Perform initial configuration
- Upload one or more configuration vectors containing definitions of the TOE information flow policy and other settings for the TSF,
- Delete configuration vectors,

Local and Remote System Administrators are permitted to:

- The TOE allows a System Administrator to set the active configuration vector.
- Inspect system logs and audit data, delete exported audit records

- Inspect state information
- Change TOE state to active or management mode
- Initiate warmstart or full restart
- Initiate secure erase (factory reset)
- Initiate emergency erase

The TOE restricts access to management functions according to command access attributes for each administrator.

This implements FDP_ACC.2, FDP_ACF.1, FIA_ATD.1, FIA_UID.2, FMT_MOF.1/RSP, FMT_MOF.1/LSP, FMT_MOF.1/AUD, FMT_MTD.1, FMT_MTD.3, FMT_REV.1, FMT_SMF.1, and FMT_SMR.2. The SF provides restrictive values for configuration vector attributes (FMT_MSA.1, FMT_MSA.3).

7.1.4 SF.DOMAIN_ISOLATION

The TOE mitigates limits the possibility of unintended disclosure of information between the domains by disassembling and reconstructing each information object being released by the Guard.

This function also performs mapping of addresses and identifiers present in the message. This SF implements the information flow control requirements FDP_IFC.2/r, FDP_IFF.1/r, and FDP_RIP.2 protects against unintentional information flow.

7.1.5 SF.EMERGENCY_ERASE

The TOE allows a system administrator to perform an emergency erase, leaving the TOE in an inoperable state. This SF implements a subset of FMT_SMF.1.

7.1.6 SF.FILTER

The TOE ensures that only attributes that do not violate the Information Flow Policy are allowed to be part of released Information Objects. This security function may be configured to remove attributes that violate the information flow policy, or ensure objects are rejected if required by the current configuration. The SF support implementation of the filtering requirements from FDP_IFC.2/r and FDP_IFF.1/r,

7.1.7 SF.INTERNAL_PARTITIONING

The TOE is composed of a number of modules, separated according to responsibilities and security implementation. This SF enforces limitations on the TOE internal information flow according to well-defined communication paths. This implements the information flow requirements in FDP_IFC.2/i and FDP_IFF.1/i.

The TOE also ensures internal communication resources are routinely cleared of sensitive information (FDP_RIP.2).

7.1.8 SF.MAC

The TOE implements MAC for information objects considered for release by the TOE.

The Mandatory Access Control (MAC) implements access rights according to hierarchical classification level (HCL), non-hierarchical categories (NHC) and security policy (SP). For MAC evaluation, Security Policy is handled in the same way as non-hierarchical categories. A subject must pass both MAC and DAC (SF.ABAC) to access an object. This implements FDP_IFF.2 and FDP_ITC.2 for internal (when the information is available) and external information flows (for Information Objects exchanged between security domains).

7.1.9 SF.OBJECT_INTEGRITY

The TOE supports S/MIME digital signatures and validates the integrity of information objects being handled by the Guard. By digitally signing information objects mediated by the Guard, the TOE provides integrity protection, non-repudiation of origin, and non-repudiation of receipt.

The TOE verifies the digital signatures of information objects when received. As information objects are rebuilt by the TOE, existing signatures will not be valid when the Information Object is released. Thus, the TOE digitally signs Information Objects before they are released.

The TOE can be configured to operate without use of digital signatures on one or both connected traffic interfaces, to accommodate connectivity towards legacy networks.

Digital signature validation is a mandatory part of the release decision. If a digital signature is present and fails to validate, the TOE will prevent release of the information object.

The signature validation and creation function with attached identity and integrity protection implements FCO_NRO.1, FDP_UIT.1 and FTP_TRP.1. The TOE implements hashing algorithms (FCS_COP.1), which is forwarded to a connected HSM (TOE Environment) as part of signature creation and validation.

7.1.10 SF.PRIORITY

The TOE uses priority attributes embedded in the information objects to prioritize traffic, and provides reserved resources for flash/override (high-priority) traffic to minimize delay of these information objects even during network congestion. Priority attributes are embedded in in information objects and internal IPC mechanisms. This SF implements FRU_PRS.1.

This SF only applies to the STANAG 4406 and SMTP Guards.

7.1.11 SF.SECURE_STATE

The TOE ensures that it is initialized to a secure state before entering operation.

The TOE maintains a secure state during operation, whenever transitioning between runtime states and on detected failures.

The TOE will transition to management mode or halt if required to preserve a secure state. The secure state and recovery SF implements FPT_FLS.1, FPT_RCV.4.

7.1.12 SF.SECURITY_LABELS

The TOE ensures that all information objects are associated with a security label. The information objects may optionally be associated with an additional original security label, set by SF.POLICY_MAPPING.

The information object's security label is decoded and verified for validity and consistency. SF.MAC ensures that the security label is within the required label range defined for the traffic interface.

The Guard supports security domains where a information objects exists without a trusted security label if the security domain is a system high domain. In this configuration, the TOE will assign information objects a fixed security label.

This implements security label mapping as described in FDP_ETC.2, FDP_ITC.2, FPT_TDC.1.

7.1.13 SF.POLICY_MAPPING

The TOE supports mapping of security labels from one security policy to another, according to defined rules, preserving the level of protection required for the specific marking. The function also preserves the original security label when the object security label is mapped. This security function provides the IT implementation of bilateral agreements on protection of shared classified information.

