Färist 4 - Security Target

Document version: 2.0

Date: 2015-06-08

Title	**Färist 4 - Security Target**		
Version:	2.0	Developer:	Tutus Data AB
Status:	Released	Classification:	Public

Tutus Data AB, Svärdvägen 11, SE-182 33 Danderyd, Tel: +46 (0)8 551 102 30 Fax: +46 (0)8 551 102 39, E-mail: info@tutus.se, Web site: http://www.tutus.se

Document history

Version	Date	Author	Changes to previous version
2.0	2015-06-08	Per Holmer	Initial released version

Table of contents

Index of tables

Index of illustrations

1 Introduction

1.1 Security Target Identification and organisation

Title:	Färist 4 – Security Target
Version:	2.0
Status:	Released
Date:	2015-06-08
Sponsor:	Tutus Data AB
Developer:	Tutus Data AB
Keywords:	Security Target, Common Criteria, Tutus, Virtual Private Network, VPN, Firewall, Networking

This ST has been structured in accordance with [CC] Part 1. The main sections of the ST are the introduction, security problem definition, security objectives, security requirements, TOE summary description and annexes.

The introduction provides general information about the TOE, serves as an aid to understand the nature of the TOE and its security functionality, and provides context for the ST's evaluation.

The security problem definition describes the security aspects of the environment in which the TOE is to be used and the manner in which it is to be employed. The TOE security environment includes:

a) assumptions regarding the TOE's intended usage and environment of use

b) threats relevant to secure TOE operation

c) organisational security policies with which the TOE must comply

The security objectives reflect the stated intent of the ST. They pertain to how the TOE will counter identified threats and how it will cover identified organisational security policies and assumptions. Each security objective is categorised as being for the TOE or for the environment. The security objectives rationale demonstrates that the stated security objectives are traceable to all of the aspects identified in the TOE security problem definition and that they are suitable to cover them.

The security requirements section provides detailed requirements, in separate subsections, for the TOE and its environment. The security requirements are further divided into the TOE security functional requirements and the TOE security assurance requirements.

The TOE summary specification addresses the security functions that are represented by the TOE to answer the security requirements.

The annex contains a list of abbreviations and a glossary relevant for this ST.

1.2 TOE Identification

The TOE is the firmware part of Färist and Färist Micro version 4.0.1.

1.3 TOE Type

The TOE type is a networking device. In particular, the TOE covered by this ST is a VPN (virtual private network) software part of a network appliance that is extendable

with Firewall functionality, allowing additional application filtering functionality to be added to the VPN channel. This security functionality of the ST is limited to only the VPN functionality and does not cover any filtering functionality.

1.4 TOE Overview

The TOE is available in two different version, Färist and Färist Micro both providing similar functionality and security. The Färist supports both MACSec and IPSec, while the Färist Micro only supports IPSec. To the user they differ mostly in performance, hardware configuration and physical size. They also share most of the components. This ST covers both versions.

The general functionality provided by the TOE is to securely interconnect two or more networks over a carrier network. The connection can be made on either OSI level 2 or level 3, i.e. the TOE will work as a bridge or router.

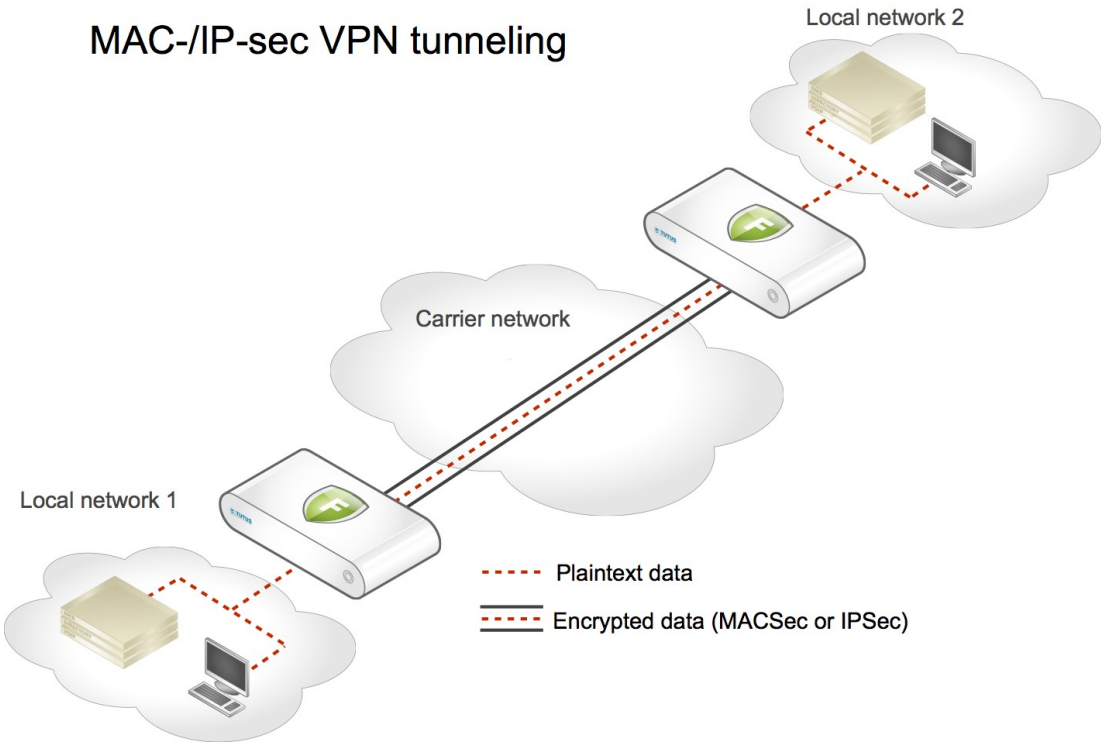

Illustration 1: General functionality of the Färist

This includes the following security functions:

- VPN-functionality: The TOE can negotiate SKUT + VPN, layer 2 MACSec for Färist and layer 3 IPSec (ESP tunnel mode).
- Audit: The main purpose of the log is to maintain traceability and to log events like startup, reconfiguration and similar. But it also logs security related events like attempts to connect using wrong/old credentials, integrity error on packets, internal integrity control and similar. All log events are sent to a configured log server. The Färist supports different log levels (severities) of auditing. In the evaluated configuration the log level "info" has to be used.

- Administration capabilities: An HTTPS API is offered to administrators for all relevant configuration of security functionality. There is also a front panel for displaying status information and importing a certificate. The HTTPS API uses a certificate based authentication. When using the HTTPS API there are two roles, one administrator with full privileges, and one that can only read information (operator). The only thing that can be written is the configuration. Things that can be read includes configuration, run time status, statistics and the locally saved log.
- Autoupdate: The TOE can be configured to automatically check for new firmware versions on a specified server and update itself. The integrity and authenticity of the updates will be verified. Only updates to newer (higher) versions can be done.
- Selftest and failsafe: The TOE also has selftest and failsafe capabilities.

In addition, all traffic is directed through a filter that can further limit the traffic over the VPN-channel. Included in the evaluation is the filter allowing all traffic to pass. This means that the filter does not provide any security functionality, but provides a proof of concept and interfaces that allow this empty filter to be replaced with other filters with actual filtering functions. The filter works at the level of the tunnel, i.e. it filters Ethernet packets in a layer 2 tunnel and IP-packets in a layer 3 tunnel.

Note that all cryptographic functions used by the TOE are implemented using the cryptographic library FMSSL. This cryptographic library has been tested and approved by the Swedish NCSA. FMSSL is outside the TOE scope, and therefore its internals are not covered by the evaluation.

1.5 TOE Description

1.5.1 Introduction

This document describes the architecture of the fourth generation Färist. The Färist system started off as a firewall and the VPN functionality was added in the second generation. The new generation is based on the Färist Micro VPN platform. This is a pure VPN platform where the firewall functionality is now instead added as a filter functionality.

The architecture is implemented in the Färist and a slightly simplified version of the architecture is implemented in the Färist Micro. The main difference is that the Färist micro only has two Ethernet interfaces.

Encryption is done on either IP-packets or Ethernet packets, which means that the Färist will function as an IP Router or Ethernet Switch. The encrypted packets can be transmitted according to either the IPSec or MACSec standard.

1.5.2 Intended use

The intended use of the TOE is to provide a VPN tunnel between two networks over a carrier network. The Färist VPN tunnel separates the traffic in the tunnel from the carrier network. This may be used either to extend an organisations local network over untrusted carrier networks or to tunnel untrusted network traffic through a classified carrier network.

The device at the other end of the tunnel has to be a supported Färist model.

To operate the TOE securely the operating environment must:

- Physically protect the TOE from unauthorised access at all times
- Contain a log server
- Have the ability to generate certificates of high quality

It is recommended that the environment also contain:

- An NTP-server
- An administrative client

1.5.3 Architecture

1.5.3.1 Overview

The TOE consist of two subsystems, the control plane and the data plane. The data plane is further subdivided into a "local traffic" and a "bulk traffic" part. The control plane runs on top of a Linux kernel while the data plane operates directly on top of the hardware. All packets are received by the data plane and a check is first made to see if the packet is directed to the TOE itself, in which case the traffic is filtered before it is sent to the control plane. All other traffic is sent to the crypto engine and filter system in the data plane. The components of all subsystems are described below.