SF.SECURITY_LABELS provides security policy definitions that are used internally by the TOE. The security policy definitions are consistent across security policies, and when enabled, SF.SECURITY_LABEL_MAPPING transforms security labels from one security policy to another according to configuration. This implements security label mapping as described in FDP_ETC.2, FDP_ITC.2.

7.1.14 SF.SELF_TEST

The TOE performs self-tests on the integrity of TSF data and TSF executable code during startup and state changes. If a self-test fails, the TOE will perform a state change to management mode, or halt if required.

TSF integrity validation is implemented in part by the TOE and the platform (TOE Environment). Validation of the TSF integrity can be triggered through a restart by System Administrators. Validation of configuration vectors (TSF data) is automatically performed whenever the active configuration vector is changed. The SF implements FPT_TST.1.

8. RATIONALE

The rationale demonstrates that threats, assumptions and policies form a basis for the definition of security objectives. Likewise, it is demonstrated that the chosen security requirements cover all security objectives, and that security functions in the TOE or its environment fully cover the security requirements.

8.1 SECURITY OBJECTIVES RATIONALE

In the following subsections every security objective is correlated with identified threats and assumptions. It is furthermore shown that all identified threats are covered by a security objective.

The following three tables (Table 8-1, Table 8-2 and Table 8-3) demonstrate that all threats, assumption and policies are covered by a security objective. Some threats are fully covered by a single security objective, while others need more than one security objective to be fully covered.

Threats **Security objectives**	T.ADMIN_MASQUERADE	T.AUDIT_COMPROMISE	T.OBJECT_TAMPERING	T.COVERT_CHANNEL	T.DOS	T.INFORMATION_LEAK	T.INSECURE_STATE	T.MALWARE_INJECTION	T.METADATA_LEAK	T.NETWORK_ATTACK	T.RECONNAISSANCE	T.RESIDUAL_DATA	T.TSF_COMPROMISE	T.UNATTENDED_ADMIN_SESSION	T.UNAUTHORIZED_ACCESS	T.UNNOTICED_ATTACK
O.ACCESS												X				
O.AUDIT	X	X		X						X	X			X	X	X
O.CMD_ACL	X	X											X		X	X
O.CONFIGURATION_CHANGE							X									
O.CONTROLLED_INFORMATION_FLOW						X		X	X							
O.CORRECT_TSF_OPERATION							X									
O.INTERNAL_LEAST_PRIVILEGE		X		X		X				X	X					
O.LABEL_MAPPING									X							
O.MAC																
O.MGMT_MODE							X									
O.MINIMAL_PROXY				X				X	X		X					
O.OBJECT_INTEGRITY			X													
O.RESIDUAL_INFORMATION						X	X					X				
O.RESOURCE_SHARING					X											

Copyright © THALES Norway AS

Threats Security objectives	T.ADMIN_MASQUERADE	T.AUDIT_COMPROMISE	T.OBJECT_TAMPERING	T.COVERT_CHANNEL	T.DOS	T.INFORMATION_LEAK	T.INSECURE_STATE	T.MALWARE_INJECTION	T.METADATA_LEAK	T.NETWORK_ATTACK	T.RECONNAISSANCE	T.RESIDUAL_DATA	T.TSF_COMPROMISE	T.UNATTENDED_ADMIN_SESSION	T.UNAUTHORIZED_ACCESS	T.UNNOTICED_ATTACK
O.SECURE_STATE					X		X			X			X			
O.SUBJECT_ISOLATION						X	X			X	X	X	X			
O.TRANSITION							X									
O.TSF_INTEGRITY							X						X			X
O.VALID_LABEL			X													
OE.BORDER_PROTECTION					X					X	X		X		X	X
OE.CONTENT_INSPECTION_SERVICE				X				X	X	X						X
OE.DIRECTORY_SERVICE			X													
OE.APPROVED_CRYPTO			X													
OE.APPROVED_PKI			X													
OE.MINIMAL_POSTURE	X									X	X		X			
OE.NETWORK			X		X					X	X				X	X
OE.PHYSICAL_ACCESS_MANAGED													X	X	X	X
OE.CONFIGURATION			X	X		X										
OE.TRUSTED_AND_TRAINED_ADMIN		X				X							X	X		X
OE.TRUSTED_LABELLER						X										
OE.PLATFORM							X			X			X			X
OE.TIME_SOURCE		X														

Table 8-1: TOE threats coverage

Copyright © THALES Norway AS

Security objectives	A.APPROVED_CRYPTO	A.APPROVED_PKI	A.CORRECT_CONFIGURATION	A.NETWORK_PROTECTED	A.PHYSICAL_ACCESS_MANAGED	A.TRUSTED_AND_TRAINED_ADMIN	A.TRUSTED_LABELLER
OE.BORDER_PROTECTION			X				
OE.CONTENT_INSPECTION_SERVICE			X				
OE.DIRECTORY_SERVICE			X				
OE.APPROVED_CRYPTO	X						
OE.APPROVED_PKI		X					
OE.MINIMAL_POSTURE				X			
OE.NETWORK				X			
OE.PHYSICAL_ACCESS_MANAGED					X		
OE.CONFIGURATION			X				
OE.TRUSTED_AND_TRAINED_ADMIN			X			X	
OE.TRUSTED_LABELLER							X
OE.PLATFORM							
OE.TIME_SOURCE			X				

Table 8-2: Assumptions coverage

Security objectives	P.ACCOUNABILITY	P.CLASSIFICATION	P.CRYPTOGRAPHY	P.MINIMAL_POSTURE
O.ACCESS				X
O.AUDIT	X			
O.CONFIGURATION_CHANGE				
O.CORRECT_TSF_OPERATION				
O.INTERNAL_LEAST_PRIVILEG				X