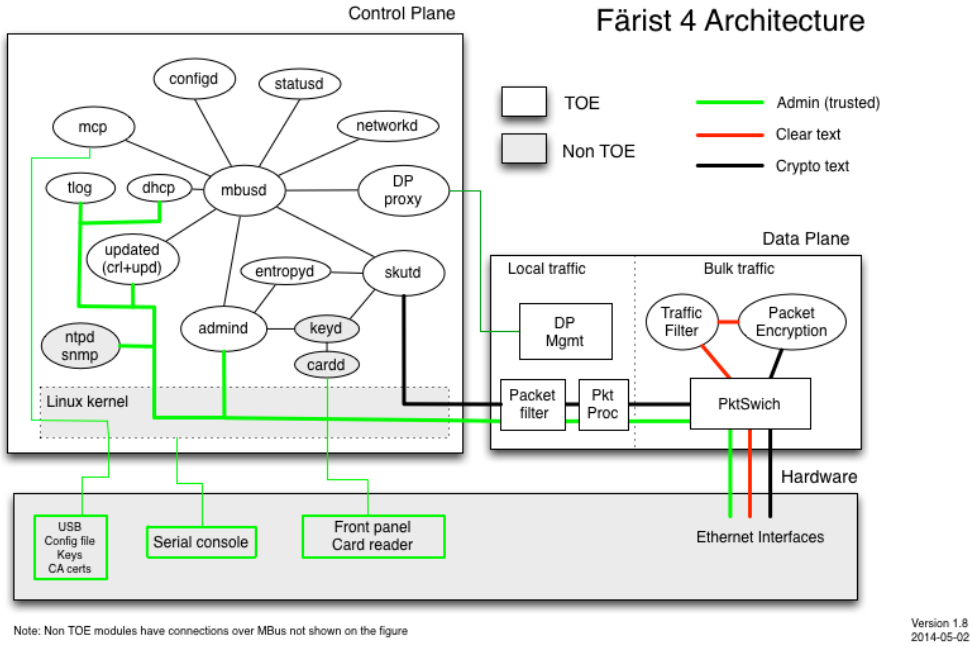

Illustration 2: Färist 4 Architecture

The network interfaces for user traffic are designated "crypto" (i.e. carrier network) and "clear" (i.e. tunnelled network). The designation indicates which kind of information is flowing through them.

1.5.4 Hardware

1.5.4.1 Färist

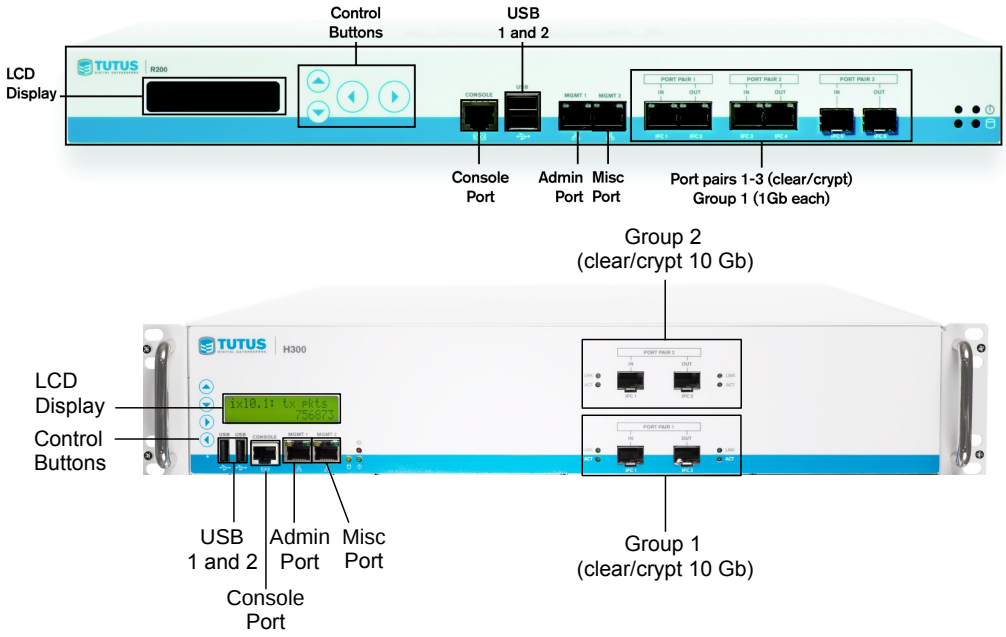

Illustration 3: Färist - Front View

Two Färist hardware devices are evaluated, they are both high end Intel Xeon systems with a serial console, dedicated Ethernet ports and a front panel for management. The R200 has 3 port pairs of 1Gb/s while the H300 has 2 port pairs of 10Gb/s. The configuration is stored on a small USB memory that must be present at all times while the system is running. The keys can either be stored on the USB-memory or on a Smart Card.

1.5.4.2 Färist Micro

Illustration 4: Färist Micro

The hardware for the Färist Micro is named C200 and is a low power Intel system with a serial console and a card reader with a keypad. The card reader is used both to handle the smart card and as a detachable front panel used for simple management. The micro uses one clear and one crypto interface. The management interface is emulated internally by the firmware as all administrative traffic is pushed through the tunnel.

1.5.5 Control Plane

The control plane handles all administrative roles in the TOE, including key negotiations. It consists of a number of tightly integrated components which run on top of a standard Linux kernel.

The components of the control plane subsystem are listed and described below:

mcp	Master component service – bootstraps the software and performs integrity checks of both the control plane and data plane.
mbusd	System multiplexor – interconnects the system components.
configd	Configuration service – configures the software.
config file	The configuration file describes all configuration options for the whole TOE. It is either saved internally or on a removable USB memory and holds the entire configuration state of the TOE. The config file can also contain CA-certificates.
DP-proxy	Bootstraps the data plane and translates commands exchanged

	between control plane and data plane. The exchange is done over a local socket.
networkd	Manages all IP configuration in the system, including setup interface addresses and routes in control plane and sending commands with respective IP configuration data to data plane.
updated	CRL and system update server – processes CRL files and performs update package downloads. It will connect to configured update server and request a new version of the software. It verifies the digital signature of the update using a factory installed update signing certificate. The update daemon will also verify that that the software version in the update is newer than the running version.
statusd	Status service – collects and prints out statistics and machine status on the front panel. Can also accept administrative commands from the front panel.
keyd	This service provides signing with the private key of the Färist. Signing is done using a key stored on the USB-memory or a Smart Card.
cardd	This service manages the smart card and the front panel.
skutd	Negotiates keys with remote peers.
tlog	Log daemon, accepts log messages from all components and sends them to a remote log server.
ntpd	Adjust system time against a remote NTP server.
dhcp	The DHCP subsystem. It can negotiate addresses for the interfaces and work as a client or server.
Snmp	Sends SNMP-traps to remote receiver/s on behalf of control plane modules.
admind	Remote administration daemon exposes a simple management interface to an authenticated administrator. The supported commands include: • Get configuration • Save configuration • Activate configuration • Get status of the TOE with traffic load and similar information The commands are in form of HTTP requests/responses. The administration is done through a separate management client.
Keys	Asymmetric keys stored in a PKCS#12 archive. The keys are either stored on the same USB memory that holds the configuration file or they are stored on an external smart card. The asymmetric keys are only used for key negotiation, all encryption is done employing AES-256 symmetric encryption algorithm with the negotiated keys.
entropyd	Collects entropy thorough various means in the system and keeps an entropy state on disk. Serves random seed to components requesting it through a Unix socket.
Linux kernel	Standard Linux kernel with special drivers to manage the data

plane and receive and send IP-packets to it.

1.5.6 Data Plane

The data plane is running directly on the hardware and is responsible for all packet processing. The data plane is further divided into a bulk part and a local part. The bulk part has a very optimized design to achieve high performance in the bulk packet handling. The local part is responsible for packets directed to the TOE itself and the communication with the control plane. The data plane consist of five modules which are listed and described below:

DP Mgmt	The management component is responsible for all configuration and management of the data plane.
PktSwitch	The packet switch is a high performance component performing packets validation, classification and multiplexing to other modules.
Packet Encryption	The raw packet encryption is done in this component.
Traffic Filter	A traffic filter can optionally be configured to filter traffic packets. A filter can accept or deny a packet, logging is also possible. Traffic filters can also call functions in filter plugins to support more advanced filtering. This is how firewall functionality is added.
Packet filter	The Packet filter is responsible for filtering packets destined to the control plane.

1.5.7 Administration

The TOE requires a configuration file and keys in order to work as intended. Administration is limited to manipulating configuration files and keys and to obtain output of status and log events.

Administration can be done using two different interfaces:

- **Front panel:** The front panel can be used to see the running status of the TOE and perform some simple administrative tasks.
- **Remote administration:** Using a separate administration client the TOE can be remotely administrated through an authenticated TLS interface.

In addition, the USB interface is used by the administrator to import management data into the TOE.

Different administrative functions are available on these interfaces. The front panel can be used to view the status of the running device as well as performing simple administrative tasks like changing keys and shutting down the device.

Remote administration can be used to view status of the device, update the configuration and read the log.

1.5.8 Physical scope of the TOE

The TOE is software only. It consists of the Färist firmware v4.0.1 and Färist Micro firmware v4.0.1.

Relevant guidance documents for the secure operation of the Färist and Färist Micro that are part of the TOE are:

- Färist 4 Administrator's manual

Physical boundaries between the TOE and its runtime environment are described in section 1.5.3. In particular:

- The base Linux operating system, other than particular applications implementing TSF, is considered part of the runtime environment.
- The hardware is considered part of the runtime environment. The following TOE and hardware appliance model combinations are covered by this evaluation:
    - Färist running on R200
    - Färist running on H300
    - Färist Micro running on C200

The following components can be found in the operational environment of the TOE on systems other than those hosting the TOE:

- Client software, the administrative client is not part of the TOE.
- Log servers (e.g. syslog server and analysis tools used for collecting the audit records and for their analysis) and NTP servers.

1.5.9 Evaluated configurations

There are two different evaluated configurations, both create encrypted tunnels, but they use different protocols.

IPSec

The IPSec tunnel configuration creates a tunnel at OSI layer 3. This means that the Färist will create a tunnel through a routed IP-network. Encryption in this configuration is done on either IP-packets or Ethernet packets. This makes the Färist function as an IP Router or Ethernet Bridge.

MACSec

The MACSec tunnel configuration creates a tunnel at OSI layer 2. This means that the Färist will need a physical transit network without any switches or similar components in between the Färists. Encryption in this configuration is done on the Ethernet packets making the Färist function as an Ethernet Bridge. a

Note: Only IPSec routed configuration is possible for the Färist Micro.

1.5.10 Logical scope of the TOE

This section provides an overview of the security functions implemented by the TOE.

1.5.10.1 VPN

The VPN functionality is the main security functionality of the Färist. It creates private networks over public networks using encryption.

The key exchange and authentication is done with the SKUT protocol [SKUT3] using an RSA-signed Diffie-Hellman key exchange. Encryption is done using either IPSec ESP [RFC4303] in tunnel mode or MACSec [IEEE802.1AE] with the AES encryption algorithm.

The cryptographic library for the VPN functionality is implemented in a proprietary library which has been approved by the Swedish NCSA.

The tunnels are configured using a configuration file and the RSA certificate is stored in a file on USB memory or on a smart card.

1.5.10.2 Audit

Audit records that are created in different components are sent to a log daemon (tlogd), that will forward the audit records to one or more remote log servers.

If the log servers are not reachable the log will be kept in RAM until it can be sent.

The last few log events are also stored in a local copy of the audit file, which is kept in RAM. This is used for troubleshooting and is not considered a TSF.

1.5.10.3 Management

The TOE can have the following administrative roles:

- Administrator
  - Remote administrator
    The remote administrator can perform all administrative tasks. Changing keys however requires local access to the machine, which means that by not allowing the administrator local access key management can be handled solely by the key manager.
    The remote administrator is authenticated through his certificate.
  - Local administrator
    The local administrator can perform the basic administrative tasks using the front panel.
    The local administrator is authenticated through organizational means, by allowing only authorized personnel physical access to the TOE.
- Operator
  The (remote) operator can perform the same tasks as the remote administrator but cannot perform any changes, such as changing configuration. The operator is authenticated through his certificate.
- Key manager
  The key manager can change cryptographic keys by either changing the smart card or reading a new key from the USB memory.

Remote management is secured through certificates. The remote user is identified through the CN (common name), UID (Unique Identifier) or Email (RFC822 email) field in his certificate.

The cryptographic library for the remote management functionality is implemented in a proprietary library which has been approved by the Swedish NCSA.

Local management is secured through organizational means and is therefore not considered a TSF.

1.5.10.4 Autoupdate

The TOE has an automatic update functionality that checks for new firmware updates, verifies the origin and integrity of the update and ensures that the update is newer than the current version. Note that it is the update function that is part of the TOE and not the updated version of the TOE.

1.5.10.5 Selftest and failsafe

The TOE has built-in functionality self tests that are run both at startup and at regular intervals. Self tests verify the integrity of the system files and ensure the proper working of the encryption engine. If a self test fails the TOE will preform a restart.

If the USB-memory or smart card holding the RSA certificate/private key is removed the TOE will also restart.

1.5.11 Operational environment support

1.5.11.1 Physical environment

The TOE must be protected against unauthorised access at all times.

1.5.11.2 IT environment

The IT environment **must** contain the following:

- Log server
- CA – certificate generation

The IT-environment **may** contain the following:

- NTP server. An NTP server on the administrative network can be used by the TOE to set correct time.
- Administrative client

2 CC Conformance Claim

This ST is CC Part 2 extended and CC Part 3 conformant, with a claimed Evaluation Assurance Level of EAL5, augmented by ALC_FLR.1.

This ST does not claim conformance to any Protection Profile.

Common Criteria [CC] version 3.1 revision 4 is the basis for this conformance claim.

3 Security Problem Definition

The TOE is set up between a local network and a remote Färist connected to its remote network. It creates a protected channel (tunnel) between the local and remote networks over the carrier network. The tunnel may be used to tunnel trusted traffic over an untrusted carrier network or the tunnel is used to tunnel untrusted traffic over a trusted network.

The TOE is not only in itself an asset, but it aims to protect all assets which are (typically) placed in the internal network and therefore shall be protected appropriately.

The TOE is intended to be used in a physically protected environment. It is assumed that no unauthorised personnel has physical access to the TOE. Therefore all attacks to the TOE have to be performed over the network connections of the TOE. Either from the untrusted carrier network or from the connected networks and the tunnel of the untrusted traffic.

The underlying operating system and hardware are by design dedicated for the TOE and can not be used for other tasks or applications.

It is assumed that the underlying hard- and firmware operates according to their specifications and have no security critical side-effects on the operation of the TOE. Hard- and firmware are not part of this TOE, but of course the functions of the TOE rely on them.

Furthermore the TOE is assumed to operate in an environment where interception of radiation is covered by other environmental measures. The evaluation will therefore not address vulnerabilities caused by emanation from the TOE.

Remote administrators and operators of the TOE authenticated with a TLS certificate, as well as local administrators, are considered to be trustworthy. The TOE will not protect itself against an administrator who tries to bring the TOE into an insecure state. It is also assumed that administrators are well trained, reducing the risk that they accidentally make security critical administration mistakes.

3.1 Threat Environment

This section describes the threat model for the TOE and identifies the individual threats that are assumed to exist in the TOE environment.

The **assets** to be protected by the TOE are:

1. Organizational data being exchanged between the local and the remote trusted network, as in the first use case. (The TOE is then used to protect the assets on those systems from unauthorized exploitation by providing a secure channel between these networks.)
2. Organisational data that are part of the trusted network in which the channel of the untrusted traffic is being tunnelled, as in the second use case. (The TOE is then used to protect the assets in the trusted network from untrusted traffic by providing a secure channel for tunnelling the untrusted traffic.)
3. The TSF, the configuration data (including encryption keys) and the audit data, in particular the availability of the TSF to legitimate users, especially to authorized administrators through the designated management interfaces.

The exact of definitions of TSF and TSF data is given by the actual design and implementation of the TOE.

The **threat agents** having an interest in manipulating the TOE and TSF behaviour to gain access to these assets can be categorized as:

1. Unauthorized third parties (“attackers”, such as malicious remote users, parties, or external IT entities) which are unknown to the TOE and its runtime environment, but may attempt to interact with the TOE. Attackers are traditionally located outside the organizational environment that the TOE is employed to protect, but may include organizational insiders too.
2. Authorized administrators of the TOE are assumed to be trustworthy, trained and to follow the instructions provided to them with respect to the secure configuration and operation of the systems under their responsibility. Hence, only inadvertent attempts to manipulate the safe operation of the TOE are expected from this community.

The motivation of threat agents is assumed to be commensurate with the assurance level pursued by this evaluation, i.e., the TOE intends to resist penetration by attackers with a **Moderate** attack potential.

Although the administrators are assumed to be trustworthy and trained, we cannot exclude that mistakes are being made. This as well as insider attacks may not be prevented, but are to a large extent addressed by the OSPs P.ADMACC and P.AUDIT.

3.1.1 Threats countered by the TOE

The threats specified below are addressed by the TOE.

T.DISCLOSE An external attacker gains unauthorised access to information transmitted between the TOE and a remote trusted network.

T.CHANNEL An external attacker gain unauthorized access to information or resources in the trusted network by breaking out of the secure channel over the trusted network. (As in the second use case.)

T.INISEC For configuration settings which are not provided by an administrator, insecure default values may be set by the TOE.

T.MEDIATE An attacker on the clear interface sends information through the TOE to the crypto interface without sending it through the trusted channel; and an attacker using the trusted tunnel on the crypto interface will break out of the trusted channel and generate traffic on the crypto interface outside of the trusted channel.

T.MODIFY The attempts of an external attacker to modify data transmitted between the TOE and a remote trusted network goes undetected.

T.ADMIN An attacker may be able to perform administration or configuration of the TOE, or gain access to administration information and configuration data, such as secret keys or audit records, or may be able to modify such data.

T.SELPRO An attacker may read, modify, or destroy TOE internal data by transmitting data to the TOE via one of its network connections that causes modification or deletion of TOE internal data.

T.UPDATE Attacker may provide malicious TOE updates or old versions of the TOE software to introduce back-doors or exploitable weaknesses into the TOE.

3.2 Organizational Security Policies

The organisational security policies are specified making demands on the accountability of administrator actions:

P.ADMACC Administrators shall be accountable for the actions they conduct by generating sufficient audit records for the actions.

P.AUDIT The TOE shall be able to record all of its security relevant actions.

P.CONFIG The TOE shall support the means to configure and manage the TSFs.

3.3 Assumptions

This section specifies the assumptions that must be satisfied by the TOE environment.

A.AUDIT The TOE environment must be able to receive, store and

protect the audit records generated by the TOE and provide the means for the audit analysis.

A.DHPARA The Diffie-Hellman parameters of the TOE and the remote host are of good quality.

A.KEYS It is assumed that private RSA keys used for remote administration and the VPN tunnel are of high quality and not disclosed.

A.NOEVIL Authorised administrators given privileges, are competent, non-hostile and follow all their guidance; however, they are capable of error.

A.NOEMA Interception of emanation of any kind is addressed by environmental controls that reduce the signal to noise ratio for an interceptor to a level that prohibits useful evaluation of the intercepted signals.

A.PHYSEC The TOE is physically secure, i.e. no unauthorised persons have physical access to the TOE and its underlying system.

A.RELHARD The underlying hardware, firmware (BIOS and device drivers) and the operating system functions needed by the TOE to guarantee secure operation, are working correctly and have no undocumented security critical side effect on the functions of the TOE.

A.TIME The TOE environment provides the TOE with a reliable time stamp.

4 Security Objectives

The security objectives provide a concise statement of the intended response to the security problem. It will describe which security needs will be addressed by the TOE and which will be addressed by the TOE environment, in the form of a statement of security objectives.