Copyright © THALES Norway AS

Security objectives	P.ACCOUNTABILITY	P.CLASSIFICATION	P.CRYPTOGRAPHY	P.MINIMAL_POSTURE
E				
O.LABEL_MAPPING		X		
O.MAC		X		
O.MGMT_MODE				
O.CONTROLLED_INFORMATION_FLOW				
O.MINIMAL_PROXY				
O.OBJECT_INTEGRITY	X		X	
O.RESIDUAL_INFORMATION				
O.RESOURCE_SHARING				
O.SECURE_STATE				
O.SUBJECT_ISOLATION				
O.TRANSITION				
O.TSF_INTEGRITY				
O.VALID_LABEL		X		
OE.BORDER_PROTECTION				X
OE.CONTENT_INSPECTION_SERVICE				
OE.DIRECTORY_SERVICE				
OE.APPROVED_CRYPTO			X	
OE.APPROVED_PKI			X	
OE.MINIMAL_POSTURE				X
OE.PHYSICAL_ACCESS_MANAGED				
OE.CONFIGURATION				
OE.TRUSTED_AND_TRAINED_ADMIN	X			
OE.TRUSTED_LABELLER		X		
OE.PLATFORM				X
OE.TIME_SOURCE				

Table 8-3: Policies coverage

Copyright © THALES Norway AS

8.1.1 THREATS MET BY OBJECTIVES RATIONALE

The following rationale describes how each threat is met by the TOE or TOE Environment.

T.ADMIN_MASQUERADE

The TOE Environment ensures that only authorized administrators may access the administrative interface (OE.MINIMAL_POSTURE). In addition, the TOE limits the operations available to a remote administrator (O.CMD_ACL), and records an audit trail for administrative operations (O.AUDIT).

T.AUDIT_COMPROMISE

The TOE ensures the TOE audit trail is available to authorized administrators only by protecting the audit trail (O.AUDIT, O.CMD_ACL). The TOE is compartmentalized to prevent attackers from interfering with audit generation or storage (O.INTERNAL_LEAST_PRIVILEGE). The TOE Environment ensures administrators are trained to correctly configure and monitor the system (OE.TRUSTED_AND_TRAINED_ADMIN), and supply the TOE with a reliable time source (OE.TIME_SOURCE).

T.OBJECT_TAMPERING

The TOE prevents object tampering attacks through use of digital signatures (O.OBJECT_INTEGRITY), when configured (OE.CONFIGURATION), supported by directory and PKI services provided by the TOE Environment (OE.DIRECTORY_SERVICE, OE.APPROVED_CRYPTO, OE.APPROVED_PKI). Object Labels are validated (O.VALID_LABEL).

When the TOE is connected to security domains that do not provide integrity protection of information objects, it is the responsibility of the TOE Environment to offer the appropriate protection (OE.NETWORK).

T.COVERT_CHANNEL

The exploitation of residual covert channels are mitigated by decomposing and rebuilding all information objects, and removing or mapping attributes according to the defined security policy (O.INTERNAL_LEAST_PRIVILEGE, O.MINIMAL_PROXY). The TOE generates audit logs that may be analysed (O.AUDIT) by the TOE Environment. The TOE Environment is responsible for providing content inspection services (OE.CONTENT_INSPECTION_SERVICE), correctly configuring the TOE to minimize covert channels (OE.CONFIGURATION),

T.DOS

The TOE provides reserved resources for high priority traffic (O.RESOURCE_SHARING) and monitors health attributes (O.SECURE_STATE) to limit the effects of denial of service attacks or during high traffic scenarios. Border protection devices (OE.BORDER_PROTECTION) provided by the TOE Environment and controlled network access (OE.NETWORK) are the primary defenses against DOS.

T.INFORMATION_LEAK

Information leaks are mitigated through a compartmentalized TOE implementation (O.CONTROLLED_INFORMATION_FLOW, O.INTERNAL_LEAST_PRIVILEGE, O.RESIDUAL_INFORMATION, O.SUBJECT_ISOLATION). Further, the TOE Environment ensures the configuration is correct (OE.CONFIGURATION, OE.TRUSTED_AND_TRAINED_ADMINS), and information objects are correctly labelled (OE.TRUSTED_LABELLER).

T.INSECURE_STATE

The TOE provides mechanisms to prevent entering an insecure state (O.SECURE_STATE). The TOE ensures a consistent configuration by performing a static configuration change where reconfiguration is

Copyright © THALES Norway AS

performed in a management mode (O.CONFIGURATION_CHANGE, O.MGMT_MODE, O.TRANSITION). This guarantees that configuration cannot be changed while an information object is being processed, and ensures a consistent configuration state across all TOE modules. During a configuration change, the TSF validates the configuration vector. TSF integrity is verified during boot (O.TSF_INTEGRITY, OE.PLATFORM) and during operation (O.CORRECT_TSF_OPERATION).

The TOE prevents interference between different parts through decomposition and controlled information flows (O.SUBJECT_ISOLATION), routine clearing of residual data (O.RESIDUAL_INFORMATION) and separation is further supported by the operating system and hardware (OE.PLATFORM).

T.MALWARE_INJECTION

The TOE prevents malware from being relayed by acting as a minimal proxy, limiting the attributes to be transferred according to configuration (O.MINIMAL_PROXY), relaying allowed attributes and objects only (O.CONTROLLED_INFORMATION_FLOW) and ensuring information objects are analysed in a content inspection service (OE_CONTENT_INSPECTION_SERVICE).

T.METADATA_LEAK

The TOE ensures information object attributes are mapped, stripped or rebuilt according to the active configuration vector (O.CONTROLLED_INFORMATION_FLOW, O.LABEL_MAPPING, O.MINIMAL_PROXY). The TOE Environment may also inspect Information Objects (OE.CONTENT_INSPECTION_SERVICE).

T.NETWORK_ATTACK

The TOE Environment mitigates networked attacks (OE.BORDER_PROTECTION, OE.NETWORK, OE.MINIMAL_POSTURE), provides a content inspection service (OE.CONTENT_INSPECTION_SERVICE) and a high assurance separation kernel platform (OE.PLATFORM). The TOE minimizes feasible attack vectors and the ability for an attacker to exploit residual vulnerabilities (O.INTERNAL_LEAST_PRIVILEGE, O.SUBJECT_ISOLATION, O.SECURE_STATE), and records security related events (O.AUDIT).

T.RECONNAISSANCE

Reconnaissance operations are audited if detected (O.AUDIT). The TOE provides a compartmentalized design that limits the ability to perform reconnaissance (O.INTERNAL_LEAST_PRIVILEGE, O.SUBJECT_ISOLATION), and sanitizes Information Objects to limit information from attributes and trace information (O.MINIMAL_PROXY).

The TOE Environment mitigates reconnaissance attempts through protecting the network environment and the deployed TOE. (OE.BORDER_PROTECTION, OE.NETWORK, OE.MINIMAL_POSTURE)

T.RESIDUAL_DATA

The TOE prevents information leaks between Information Objects by isolating internal components and resetting resource between processing each object (O.RESIDUAL_INFORMATION, O.SUBJECT_ISOLATION). This ensures faults during processing of one Information Object will not be propagated to subsequent Information Objects. The TOE limits information sent to and from external supporting services (O.ACCESS).

T.TSF_COMPROMISE

The TOE and TOE Environment validates the integrity of ASSET.TSF_INTERNALS during startup (O.TSF_INTEGRITY). The TOE architecture and security policy mitigates an attacker’s ability to exploit residual vulnerabilities to bypass the TOE Security Policy (O.SUBJECT_ISOLATION, OE.MINIMAL_POSTURE, OE.PLATFORM). Restrictive access to management functions reduces the risk of unintentional administrator errors (O.CMD_ACL) The TOE provides mechanisms to ensure a secure state (O.SECURE_STATE).

The TOE Environment mitigates physical and network attack vectors (OE.PHYSICAL_ACCESS_MANAGED, OE.BORDER_PROTECTION) and ensures administrators follow guidance documentation (OE.TRUSTED_AND_TRAINED_ADMIN).

T.UNATTENDED_ADMIN_SESSION

The TOE Environment ensures administrator sessions are not left unattended (OE.PHYSICAL_ACCESS_MANAGED, OE.TRUSTED_AND_TRAINED_ADMIN). Administrative operations are audited by the TOE (O.AUDIT).

T.UNAUTHORIZED_ACCESS

The TOE Environment mitigates the ability for an attacker to gain unauthorized access to the TOE through physical and network protection appropriate for the information in the connected security domains (OE.BORDER_PROTECTION, OE.NETWORK, OE.PHYSICAL_ACCESS_MANAGED). Administrative operations are controlled (O.CMD_ACL) and audited by the TOE (O.AUDIT). Mandatory Access Control is enforced by the TSF to limit internal and external information flow.

T.UNNOTICED_ATTACK

The TOE and TOE Environment (O.TSF_INTEGRITY, OE.PLATFORM) verifies the integrity of the TOE executable code and configuration vectors.

The TOE Environment mitigates the risk of an undiscovered attack by ensuring proper configuration of the TOE (O.AUDIT, O.CMD_ACL, OE.TRUSTED_AND_TRAINED_ADMIN), inspection of information objects (OE.CONTENT_INSPECTION_SERVICE), and protecting networks and physical access (OE.BORDER_PROTECTION, OE.NETWORK, OE.PHYSICAL_ACCESS_MANAGED).

8.1.2 ASSUMPTIONS MET BY OBJECTIVES FOR THE ENVIRONMENT RATIONALE

A.APPROVED_PKI

OE.APPROVED_PKI directly upholds the assumption.

A.APPROVED_CRYPTO

OE.APPROVED_CRYPTO directly upholds the assumption.

A.CORRECT_CONFIGURATION

The assumption is upheld by OE.TRUSTED_AND_TRAINED_ADMIN and OE.CONFIGURATION, supported by external services (OE.BORDER_PROTECTION, OE.CONTENT_INSPECTION_SERVICE, OE.DIRECTORY_SERVICE, OE.TIME_SOURCE).

A.NETWORK_PROTECTED

OE.NETWORK and OE.MINIMAL_POSTURE directly upholds the assumption.

A.PHYSICAL_ACCESS_MANAGED

OE.PHYSICAL_ACCESS_MANAGED directly upholds the assumption.

A.TRUSTED_AND_TRAINED_ADMIN

OE.TRUSTED_AND_TRAINED_ADMIN directly upholds the assumption.

A.TRUSTED_LABELLER

OE.TRUSTED_LABELLER directly upholds the assumption.

8.1.3 POLICIES

P.ACCOUNTABILITY

Accountability is implemented by the TOE through an audit trail of all released objects (O.AUDIT), which is monitored or kept for later inspection by administrators (OE.TRUSTED_AND_TRAINED_ADMIN). For security domains using digital signatures, the TOE is able to verify the source address for non-repudiation (O.OBJECT_INTEGRITY).