4.1 Objectives for the TOE

The following are the IT security objectives to be met by the TOE.

O.MEDIATE The TOE must mediate the flow of all information flowing between the TOE's crypto and clear interfaces and ensure that all traffic from the internal network is passed though the trusted channel and ensure that only traffic coming from the channel is passed to the clear interface.

O.CHANNEL The TOE must be able to provide trusted channels to remote trusted networks and protect information transmitted to and received from such networks against unauthorised disclosure and to detect any modification of incoming information transmitted from such networks, and to provide the means for the remote network to verify the integrity of information transmitted out of the TOE to such networks.

O.AUDIT The TOE must be able to provide audit evidence of security relevant events as well as for authorised use of security functions to allow an authorised administrator to read the audit

trail.

O.CONFIG	The TOE must provide the means for an authorized administrator to configure and manage the TOE security functions.
O.LIMEXT	The TOE must restrict the means to configuration and control of the TOE to authorised administrators.
O.REMOTE	The TOE must uniquely identify and authenticate the identity of all administrators and provide them with a secure communication channel before allowing remote administrators any access to the TOE.
O.SECSTA	Upon initial start-up of the TOE or during configuration, the TOE shall provide well-defined initial settings for security relevant functions.
O.SELPRO	The TOE must protect itself against attempts by attackers to bypass, deactivate or tamper with TOE security functions.
O.UPDATE	The TOE must only accept updates that are newer than the current running version and updates where the origin and integrity can be trusted.

4.2 Objectives for the Operational Environment

The following are the security objectives for the operational environment of the TOE that are necessary for the TOE to meet its security objectives.

Thus, the following environmental objectives may partly be IT specific and partly related to administrative methods and/or procedural measures.

OE.AUDIT	The TOE environment must be able to receive, store and protect the audit records generated by the TOE and provide the means for the audit analysis.
OE.DHPARA	The Diffie-Hellman parameters of the Färist and the remote host are safe and not arbitrarily generated.
OE.KEYS	It is assumed that private RSA keys used for remote administration and the VPN tunnel are of high quality and not disclosed.
OE.NOEVIL	Authorised administrators and operators given privileges, are competent, non-hostile and follow all their guidance; however, they are capable of error.
OE.NOEMA	Interception of emanation of any kind is addressed by environmental controls that reduce the signal to noise ratio for an interceptor to a level that prohibits useful evaluation of the intercepted signals.
OE.PHYSEC	The TOE is physically secure, i.e. no unauthorised persons have physical access to the TOE and its underlying system.
OE.RELHARD	The underlying hardware, firmware (BIOS and device drivers) and the operating system functions needed by the TOE to guarantee secure operation, are working correctly and have no undocumented security critical side effect on the functions of the TOE.
OE.TIME	The TOE environment provides the TOE with a reliable time

stamp.

4.3 Security Objectives Rationale

4.3.1 Security Objectives Coverage

The following tables provide a mapping of security objectives both for the TOE and the TOE environment to the environment defined by the threats, policies and assumptions, illustrating that each security objective for the TOE covers at least one threat or policy, and that each security objective for the TOE environment covers at least one policy, threat or assumption.

	T.DISCLOSE	T.INISEC	T.MEDIATE	T.MODIFY	T.ADMIN	T.SELPRO	T.UPDATE	P.ADMACC	P.AUDIT	P.CONFIG	A.AUDIT	A.DHPARA	A.KEYS	A.NOEVIL	A.NOEMA	A.PHYSEC	A.RELHARD	A.TIME
O.MEDIATE			X															
O.CHANNEL	X			X														
O.AUDIT								X	X									
O.CONFIG										X								
O.LIMEXT					X													
O.REMOTE					X			X										
O.SECSTA		X																
O.SELPRO						X												
O.UPDATE							X											
OE.AUDIT									X		X							
OE.DHPARA	X			X								X						
OE.KEYS	X			X	X								X					
OE.NOEVIL														X				
OE.NOEMA	X														X			
OE.PHYSEC					X											X		
OE.RELHARD																	X	
OE.TIME								X	X									X

Table 1: Security objective coverage

4.3.2 Security Objectives Sufficiency

The following rationale provides justification that the security objectives are suitable to counter each individual threat and that each security objective tracing back to a threat actually contributes to the mitigation of that threat.

Threat	Rationale for the security objectives
T.DISCLOSE	By requiring the TOE to be able to provide a trusted channel against disclosure and against modification as in O.CHANNEL, the threats of T.DISCLOSE and T.MODIFY are respectively being met. This is also supported by the assumption that the cryptographic parameters and keys provided by the environment are secure as in OE.DHPARA and OE.KEYS. OE.NOEMA ensures that no information disclosure is caused by emanations of the TOE.
T.INISEC	By requiring well-defined default setting in O.SECSTA, an initial insecure configuration of the TOE is prevented and the threat

Threat	Rationale for the security objectives
	T.INISEC of insecure configuration due to lack of administrator settings is removed.
T.MEDIAT	By demanding that the TOE must mediate (i.e. control) all data sent between networks connected to the TOE as in O.MEDIAT, ensuring that only data coming from the trusted channel are passed to the internal network and data from the internal network will only be sent to the trusted channel to the remote network.
T.MODIFY	By requiring the TOE to be able to provide a trusted channel against disclosure and against modification as in O.CHANNEL, the threats of T.DISCLOSE and T.MODIFY are respectively being met. This is also supported by the assumption that the cryptographic parameters and keys provided by the environment are secure as in OE.DHPARA and OE.KEYS.
T.ADMIN	The threat of a non-administrator performing administration of the TOE as in T.ADMIN is addressed by requiring that all remote administration is performed using a remote trusted channel, based on OE.KEYS, with identified and authorized administrators as in O.REMOTE, and by further restricting administration to these administrators as in O.LIMEXT. T.ADMIN is only relevant for the remote administration, because no unauthorized personal can physically access the TOE as demanded by OE.PHYSEC. OE.PHYSEC fulfils A.PHYSEC.
T.SELPRO	By protecting itself against bypass, deactivation and tampering as in O.SELPRO, the threat T.SELPRO is diminished to an acceptable level.
T.UPDATE	By only accepting newer versions of the TOE, the TOE will prevent that older versions with possibly known vulnerabilities will be installed (O.UPDATE). The TOE will also verify the origin and integrity of the TOE version and thereby prevent malicious version from being installed (O.UPDATE).

Table 2: Security objectives rationale for the threats

The following rationale provides justification that the security objectives of the TOE and the TOE environment are suitable to address each individual assumption and that each security objective tracing back to a OSP actually contributes in addressing the OSP.

OSP	Rationale for the security objectives
P.ADMACC	The auditing of administrator actions as in O.AUDIT, assisted by correct time delivery in OE.TIME and unique identification in O.REMOTE, satisfies the organisational security policy of administrators being accountable for their actions as in P.ADMACC. It is further supported by OE.AUDIT that the TOE environment must be able to receive, store and protect the audit records generated by the TOE and provide the means for the audit analysis.
P.AUDIT	The auditing of security relevant reactions is addressed by O.AUDIT, assisted by correct time delivery in OE.TIME, satisfies the organisational security policy that the TOE shall be able to record all of its security relevant actions as in P.AUDIT. It is further supported by OE.AUDIT that the TOE environment must be able to receive, store and protect the audit records generated by the TOE and provide the means for the audit analysis.
P.CONFIG	The management the TOE security functions is addressed by O.CONFIG, which ensures that the TOE provides the means for an

OSP	Rationale for the security objectives
	authorized administrator to configure and manage the TOE security functions as in P.CONFIG.

Table 3: Security objectives rationale for the OSPs

The following rationale provides justification that the security objectives of the TOE environment are suitable to address each individual assumption and that each security objective tracing back to an assumption actually contributes in addressing the assumption.

Assumption	Rationale for the security objectives
A.AUDIT	Addressed by OE.AUDIT which is is identical to the assumption
A.DHPARA	Addressed by OE.DHPARA which is is identical to the assumption
A.KEYS	Addressed by OE.KEYS which is is identical to the assumption
A.NOEVIL	Addressed by OE.NOEVIL which is is identical to the assumption
A.NOEMA	Addressed by OE.NOEMA which is is identical to the assumption
A.PHYSEC	Addressed by OE.PHYSEC which is is identical to the assumption
A.RELHARD	Addressed by OE.RELHARD which is is identical to the assumption
A.TIME	Addressed by OE.TIME which is is identical to the assumption

Table 4: Security objectives rationale for the assumptions

5 Extended Components Definition

The extended requirement, FPT_TUD_EXT.1 for trusted updates is used to specify the SFR for automatic trusted updates. It has been based on the extended component which defined by [NDPP] Protection Profile for Network Devices published by NIAP in June 2012.

5.1 FPT_TUD_EXT – Trusted Updates

Family behaviour

This family defines the requirements for the TSF to be able to create and maintain a secure audit trail. Stored audit records refers to those records within the audit trail, and not the audit records that have been retrieved (to temporary storage) through selection.

Component levelling

FPT_TUD_EXT.1 is not hierarchical.

Management

While management functions have been specified as part of this component already, the following actions could be considered for the management functions in FMT: Administrator initiation of updates, activation and deactivation of automatic updates, time for initiation of updates or specification of certificates used for signature verification.

Audit

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

1. Minimum: Software update.

2. Minimum: Failure of verification (digital signature, published hash or version number)

5.1.1 FPT_TUD_EXT.1 Trusted Update

Hierarchical to: none

Dependencies: FCS_COP.1 Cryptographic operation

FPT_TUD_EXT.1.1 The TSF shall provide administrators the ability to query the current version of the TOE software.

FPT_TUD_EXT.1.2 The TSF shall provide a mechanisms that [select: on a regular basis initiates, gives administrators the ability to initiate] updates to TOE software.

FPT_TUD_EXT.1.3 The TSF shall provide a means to verify software updates to the TOE using a [selection: digital signature mechanism, published hash] prior to installing those updates.

FPT_TUD_EXT.1.4 The TSF shall provide a means to verify software updates to the TOE to ensure that software update version is newer than the current version of the TOE prior to installing those updates.

Application note: The digital signature mechanism and hash mechanisms referenced in the third element must be specified in FCS_COP.1. The ST author should choose the mechanism implemented by the TOE; it is acceptable to implement both mechanisms.

6 Security Requirements

6.1 TOE Security Functional Requirements

The following table gives an overview of the functional components from the Common Criteria Part 2 that are relevant for this TOE.

Component	Component Name
FAU_GEN.1	Audit data generation
FCS_CKM.1	Cryptographic key generation
FCS_CKM.2	Cryptographic key distribution (iteration a, b)
FCS_COP.1	Cryptographic operation (iteration a, b, c, d, e)
FDP_ACC.2	Complete access control
FDP_ACF.1	Security attribute based access control
FDP_IFC.2	Subset information flow control
FDP_IFF.1	Simple security attributes
FDP_RIP.1	Subset residual information protection
FIA_ATD.1	User attribute definition
FIA_UAU.2	User authentication before any action
FIA_UID.2	User identification before any action
FMT_MOF.1	Management of security functions behaviour
FMT_MSA.1	Management of security attributes
FMT_MSA.3	Static attribute initialisation
FMT_MTD.1	Management of TSF data (iteration a, b)
FMT_SMR.1	Security roles
FMT_SMF.1	Specification of management functions
FPT_FLS.1	Failure with preservation of secure state
FPT_STM.1	Reliable time stamps
FPT_TSF.1	Self test
FPT_TUD_EXT.1	Trusted update
FTP_ITC.1	Inter-TSF trusted channel

Table 5: Functional Requirements on the TOE

The following paragraphs give an overview on the functional requirements listed in the table above with respect to the TOE. They serve as an introduction to the detailed definition of the functional requirements, which are presented in the next section.

Class FAU components are selected to describe the capability of the TOE to generate, read and protect audit data. The TOE generates audit data for events associated with the communication links it monitors. Administrators are able to select the audited events, and the log data is transferred, read and analysed in the environment.

Class FCS contains the requirements related to cryptographic operations. There are two areas in which the TOE performs cryptographic operations: Remote administration using TLS and the IPSec/MACSec VPN connection which is also using TLS and SKUT for key management. The TOE is importing certificates for the remote administration and for the VPN.

Class FDP contains the security requirements associated with the access control between remote administrators and configuration data of the TOE. The class also contains the security requirements associated with the information flow control between the clear and crypto network interface of the TOE. Information flow control is enforced both by the TOE subsystems as well as by the VPN tunnel.

Class FIA contains the security requirements for identification and authentication of a remote administrator performing administrative tasks. The authentication is relying on the TLS authentication using X.509 certificates.

Class FMT contains the security management requirements. This includes the security management role of the administrator, the management functions available to the administrator and that the initial default values of the TOE will be well-defined.

Class FPT contains the requirement for the protection of the TSF, which contains the requirements for the preservation of secure state, reliable time stamps, self test and trusted update.

Class FTP contains requirements for trusted communication path between the TSF and other trusted IT products, i.e. the VPN connection.

6.1.1 Security Functional Policies implemented by the TOE

6.1.1.1 TRAFFIC SFP

The TOE will implement the information flow control policy SFP named TRAFFIC SFP. The TSF shall enforce the SFP on the traffic sent to and from the clear and crypto interface, and the VPN channel established to external IT entities that send and/or receive data using a VPN through the TOE. The policy is named TRAFFIC SFP to indicate that connections are between entities that are authenticated.

The TRAFFIC SFP applies to the physical clear (i.e. internal) and crypto (i.e. external) interfaces of the TOE. It does not apply to the physical admin interface.

The TSF shall apply the following rules to all traffic using the TRAFFIC SFP:

- When using IPSec:
  - IP packets arriving on the clear interface where the destination IP address is part of the IP networks to be reached are to be routed to the respective network.
  - IP packets arriving on the clear interface, where the IP destination address is not part of the IP networks to be reached using the VPN, are rejected.
  - IP packets arriving on the crypto interface coming through the VPN channel are to be routed to the clear interface.
- When using MACSec or IPSec bridged:
  - Ethernet frames arriving on the clear interface are sent to the remote network over the VPN channel.
  - Ethernet frames arriving on the crypto interface coming through the VPN channel are sent to the clear interface.
- Packets arriving on the crypto interface not coming through the VPN channel are rejected, except for the following list of packets which are to be processed by the TOE:
  - SKUT traffic
  - ICMPv4 Fragmentation Required (IPSec only)
  - ICMPv4 Echo Request (IPSec only)
  - ICMPv4 Echo Reply (IPSec only)

No other traffic shall be routed using the TRAFFIC SFP.

6.1.1.2 ADMINISTRATOR ACCESS SFP

The TOE will implement the access control policy ADMINISTRATOR ACCESS SFP. The TSF shall enforce identification and authentication of remote administrators and operators before giving any administrative access to the TOE (i.e. giving any access to TSF data).

6.1.2 Class FAU – Security Audit

6.1.2.1 FAU_GEN.1 – Audit Data Generation

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;

b) All auditable events for the **not specified** level of audit; and

c) the following events:

1. **connect and disconnect of a VPN channel (including connection attempts)**
2. **renegotiation of the VPN key (IPSec/MACSec)**
3. **operations performed by remote administrators and operators (including connection attempts)**
4. **changes to the configuration**

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 ST: **none**

Application note: Although the TOE is capable to generate additional audit events, these events are not considered security events and not necessary to satisfy the TOE security objectives and therefore not considered part of the TSF.

6.1.3 Class FCS – Cryptographic Support

6.1.3.1 FCS_CKM.1 Cryptographic key generation

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **as defined in the TLS v1.0 standard [RFC2246] for AES-256 [FIPS197] and HMAC with SHA-1 [RFC2104] and [FIPS180-4] keys** and specified cryptographic key sizes **256 bit (AES-256) and 160 bit (HMAC)** that meet the following: **generation and exchange of session keys as defined in the TLS v1.0 standard with the cipher suites defined in FCS_COP.1b and FCS_COP.1d**.

Application note: The session keys are negotiated and established during an TLS session for remote administration (i.e. remote administrators and operators) as well as the VPN for symmetrical encryption and integrity protection of VPN packets. The TLS

standard allows other cryptographic algorithms and key sizes, but only AES-256 and SHA-1 are supported. This functionality is provided by a TLS server on the server side. The administration client provides corresponding functionality on the TLS client's side (as part of the environment).

The key destruction of session keys as specified by CC in FCS_CKM.4 is covered by FDP_RIP.1 Subset residual information protection.

6.1.3.2 FCS_CKM.2a – Cryptographic Key Distribution (cert)

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **of digital certificates** that meets the following: **X.509 Version 3 [RFC5280]**.

Application note: This requirement addresses the exchange of X.509 certificates as part of the TLS authentication of the remote administration (i.e. remote administrators and operators) and the key management of the VPN channel.

6.1.3.3 FCS_CKM.2b – Cryptographic Key Distribution (keys)

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **TLS handshake using DH_RSA key exchange of AES-256 session keys and HMAC keys** that meets the following: **TLS v1.0 [RFC2246]**.

Application note: This requirement addresses the exchange of AES-256 session keys and HMAC keys as part of the TLS handshake protocol using Diffie-Hellman RSA for remote administration and peer configuration used as part of the VPN. No other key exchange than DH_RSA is accepted.

6.1.3.4 FCS_COP.1a – Cryptographic Operation (RSA)

FCS_COP.1.1 The TSF shall perform **digital signature generation and verification** in accordance with a specified cryptographic algorithm **RSA [RSASSA-PKCS1-v1_5]** and cryptographic key sizes **2048 bit** that meet the following: **[PKCS1v2.1]**.

Application note: This requirement addresses the RSA digital signature generation and verification operations using the RSA algorithm as required by the TLS session establishment protocol.

6.1.3.5 FCS_COP.1b – Cryptographic Operation (AES-CBC)

FCS_COP.1.1 The TSF shall perform **encryption and decryption** in accordance with a specified cryptographic algorithm **AES-256 in CBC mode** and cryptographic key sizes **256 bit** that meet the following: **[FIPS197] and [NIST SP 800-38A]**.

Application note: This is used by the TLS used for administrator authentication, for the SKUT key management used by the VPN. If a client or VPN node tries to use any other cipher suite, the client or peer will be rejected by the TOE.

6.1.3.6 FCS_COP.1c – Cryptographic Operation (AES-GCM)

FCS_COP.1.1 The TSF shall perform **encryption and decryption** in accordance with a specified cryptographic algorithm **AES-256 in GCM mode** and cryptographic key sizes **256 bit** that meet

the following: **[FIPS197] and [NIST SP 800-38D]**.

Application note: This is used by the VPN and for the IPSec or MACSec payload If a client or VPN node tries to use any other cipher suite, the client or peer will be rejected by the TOE.

6.1.3.7 FCS_COP.1d – Cryptographic Operation (SHA)

FCS_COP.1.1 The TSF shall perform **message digest generation and verification** in accordance with a specified cryptographic algorithm **HMAC with SHA-1** and cryptographic key sizes **160 bit** that meet the following: **[RFC2104] and [FIPS180-4]**.

Application note: The TLS standard allows other ciphers, but the TOE supports only SHA-1. If a client or VPN node tries to use any other another cipher suite for the message digest, the client or peer will be rejected by the TOE.