P.CLASSIFICATION

The classification of information is done by the originator (O.MAC, OE.TRUSTED_LABELLER) or added by the TOE in system high configurations (O.VALID_LABEL). Classifications are mapped according to the configured policy (O.LABEL_MAPPING).

P.CRYPTOGRAPHY

The TOE uses approved cryptographic algorithms and implementation (O.OBJECT_INTEGRITY, OE.APPROVED_CRYPTO, OE.APPROVED_PKI).

P.MINIMAL_POSTURE

The TOE and TOE Environment provides the minimum external interfaces required to implement the Guard functionality (O.ACCESS, O.INTERNAL_LEAST_PRIVILEGE, OE.BORDER_PROTECTION, OE.MINIMAL_POSTURE, OE.PLATFORM)

8.2 SECURITY REQUIREMENTS RATIONALE

8.2.1 RATIONALE FOR SECURITY FUNCTIONAL REQUIREMENTS TO COVER OBJECTIVES

The following tables (Table 8-4 and Table 8-5) show that requirements are appropriate to cover TOE security objectives.

Copyright © THALES Norway AS

Objectives Req.	O.ACCESS	O.AUDIT	O.CMD_ACL	O.CONFIGURATION_CHANGE	O.CORRECT_TSF_OPERATION	O.INTERNAL_LEAST_PRIVILEGE	O.LABEL_MAPPING	O.MAC	O.MGMT_MODE	O.CONTROLLED_INFORMATION_FLOW	O.MINIMAL_PROXY
FAU_ARP.1		X									
FAU_GEN.1		X									
FAU_GEN.2		X									
FAU_SAA.1		X									
FAU_SAR.1		X									
FAU_SAR.2		X									
FAU_STG.1		X									
FAU_STG.3		X									
FAU_STG.4		X									
FCO_NRO.1											
FCS_COP.1											
FDP_ACC.2			X								
FDP_ACF.1			X								
FDP_ETC.2							X	X			
FDP_IFC.2/r										X	X
FDP_IFC.2/i	X					X				X	
FDP_IFF.1/r										X	X
FDP_IFF.1/i	X					X				X	
FDP_IFF.2								X		X	
FDP_ITC.2							X	X			
FDP_RIP.2											
FDP_UIT.1											
FIA_ATD.1			X								
FIA_UID.2		X	X								
FMT_MOF.1/RSP				X							
FMT_MOF.1/L				X							

Copyright © THALES Norway AS

Objectives Req.	O.ACCESS	O.AUDIT	O.CMD_ACL	O.CONFIGURATION_CHANGE	O.CORRECT_TSF_OPERATION	O.INTERNAL_LEAST_PRIVILEGE	O.LABEL_MAPPING	O.MAC	O.MGMT_MODE	O.CONTROLLED_INFORMATION_FLOW	O.MINIMAL_PROXY
SP											
FMT_MOF.1/AUD				X							
FMT_MSA.1				X							
FMT_MSA.3				X							
FMT_MTD.1				X					X		
FMT_MTD.3				X							
FMT_REV.1				X							
FMT_SMF.1				X							
FMT_SMR.2				X							
FPT_FLS.1					X						
FPT_RCV.4					X						
FPT_TDC.1											
FPT_TST.1											
FRU_PRS.1											
FTP_TRP.1											

Table 8-4: Security objectives satisfaction (part 1/2)

Objective Req.	O.OBJECT_INTEGRITY	O.RESIDUAL_INFORMATION	O.RESOURCE_SHARING	O.SECURE_STATE	O.SUBJECT_ISOLATION	O.TRANSITION	O.TSF_INTEGRITY	O.VALID_LABEL
FAU_ARP.1								
FAU_GEN.1								

Copyright © THALES Norway AS

Objective Req.	O.OBJECT_INTEGRITY	O.RESIDUAL_INFORMATION	O.RESOURCE_SHARING	O.SECURE_STATE	O.SUBJECT_ISOLATION	O.TRANSITION	O.TSF_INTEGRITY	O.VALID_LABEL
FAU_GEN.2								
FAU_SAA.1								
FAU_SAR.1								
FAU_SAR.2								
FAU_STG.1								
FAU_STG.3								
FAU_STG.4								
FCO_NRO.1	X							
FCS_COP.1	X							
FDP_ACC.2								
FDP_ACF.1								
FDP_ETC.2								
FDP_IFC.2/r								
FDP_IFC.2/i					X			
FDP_IFF.1/r								
FDP_IFF.1/i					X			
FDP_IFF.2								
FDP_ITC.2								X
FDP_RIP.2		X			X			
FDP_UIT.1	X							
FIA_ATD.1								
FIA_UID.2								
FMT_MOF.1/INV								
FMT_MOF.1/AUD								
FMT_MOF.1/SP								
FMT_MSA.1								
FMT_MSA.3								

Copyright © THALES Norway AS

Objective Req.	O.OBJECT_INTEGRITY	O.RESIDUAL_INFORMATI ON	O.RESOURCE_SHARING	O.SECURE_STATE	O.SUBJECT_ISOLATION	O.TRANSITION	O.TSF_INTEGRITY	O.VALID_LABEL
FMT_MTD.1						X		
FMT_MTD.3								
FMT_SMF.1								
FMT_SMR.2								
FPT_FLS.1				X				
FPT_RCV.4				X				
FPT_TDC.1								X
FPT_TST.1							X	
FRU_PRS.1			X					
FTP_TRP.1	X							

Table 8-5 Security objectives satisfaction (part 2/2)

O.ACCESS

The Internal Flow Control Policy limits information flow to and from the supporting services (FDP_IFC.2/i, FDP_IFF.1/i).

O.AUDIT

The objective is implemented by requirements for generating (FAU_ARP.1, FAU_GEN.1, FAU_GEN.2), inspecting (FAU_SAA.1, FAU_SAR.1, FAU_SAR.2) and storing (FAU_STG.1, FAU_STG.3, FAU_STG.4) audit data. User identification for System Administrators and Information Object originators is ensured by FIA_UID.2.

O.CMD_ACL

The objective requires restrictive access to management functions, which is implemented in full by FDP_ACC.2, FDP_ACF.1, FIA_ATD.1, FIA_UID.2.

O.CONFIGURATION_CHANGE

The objective is implemented by the series of FMT_* requirements, defining the required administrator roles, and TSF management operations (FMT_MOF.1/RSP, FMT_MOF.1/LSP, FMT_MOF.1/AUD, FMT_MSA.1, FMT_MSA.3, FMT_MTD.1, FMT_MTD.3, FMT_REV.1, FMT_SMF.1, FMT_SMR.2).

O.CORRECT_TSF_OPERATION

The TOE provides secure fault and recovery mechanisms (FPT_FLS.1, FPT_RCV.4).

O.INTERNAL_LEAST_PRIVILEGE

Copyright © THALES Norway AS

The TSF strictly limits the information flow between components (FDP_IFC.2/i, FDP_IFF.1/i).

O.LABEL_MAPPING

The TOE preserves security labels embedded with Information Objects (FDP_ETC.2, FTP_TRP.2).

O.MAC

The TOE preserves security labels embedded with Information Objects (FDP_ETC.2, FTP_TRP.2) and implements Mandatory Access Control according to the Bell & LaPadula security policy model (FDP_IFF.2).

O.MGMT_MODE

The objective is implemented through management functions for entering management mode and managing configuration (FMT_MTD.1).

O.CONTROLLED_INFORMATION_FLOW

The objective is implemented through rigorous information flow control mechanisms internally and externally (FDP_IFC.2/r, FDP_IFC.2/i, FDP_IFF.1/r, FDP_IFF.1/i, FDP_IFF.2) on all information flows in and out of the TOE.

O.MINIMAL_PROXY

The objective is implemented by the Object Release SFP, which minimizes covert channels and relayed attributes for information objects relayed by the TOE (FDP_IFC.2/r, FDP_IFF.1/r).

O.OBJECT_INTEGRITY

The TOE implements integrity protection for Information Objects ((FCO_NRO.1, FDP_UIT.1, FTP_TRP.1), using signature extraction and creation and hashing of the Information Objects using approved cryptographic algorithms (FCS_COP.1). The TOE Environment is relied upon to perform cryptographic operations such as signature validation, signing and key management.

O.RESIDUAL_INFORMATION

The TOE clears residual information from information objects and internal data structures (FDP_RIP.2).

O.RESOURCE_SHARING

The objective is met through priority of service requirement (FRU_PRS.1).

O.SECURE_STATE

The objective is met by requirements for secure state recovery and secure failure handling (FPT_FLS.1, FPT_RCV.4).

O.SUBJECT_ISOLATION

The objective is met by implementing the Internal Information Flow Control SFP, and clearing of residual information (FDP_IFC.2/i, FDP_IFC.2/i, FDP_IFF.1/i, FDP_RIP.2).

O.TRANSITION

Copyright © THALES Norway AS

The objective is met by requirements for providing administrators with the ability to transition the TOE to different run states, shut down or restart (FMT_MTD.1).

O.TSF_INTEGRITY

The TSF validates internals and configuration (FPT_TST.1)

O.VALID_LABEL

The Object Release SFP ensures security labels present in Information Objects are valid (FDP_ITC.2, FPT_TDC.1).

8.2.2 FUNCTIONAL SECURITY REQUIREMENTS DEPENDENCIES

The table shows each component's direct dependencies to other components. This demonstrates that the set of security requirements form a mutually supportive and consistent whole.

TOE Requirement	Dependency	Included	Rationale
FAU_ARP.1	FAU_SAA.1	Yes	
FAU_GEN.1	FPT_STM.1	No	Timestamps are provided by the TOE Environment through OE.TIME_SOURCE
FAU_GEN.2	FAU_GEN.1 FIA_UID.1	Yes Yes	 Higher-level SFR: FIA_UID.2
FAU_SAA.1	FAU_GEN.1	Yes	
FAU_SAR.1	FAU_GEN.1	Yes	
FAU_SAR.2	FAU_SAR.1	Yes	
FAU_STG.1	FAU_GEN.1	Yes	
FAU_STG.3	FAU_STG.1	Yes	
FAU_STG.4	FAU_STG.1	Yes	
FCO_NRO.1	FIA_UID.1	Yes	Higher-level SFR: FIA_UID.2 Originator identification is part of each specific Information Object protocol.
FCS_COP.1	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 FCS_CKM.4	Yes No	 Key destruction is not relevant for the TOE, as FCS_COP.1 covers hashing only. Keys and signing operations are handled by the TOE Environment (OE.APPROVED_CRYPTO).
FDP_ACC.2	FDP_ACF.1	Yes	