6.1.3.8 FCS_COP.1e – Cryptographic Operation (RSASSA-PKCS1-v1_5)

FCS_COP.1.1 The TSF shall perform **signature verification** in accordance with **S/MIME version 3.2 [RFC5751]** and cryptographic key sizes **2048 bit** that meet the following: **RSA [RSASSA-PKCS1-v1_5] and SHA-1 [FIPS180-4]**.

Application note: The S/MIME standard allows other ciphers, but the TOE supports only SHA-1 [FIPS180-4] for hashing and RSA [RSASSA-PKCS1-v1_5] 2048 bit for signing.

6.1.4 Class FDP – User Data Protection

6.1.4.1 FDP_ACC.2 – Complete access control

FDP_ACC.2.1 The TSF shall enforce the **ADMINISTRATOR ACCESS SFP** on **the subjects:**

- **remote administrators**
- **operators**

and objects:

- **configuration data of the TOE**
- **audit data locally stored in the TOE**

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 FDP_ACF.1 – Security attribute based access control

FDP_ACF.1.1 The TSF shall enforce the **ADMINISTRATOR ACCESS SFP** to objects based on the following:

subject remote administrator and operator:

- **TLS certificate**
- **CN, UID or Email field of the certificate**

objects (configuration data of the TOE, resources in the internal network):

- **none.**

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

- **If the CN, UID or Email field of the subject's certificate is part of a list managed by the TOE that allows to connect as TLS client to the TOE, the user is allowed access to resources on the TOE.**

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **none**.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules:

If the client certificate is not signed by a certificate of a certification authority trusted by the TOE, then access is denied.

6.1.4.3 FDP_IFC.2 – Complete information flow control

FDP_IFC.2.1 The TSF shall enforce the **TRAFFIC SFP** on **incoming data packages or frames based on incoming interface and when using IPSec or MACSec on the destination address of the package** and all operations that cause that information to flow to and from subjects covered by the SFP.

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

Application note: The TRAFFIC SFP shall ensure that all traffic from the internal network will only be passed to the VPN channel and that all traffic from the external network will be rejected unless it is traffic coming from the VPN channel and in this case it will be passed to the internal network.

When using IPSec routed configuration, the TOE acts a router. It will forward the incoming internal packets to the appropriate remote network. The necessary routing information has to be supplied by the administrator in the configuration file.

When using MACSec or IPSec bridged configuration, the TOE acts as a layer 2 hub. It will forward all incoming internal Ethernet frames to the remote network.

6.1.4.4 FDP_IFF.1 – Simple security attributes

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

subjects

- **interface (clear interface or crypto interface)**

objects

- **IP packet (for IPSec routed) or Ethernet frame (for MACSec and IPSec bridged)**

security attributes

- **incoming interface for the package or frame**
- **destination address for the package or frame**

- **traffic type (SKUT traffic, ICMPv4 Fragmentation Required, ICMPv4 Echo Request, ICMPv4 Echo Reply or any other traffic)**

operations

- **pass or reject the package or frame**

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

- **incoming package or frame arriving on the clear interface will be passed to the appropriate VPN channel, when using IPSec routed based on the destination address**
- **incoming package or frame arriving on the crypto interface inside a VPN channel will be passed to the internal network.**

FDP_IFF.1.3 The TSF shall enforce the **no additional information flow control SFP rules**.

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

FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: **data packages or frames arriving at the crypto interface but not inside any VPN channel, except for the following packets when using IPSec:**

-
- **ICMPv4 Fragmentation Required**
- **ICMPv4 Echo Request**
- **ICMPv4 Echo Reply**

6.1.4.5 FDP_RIP.1 – Subset residual information protection

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the **deallocation of the resource from** the following objects: **cryptographic keys**.

Application note: As soon as a cryptographic key is not needed any more, the TOE overwrites the memory area.

6.1.5 Class FIA – Identification and Authentication

6.1.5.1 FIA_ATD.1 – User Attribute Definition

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

- **identity of the remote administrator or operator in form of the CN, UID or Email record of the client certificate**
- **association of the remote administrator or operator with a TLS client certificate.**

Application note: Only remote administrators and operators are know to the TOE. Local administrators are not individually identified by the TOE, but are identified and

authorized through organizational means. All certificates have to be signed by a trusted root CA. This root certificate is internal or provided on the USB memory or smart card.

6.1.5.2 FIA_UAU.2 – User Authentication before any Action

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

Application note: Only the remote administrators and operators are subject to any authentication by the TOE. Authentication is performed by verifying that the administrator or operator possesses the private key part to the TLS client certificate. Local administrators are authenticated through organizational means, by only allowing authorized personnel physical access to the TOE.

6.1.5.3 FIA_UID.2 – User Identification before any Action

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.

Application note: Identification of remote administrators and operators is performed by presenting a TLS client certificate.

6.1.6 Class FMT – Security Management

6.1.6.1 FMT_MOF.1 – Management of security functions behaviour

FMT_MOF.1.1 The TSF shall restrict the ability to **modify the behaviour of** the functions **listed below** to **an administrator**:

-
- **change the configuration of the TOE**

Application note: The TOE software update must only be possible after the authenticity of the firmware update has been verified (using the services and the trust anchor of the TOE) and if the version number of the new firmware is higher or equal to the version of the installed firmware. A TOE software update that has not been evaluated and certified will not be covered by the certification of the current software.

6.1.6.2 FMT_MSA.1 – Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the **ADMINISTRATOR ACCESS SFP** to restrict the ability to **modify** the security attributes **consisting of possible configuration options** to **administrators**.

6.1.6.3 FMT_MSA.3 – Static attribute initialisation

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

FMT_MSA.3.2 The TSF shall allow the **administrator** to specify alternative initial values to override the default values when an object or information is created.

Application note: An administrator can restrict unauthenticated access and specify security relevant initial values by changing the rules in the configuration file.

6.1.6.4 FMT_MTD.1a – Management of TSF data (administrator)

FMT_MTD.1.1 The TSF shall restrict the ability to **query or modify** the **TSF data listed below** to **an administrator:**

- **audit data [query]**
- **status of the TOE [query]**
- **status of the VPN-tunnels [query]**
- **configuration files [query, modify].**

Application note: The configuration files do not include the cryptographic key and seed file on the USB storage or smart card. Status of the TOE includes the status of the interfaces, firmware versions and other information that are security relevant for the administrator.

6.1.6.5 FMT_MTD.1b – Management of TSF data (operator)

FMT_MTD.1.1 The TSF shall restrict the ability to **query** the **TSF data listed below** to **an operator:**

- **audit data [query]**
- **status of the TOE [query]**
- **status of the VPN-tunnels [query]**
- **configuration files [query].**

Application note: The configuration files do not include the cryptographic key and seed file on the USB storage or smart card. Status of the TOE includes the status of the interfaces, firmware versions and other information that are security relevant for the administrator.

6.1.6.6 FMT_SMF.1 – Specification of management functions

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

- **query the software version**
- **query interface status**
- **query interface statistics**
- **query tunnel status**
- **apply changes to the configuration file**
- **reboot the TOE**
- **initiate update of the TOE software**
- **query the audit files**

Application note: The security management functions related to changes of the configuration of the TOE are described in more detail in FMT_MOF.1.

6.1.6.7 FMT_SMR.1 – Security roles

FMT_SMR.1.1 The TSF shall maintain the roles:

- **administrator**
- **operator**

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

Application note: The administrator is verified by the TOE for remote administrator

login. When the administrator is local the TOE environment ensures that only administrators have local access to the TOE (OE.PHYSEC).

6.1.7 Class FPT – Protection of the TOE Security Functions

6.1.7.1 FPT_FLS.1 – Failure with preservation of secure state

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

- **invalid configuration file**
- **removing of the USB memory or smart card**
- **failed integrity verification**

Application note: If the new configuration file is unreadable or does not conform with the syntax as described in the user guidance, then the previous configuration file will be kept in use. The syntax of the configuration file has been designed to prevent insecure configurations. In other failure cases the TOE shuts down.

6.1.7.2 FPT_STM.1 – Reliable time stamps

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

Application note: The TOE relies on its environment (i.e. the operating system) for reliable time stamps. The operating system can be configured to utilize the ntpd daemon to synchronize with another external system.

6.1.7.3 FPT_TST.1 – TSF testing

FPT_TST.1.1 The TSF shall run a suite of self tests **during initial start-up** 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 **none**.

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

Application note: During startup the firmware image is verified, but not the configuration data. The mcp daemon performs integrity check verification for all control and data plane modules before they are started. In addition, the data plane performs a verification check of its encryption engine at startup, using reference test vectors [RFC3686].

6.1.7.4 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1 The TSF shall provide administrators the ability to query the current version of the TOE software.

FPT_TUD_EXT.1.2 The TSF shall provide a mechanisms that **on a regular basis initiates and gives administrators the ability to initiate** updates to TOE software.

FPT_TUD_EXT.1.3 The TSF shall provide a means to verify software updates to the TOE using a **digital signature mechanism** prior to installing those updates.

FPT_TUD_EXT.1.4 The TSF shall provide a means to verify software updates to the TOE to ensure that software update version is newer than the current version of the TOE prior to installing those updates.

Application note: The trusted update is both an automatic mechanisms as well as administrator controlled. Apart from this the TOE can be updated outside of the control of the TOE by authorized persons that have physical access to the TOE (relying on the OE.PHYSEC). The activation and deactivation of the automatic update mechanism is part of the configuration changes made to the configuration file described in FMT_MOF.1 and FMT_SMF.1.

6.1.8 Class FTP – Trusted path/channels

6.1.8.1 FTP_ITC.1 Inter-TSF Trusted Channel (IPSec/MACSec)

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

FTP_ITC.1.2 The TSF shall permit **the TSF or a remote trusted IT product** to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for **VPN services.**

Application note: This channel is the VPN communication channel (IPSec/MACSec) that the TOE may establish with other Färists. Note that the communication channel can be established either by the TOE or by the remote Färist (remote end of the VPN).