Copyright © THALES Norway AS

TOE Requirement	Dependency	Included	Rationale
FDP_ACF.1	FDP_ACC.1 FMT_MSA.3	Yes Yes	Higher-level SFR: FDP_ACC.2
FDP_ETC.2	FDP_ACC.1 or FDP_IFC.1	Yes	Higher-level SFR: FDP_IFC.2
FDP_IFC.2	FDP_IFF.1	Yes	
FDP_IFF.1	FDP_IFC.1 FMT_MSA.3	Yes Yes	Higher-level SFR: FDP_IFC.2
FDP_IFF.2	FDP_IFC.1 FMT_MSA.3		Higher-level SFR: FDP_IFC.2
FDP_ITC.2	FDP_ACC.1 or FDP_IFC.1 FTP_ITC.1 or FTP_TRP.1 FPT_TDC.1	Yes Yes Yes	Higher-level SFR: FDP_IFC.2
FDP_RIP.2	None		
FDP_UIT.1	FDP_ACC.1 or FDP_IFC.1 FTP_ITC.1 or FTP_TRP.1	Yes Yes	Higher-level SFR: FDP_IFC.2
FIA_ATD.1	None.		
FIA_UID.2	None		
FMT_MOF.1	FMT_SMF.1 FMT_SMR.1	Yes Yes	Higher-level SFR: FMT_SMR.2
FMT_MSA.1	FDP_ACC.1 or FDP_IFC.1 FMT_SMF.1 FMT_SMR.1	Yes Yes Yes	Higher-level SFR: FDP_IFC.2 Higher-level SFR: FMT_SMR.2
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	Yes Yes	Higher-level SFR: FMT_SMR.2
FMT_MTD.1	FMT_SMF.1 FMT_SMR.1	Yes Yes	Higher-level SFR: FMT_SMR.2
FMT_MTD.3	FMT_MTD.1	Yes	
FMT_REV.1	FMT_SMR.1	Yes	Higher-level SFR: FMT_SMR.2
FMT_SMF.1	None		
FMT_SMR.2	FIA_UID.1	Yes	Higher-level SFR: FIA_UID.2
FPT_FLS.1	None		

Copyright © THALES Norway AS

TOE Requirement	Dependency	Included	Rationale
FPT_RCV.4	None		
FPT_TDC.1	None		
FPT_TST.1	None		
FRU_PRS.1	None		
FTP_TRP.1	None		

Table 8-6: Functional requirements dependency check

8.2.3 TOE SECURITY ASSURANCE REQUIREMENTS RATIONALE

The TOE meets the assurance requirements for EAL4 augmented by ALC_FLR.3 and AVA_VAN.4.

The TOE stresses assurance from best practice development practices. Through review of vendor-supplied evidence and independent testing the Assurance Requirements confirm the implementation of these practices.

The selected assurance level ensures the TOE fulfills national requirements for use in military and governmental networks, handling and separating information as specified in the TOE Overview and TOE Description. In particular, mediation of information flow between national, international and inter-organizational security domains operating at similar or different sensitivity levels.

8.3 TOE SUMMARY SPECIFICATION RATIONALE

8.3.1 TOE SECURITY FUNCTIONAL REQUIREMENTS SATISFACTION

This chapter demonstrates that the TOE Security Functions completely implement the TOE Security Functional Requirements.

Table 8-7 shows that each Security Functional Requirement is covered by at least one TOE Security Function and vice versa.

The table is supported by a rationale demonstrating that each SFR is completely implemented by one or more TSFs, described along with each TSF in Ch. 7.1.

SF Req.	SF.ABAC	SF.AUDIT	SF.CONFIGURATION	SF.DOMAIN_ISOLATION	SF.EMERGENCY_ERASE	SF.FILTER	SF.INTERNAL._PARTITIONING	SF.MAC	SF.OBJECT_INTEGRITY	SF.PRIORITY	SF.SECURE_STATE	SF.SECURITY_LABELS	SF.POLICY_MAPPING	SF.SELF_TEST
FAU_ARP.1		X												
FAU_GEN.1		X												
FAU_GEN.2		X												
FAU_SAA.1		X												
FAU_SAR.1		X												
FAU_SAR.2		X												
FAU_STG.1		X												
FAU_STG.3		X												
FAU_STG.4		X												
FCO_NRO.1									X					
FCS_COP.1									X					
FDP_ACC.2			X											
FDP_ACF.1			X											
FDP_ETC.2												X	X	
FDP_IFC.2/r	X			X		X								
FDP_IFC.2/i							X							
FDP_IFF.1/r	X			X		X								
FDP_IFF.1/i							X							
FDP_IFF.2	X							X						
FDP_ITC.2								X				X	X	
FDP_RIP.2				X			X							
FDP_UIT.1									X					
FIA_ATD.1			X											
FIA_UID.2		X	X											
FMT_MOF.1/RSP			X											
FMT_MOF.1/LSP			X											
FMT_MOF.1/AU			X											

SF Req.	SF.ABAC	SF.AUDIT	SF.CONFIGURATION	SF.DOMAIN_ISOLATION	SF.EMERGENCY_ERASE	SF.FILTER	SF.INTERNAL_PARTITIONING	SF.MAC	SF.OBJECT_INTEGRITY	SF.PRIORITY	SF.SECURE_STATE	SF.SECURITY_LABELS	SF.POLICY_MAPPING	SF.SELF_TEST
D														
FMT_MSA.1			X											
FMT_MSA.3			X											
FMT_MTD.1			X											
FMT_MTD.3			X											
FMT_REV.1			X											
FMT_SMF.1			X		X									
FMT_SMR.2			X											
FPT_FLS.1											X			
FPT_RCV.4											X			
FPT_TDC.1												X		
FPT_TST.1														X
FRU_PRS.1										X				
FTP_TRP.1									X					

Table 8-7: Functional requirements satisfaction

8.4 PP RATIONALE

Not applicable

9. NOTES

9.1 NOTATION

The following notation is used for detailing Security Functional Requirements:

- **Bold text** is used for minor changes to the standard requirement text, to improve language or readability.
- *Italic text* is used to show where assignments or selections have been made by the developer.
- ~~Strikethrough~~ is used to show where requirement text or irrelevant assignment text has been removed from requirements.

Iteration of security requirements is done by adding an abbreviation to the requirement. The title of each related chapter will contain a short description or reference. Example:

FMT_MTS.1/ADM Management of TSF Data (System Administrators)

FMT_MTD.1/SYS Management of TSF Data (System partition API)

9.2 ABBREVIATION AND ACRONYMS

Acronym	Extended
BPD	Border Protection Device
AHCI	Advanced Host Controller Interface
COTS	Commercial Off The Shelf
CPR	Content-based Protection and Release
CRL	Certificate Revocation List
HSM	Hardware Security Module Cryptographic module with a standardized interface to cryptographic functions. Typically used to provide a trusted implementation of cryptographic algorithms and secure storage of tokens or keys.
DAC	Discretionary Access Control
FSM	Finite State Machine, an implementation technique
HSM	Hardware Security Module Cryptographic module with a standardized interface to cryptographic functions. Typically used to provide a trusted implementation of cryptographic algorithms and secure storage of tokens or keys.
ICAP	Internet Content Adaptation Protocol Protocol aimed at providing simple object-based content vectoring for HTTP services. The protocol is well suited to offload a body of data for external processing, and is extensively used by antivirus vendors. The Guard uses this standardized protocol to offload information object data for external processing.
IPC	Inter-Process Communication See IPC object below.
LDAP	Lightweight Directory Access Protocol
MAC	Mandatory Access Control

MILS	Multiple Independent Levels of Security An architecture using decomposition and compartmentalization to implement evaluatable secure software systems.
MMHS	Military Message Handling System Message handling according to military requirements, including priority handling and multi-level security requirements, in this document related to STANAG 4066 [ST4406].
MTA	Mail Transfer Agent
NTP	Network Time Protocol, RFC5905
NVRAM	Non-volatile RAM Stores the BIOS executable and configuration data
PKCS	Public Key Cryptography Standards
PKE	Public Key Enablement
PKI	Public-Key Infrastructure
S/MIME	Secure/Multipurpose Internet Mail Extensions (S/MIME)
SBC	Single Board Computer, also denoted processing unit
SOAP	Simple Object Access Protocol [SOAP] Lightweight protocol intended for exchanging structure information in a decentralized, distributed environment, using a messaging framework, implemented in XML.
SSH	Secure SHell. An encrypted network protocol used for accessing the TOE management interface.
VPX	VPX, a standardized communication bus commonly used in industrial or military applications.
XMPP	Extensible Messaging and Presence Protocol [RFC6120] An extensible protocol for implementing services related to instant messaging and presence. Also known as Jabber.

Table 9-1 Acronyms

9.3 TERMINOLOGY

Administrator	Personnel responsible for the maintenance of the TOE and connected supporting services such as HSMs and directory.
Advanced Host Controller Interface	A technical standard defined by Intel that specifies the operation of disk controllers using a serial interface.
Border Protection Device	Network nodes protecting the TOE from network based attacks and reconnaissance. A typical configuration involves a content inspecting firewall with intrusion detection, denial of service mitigation, and optionally antivirus and malware detection. The Border Protection Devices are selected by the TOE Environment according to the level of assurance required for the connected networks.
Attribute Based Access Control (ABAC)	The attribute based access control implemented by the TOE, which provides automated release of information objects based on their associated security attributes, such as originator, recipient, content types, etc. Mandatory Access Control (MAC) according to the Bell LaPadula model is also part of the release policy.
COMM module	The COMM module is a module added to PikeOS to allow efficient and secure

Copyright © THALES Norway AS

	transfer of large data amounts between Partitions. The data is passed by reference, and the COMM module allows controlled access to the data. COMM channels provide access from Partitions to the COMM module storage.
COMM channel	See COMM module.
Information Object	An object being considered for release by the Guard. The supported Information Objects are STANAG 4406 messages, SMTP messages, XML/SOAP objects and XMPP/Chat objects.
IPC object	An information object used for TOE-internal communication between the TOE partitions, or between the TOE and software processes on the connected processing units A and B. IPC objects are subject to control by the reference monitor, ensuring that only approved communication channels between TOE Partitions are allowed. The TSF implements its own mechanisms, in addition to relying on services provided by PikeOS.
Native personality	PikeOS partition type with minimal footprint.
Partition	The TOE is separated into a number of partitions, managed by an embedded separation kernel [PIKEOS-ST]. Each partition's resources, communication channels and available CPU time is managed completely by the separation kernel, ensuring that individual partitions are protected from interference from other partitions.
Partition ID	Uniquely identifies a PikeOS partition (across all processing units).
Process	Generic term for the runtime representation of TOE executable code being executed inside a TOE partition.
Process ID	Unique identifier for a specific process within the TOE (across all processing units).
S/MIME SignedAttributes	Attributes associated with an Information Object that are covered by a S/MIME digital signature. Such an attribute may be a Security Label.
System Administrator	Users accessing the TOE management functions through the Local or Remote Management Interfaces.

Table 9-2 Terminology