6.2 Security Functional Requirements Rationale

6.2.1 Coverage

The following table provides a mapping of SFR to the security objectives, showing that each security functional requirement addresses at least one security objective.

SFR	Security Objectives
FAU_GEN.1	O.AUDIT
FCS_CKM.1	O.REMOTE, O.CHANNEL
FCS_CKM.2a	O.REMOTE, O.CHANNEL
FCS_CKM.2b	O.REMOTE, O.CHANNEL
FCS_COP.1a	O.REMOTE, O.CHANNEL
FCS_COP.1b	O.REMOTE, O.CHANNEL
FCS_COP.1c	O.CHANNEL
FCS_COP.1d	O.REMOTE, O.CHANNEL
FCS_COP.1e	O.UPDATE
FDP_ACC.2	O.LIMEXT
FDP_ACF.1	O.LIMEXT, O.REMOTE
FDP_IFC.2	O.MEDIATE
FDP_IFF.1	O.MEDIATE, O.SELPRO
FDP_RIP.1	O.SELPRO
FIA_ATD.1	O.REMOTE
FIA_UAU.2	O.REMOTE, O.SELPRO

SFR	Security Objectives
FIA_UID.2	O.REMOTE, O.SELPRO
FMT_MOF.1	O.CONFIG, O.LIMEXT, O.SELPRO
FMT_MSA.1	O.LIMEXT, O.SELPRO
FMT_MSA.3	O.CONFIG
FMT_MTD.1a	O.CONFIG, O.SELPRO
FMT_MTD.1b	O.CONFIG, O.SELPRO
FMT_SMF.1	O.CONFIG
FMT_SMR.1	O.REMOTE
FPT_FLS.1	O.SECSTA, O.SELPRO
FPT_STM.1	O.AUDIT
FPT_TST.1	O.SECSTA, O.SELPRO
FPT_TUD_EXT.1	O.UPDATE
FTP_ITC.1	O.CHANNEL

Table 6: Security functional requirement coverage

6.2.2 Sufficiency

The following rationale provides justification for each security objective for the TOE, showing that the security functional requirements are suitable to meet and achieve the security objectives.

Security Objective	Rationale
O.MEDIATE	The objective: • The TOE must mediate the flow of all information flowing between the TOE's clear and crypto interfaces and ensure that all traffic from the internal network is passed though the trusted channel and ensure that only traffic coming from the channel is passed to the internal network. is met by: • FDP_IFC.2, which enforces the TRAFFIC SFP on incoming traffic. • FDP_IFF.1, which defines rules for the TRAFFIC SFP.
O.CHANNEL	The objective: • The TOE must be able to provide trusted channels to remote trusted networks and protect information transmitted to and received from such networks against unauthorised disclosure and to detect any modification of incoming information transmitted from such networks, and to provide the means for the remote network to verify the integrity of information transmitted out of the TOE to such networks. is met by: • FCS_CKM.1, which specifies the cryptographic key generation for SKUT • FCS_CKM.2a, which specifies the certificate handling for SKUT • FCS_CKM.2b, which specifies the cryptographic key distribution for SKUT • FCS_COP.1a, which specifies RSA for SKUT • FCS_COP.1b, which specifies AES-256 for SKUT • FCS_COP.1c, which specifies AES-256 for IPSec and

Security Objective	Rationale
	MACSec • FCS_COP.1d, which specifies SHA-1 for SKUT • FTP_ITC.1, which mandates a dedicated encrypted communication channel
O.AUDIT	The objective: • The TOE must be able to provide audit evidence of security relevant events as well as for authorised use of security functions to allow an authorised administrator to read the audit trail. is met by: • FAU_GEN.1, which specifies the list of audit events • FPT_STM.1, which mandates reliable time stamps
O.CONFIG	The objective: • The TOE must provide the means for an authorized administrator to configure and manage the TOE security functions. is met by: • FMT_MOF.1, which specifies the list of management actions • FMT_MSA.3, which allows for different initial values • FMT_MTD.1a, which lists the management functions for administrators • FMT_MTD.1b, which lists the management functions for operators • FMT_SMF.1, which lists the TSF security management functions
O.LIMEXT	The objective: • The TOE must restrict the means to configuration and control of the TOE to authorised administrators. is met by: • FDP_ACC.2, which mandates the ADMINISTRATOR ACCESS SFP • FDP_ACF.1, which specifies the identification of administrators and operators • FMT_MOF.1, which limits the management actions to administrators • FMT_MSA.1, which limits the configuration options to authorised administrators
O.REMOTE	The objective: • The TOE must uniquely identify and authenticate the identity of all administrators and provide them with a secure communication channel before allowing remote administrators any access to the TOE. is met by: • FCS_CKM.1, which specifies the cryptographic key generation for remote administration • FCS_CKM.2a, which specifies the certificate handling for remote administration • FCS_CKM.2b, which specifies the cryptographic key distribution for remote administration • FCS_COP.1a, which specifies RSA for remote administration • FCS_COP.1b, which specifies AES-256 for remote administration • FCS_COP.1d, which specifies SHA-1 for remote administration

Security Objective	Rationale
	• FDP_ACF.1, which specifies the authentication for remote administration • FIA_ATD.1, which specifies the identity attributes for remote administration authentication • FIA_UAU.2, which mandates authentication for remote administration • FIA_UID.2, which requires identification for remote administration • FMT_SMR.1, which specifies the user roles known to the TSF
O.SECSTA	The objective: • Upon initial start-up of the TOE or during configuration, the TOE shall provide well-defined initial settings for security relevant functions. is met by: • FPT_FLS.1, which specifies how the TSF preserves a secure state in case of a failure (e.g., incorrect configuration) • FPT_TST.1, which specifies the self tests for the TSF to ensure correct operation of the TOE
O.SELPRO	The objective: • The TOE must protect itself against attempts by attackers to bypass, deactivate or tamper with TOE security functions. is met by: • FDP_IFF.1, which rejects traffic from the external network outside the trusted channel • FDP_RIP.1, which specifies that cryptographic keys are deallocated after use • FIA_UAU.2, which requires users to be authenticated before any TSF-mediated action is allowed • FIA_UID.2, which requires users to be identified before any TSF-mediated action is allowed • FIA_MOF.1, which restricts configuration changes and restarts to administrators • FMT_MSA.1, which limits configuration changes to authorised administrators • FMT_MTD.1a, which limits configuration changes to authorised administrators • FMT_MTD.1b, which limits operators to read-only actions • FPT_FLS.1, which specifies how the TSF preserves a secure state in case of a failure • FPT_TST.1, which specifies the self tests of the TSF to ensure correct operation of the TOE.
O.UPDATE	The objective: • The TOE must only accept updates that are newer than the current running version and updates where the origin and integrity can be trusted. is met by: • FPT_TUD_EXT.1, which specifies trusted updates are verified and accepted. • FCS_COP.1e provides the cryptographic operation for verification of signatures of the software image.

Table 7: Security functional requirement sufficiency

6.2.3 Security Requirements Dependency Analysis

Dependencies within the EAL package selected (EAL5) for the security assurance requirements have been considered by the authors of CC Part 3 and are not analysed here again. The augmentation by flaw remediation, ALC_FLR.1, has no dependencies on other requirements. The security functional requirements in this Security Target do not introduce dependencies on any security assurance requirement; neither do the security assurance requirements in this Security Target introduce dependencies on any security functional requirement.

The following table demonstrates the dependencies of SFRs modelled in CC Part 2 and how the SFRs for the TOE resolve those dependencies.

SFR	Dependencies	Resolution
FAU_GEN.1	FPT_STM.1	Yes, by FPT_STM.1
FCS_CKM.1	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	Yes, by FCS_CKM.2a, b Yes, by FCS_COP.1a, b, c Yes, by FDP_RIP.1
FCS_CKM.2a (cert)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	No, but covered by A.PHYSEC and A.NOEVIL
FCS_CKM.2b (keys)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	– – Yes, by FCS_CKM.1 adressed by the operational environment (FMSSL)
FCS_COP.1a (RSA)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	- - Yes, by FCS_CKM.1 adressed by the operational environment (FMSSL)
FCS_COP.1b (AES-CBC)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	- - Yes, by FCS_CKM.1 adressed by the operational environment (FMSSL)
FCS_COP.1c (AES-GCM)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	- - Yes, by FCS_CKM.1 adressed by the operational environment (FMSSL)
FCS_COP.1d (SHA)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	- - Yes, by FCS_CKM.1 adressed by the operational environment (FMSSL)
FCS_COP.1e (S/MIME)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	- - No, since the certificate is part of the an existing image. No, since the certificate is public.
FDP_ACC.2	FDP_ACF.1	Yes
FDP_ACF.1	FDP_ACC.1	Yes, by FDP_ACC.2

SFR	Dependencies	Resolution
	FMT_MSA.3	Yes
FDP_IFC.2	FDP_IFF.1	Yes
FDP_IFF.1	FDP_IFC.1 FMT_MSA.3	Yes, vy FDP_IFC.2 Yes
FDP_RIP.1	No dependencies	–
FIA_ATD.1	No dependencies	–
FIA_UAU.2	FIA_UID.1	Yes, by FIA_UID.2
FIA_UID.2	No dependencies	–
FMT_MOF.1	FMT_SMR.1 FMT_SMF.1	Yes and by A.PHYSEC and A.NOEVIL Yes
FMT_MSA.1	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	Yes, by FDP_ACC.2 - Yes Yes
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	Yes Yes
FMT_MTD.1a (administrator)	FMT_SMR.1 FMT_SMF.1	Yes Yes
FMT_MTD.1b (operator)	FMT_SMR.1 FMT_SMF.1	Yes Yes
FMT_SMF.1	No dependencies	–
FMT_SMR.1	FIA_UID.1	Yes, by FIA_UID.2
FPT_FLS.1	No dependencies	–
FPT_STM.1	No dependencies	–
FPT_TST.1	No dependencies	–
FPT_TUD.EXT.1	FCS_COP.1	Yes, by FCS_COP.1e
FTP_ITC.1	No dependencies	–

Table 8: Security functional requirements dependency analysis

6.3 Security Assurance Requirements

The assurance requirements are the EAL5 package augmented with ALC_FLR.1.

Assurance class	Assurance components
ADV: Development	ADV_ARC.1 – Security architecture description
	ADV_FSP.5 – Complete semi-formal functional specification with additional error information
	ADV_IMP.1 – Implementation representation of the TSF
	ADV_INT.2 – Well-structured internals
	ADV_TDS.4 – Semiformal modular design
AGD: Guidance documents	AGD_OPE.1 – Operational user guidance
	AGD_PRE.1 – Preparative procedures
ALC: Life-cycle support	ALC_CMC.4 – Production support, acceptance procedures and automation
	ALC_CMS.5 – Development tools CM coverage
	ALC_DEL.1 – Delivery procedures

	ALC_DVS.1 – Identification of security measures
	ALC_LCD.1 – Developer defined life-cycle model
	ALC_TAT.2 – Compliance with implementation standards
	ALC_FLR.1 – Basic flaw remediation
ASE: Security Target evaluation	ASE_CCL.1 – Conformance claims
	ASE_ECD.1 – Extended components definition
	ASE_INT.1 – ST introduction
	ASE_OBJ.2 – Security objectives
	ASE_REQ.2 – Derived security requirements
	ASE_SPD.1 – Security problem definition
	ASE_TSS.1 – TOE summary specification
ATE: Tests	ATE_COV.2 – Analysis of coverage
	ATE_DPT.3 – Testing: modular design
	ATE_FUN.1 – Functional testing
	ATE_IND.2 – Independent testing – sample
AVA: Vulnerability assessment	AVA_VAN.4 – Methodical vulnerability analysis

Table 9: Security assurance requirements

6.3.1 Security Assurance Requirements Rationale

The assurance level EAL5 augmented with ALC_FLR.1 has been chosen as appropriate for a network device (VPN endpoint) separating an internal classified network from an external non classified (public) network since it provides a moderate to high level of independently assured security, and a thorough investigation of the TOE. Since it can be expected that the mechanisms separating a classified network from from unclassified or public networks will be subject to moderate attack potential, choosing EAL5 makes the TOE capable of meeting this attack potential.

By choosing EAL5, a modular design is required. This will support the model of separating the base platform and management functionality from the VPN separation mechanisms. This will also support the model of adding additional separating mechanisms such as proxies without affecting the VPN functionality or the security of the underlying platform.

It is assumed that the TOE is operated in an environment where attackers have average expertise of the involved systems (e.g., general and publicly available knowledge on network protocols), limited resources and may have an average motivation because of possible high-value assets protected by the TOE. The overall attack potential is assumed to be moderate, which means that EAL5 is considered an appropriate assurance level, because it contains AVA_VAN.4 which ensures resistance against attackers with moderate attack potential. This level of assurance is needed in environments where the confidence in the ability of the TOE to provide a high degree of separation is necessary such as the separation of environments with different security classifications.

7 TOE Summary Specification

The TOE summary specification identifies the security functions that the TOE implements to meet the requirements defined in chapter 6 to the security target.

7.1 SF.PKTCLASS – Packet classification

Incoming packets on all interfaces are classified by the PktSwitch component. The classification is done differently depending on incoming interface and the packets are handled according to the TRAFFIC SFP.

It maps to the following SFRs:

- FDP_IFC.2
- FDP_IFF.1

7.1.1 Crypto interface

Incoming traffic on the crypto interface that is directed to the TOE and is encrypted (IPSec or MACSec packets) are sent to the "Packet Encryption" component. All other incoming traffic directed to the TOE is sent to the "Packet Filter". All incoming traffic that isn't directed to the TOE is dropped.

The "Packet filter" will only accept the following traffic:

- SKUT traffic
- ICMPv4 Fragmentation Required
- ICMPv4 Echo Request
- ICMPv4 Echo Reply

7.1.2 Clear interface

For MACSec and IPSec bridged all incoming traffic directed to the TOE itself is dropped. All other incoming traffic is sent to the "Traffic Filter" component and further sent to the "Packet Encryption" component.

For IPSec routed all incoming packets with an IP destination address that is configured to be sent to the tunnel is sent to the "Traffic Filter" component and further sent to the "Packet Encryption" component. Packets destined directly to the TOE is allowed if they match the following:

- ICMPv4 Fragmentation Required
- ICMPv4 Echo Request
- ICMPv4 Echo Reply

All other packets are dropped.

7.2 SF.VPN – VPN Functionality

The VPN functionality has two parts, key negotiation and bulk encryption.

Key negotiation is done by the skut component (skutd) using the SKUT protocol described in [SKUT3]. The skut component will generate encryption keys according to the TLS v1.0 protocol (FCS_CKM.1). The certificate and keys used in the key negotiation are RSA based (FCS_COP.1a, FCS_CKM.2a and FCS_CKM.2b). The certificate and private key is either in a file or in a smart card device. The SKUT protocol itself uses AES-256 (FCS_COP.1b) and SHA-1 (FCS_COP.1d).

Bulk encryption is done in the "packet encryption" component. The encryption uses either MACSec [IEEE802.1AE] or IPSec [RFC4303] with GCM-AES-256 (FCS_COP.1c).

This section also maps to the FTP_ITC.1 SFR.

7.3 SF.AUDIT – Security Audit

The central part of the security audit system is the syslog server component (Tlogd). It accepts audit records from other components over local sockets and sends the records to a remote log server over a TCP-connection. If the log server isn't available the log records are kept in RAM.

The syslog server component generates audit records for start-up and shutdown of the audit system as well as adding date and time of the record.

The following audit records are generated:

1. Connect and disconnect of a VPN-channel, generated by DPproxy
2. Renegotiation of the VPN key, generated by skutd
3. Operations performed by remote administrators and operators, generated by remadm
4. Login and logout of local administrators, generated by the login process
5. Changes to the configuration and keys, generated by configd

It maps to the FAU_GEN.1 SFR.

7.4 Security Management

The TOE offers administrators several interfaces to configure and manage the TSF. This includes the front panel, the USB interface and remote administration using the HTTPS interface.

The control panel and USB interface assume physical access to the TOE and are available to authorised administrators. Identification and authentication of administrators is only performed for remote administrators.

7.4.1 SF.REMADM – Remote administration

Remote administration is handled by the remadm component. All remote administrators and operators are authenticated using a TLS v1.0 encrypted and mutually authenticated connection to the remadm component. (FCS_CKM.1, FCS_CKM.2a and FCS_CKM.2b). The TLS_DHE_RSA_WITH_AES_256_CBC_SHA is the only supported TLS cipher suite. (FCS_COP.1a, FCS_COP.1b and FCS_COP.1d). The local TOE RSA certificate/private key is stored in a file on USB memory or on a smart card (FCS_CKM.2a and FCS_CKM.2b).

Users are authorised as either an administrator or an operator role. Each role has a list of CN, UID or Email addresses that when it matches a field in the client certificate will authorise the user for that role.

It is the remadm component that decides which actions are allowed for each role. The operator role can read status of the device, audit files and the configuration. The administrator role can also change the configuration.

The remadm component sends commands to the configd and statusd components to handle configuration and status control. Audit files are read directly by the remadm component.

This section maps also to the following SFRs:

- FDP_ACC.2
- FDP_ACF.1
- FIA_ATD.1
- FIA_UAU.2
- FIA_UID.2
- FMT_MSA.1
- FMT_MTD.1a
- FMT_MTD.1b
- FMT_SMR.1

7.4.2 SF.MGMT

Status and statistics of the TOE are managed through the statusd component. It interacts with other components to provide its services.

This section maps the following SFRs:

- FMT_MOF.1
- FMT_SMF.1

7.4.3 SF.CONF – Configuration

The TOE is configured through a configuration file in text format. It can be edited outside of the TOE by exporting it on a USB-memory or changed using the remote administration.

The syntax of the configuration file is described in the user guidance.

This section maps to the following SFRs:

- FMT_MSA.3
- FMT_SMF.1

7.5 TSF protection and support functions

7.5.1 SF.CRYPTO – Cryptographic Support

All cryptographic functions in the TOE is implemented using FMSSL, an encryption library approved by the Swedish NCSA.

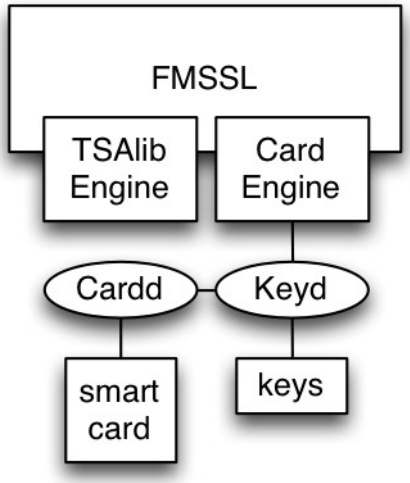

Illustration 5: FMSSL

The FMSSL version used is 2.3.

FMSSL implements:

- TLS v1.0 with the cipher suite TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- S/MIME

This maps to the following SFRs:

- FCS_CKM.1
- FCS_CKM.2a
- FCS_CKM.2b
- FCS_COP.1a
- FCS_COP.1b
- FCS_COP.1c
- FCS_COP.1d

7.5.2 SF.CONFVER – Configuration verification

The TOE verifies the configuration file during import. If the file is unreadable or contains syntactical errors, then it will be rejected and the previous configuration, if any, stays in place.

This section maps to the FPT_FLS.1.

7.5.3 SF.SELFTEST – Self-test

Self tests are run both at startup and at regular intervals.

The mcp component is the first component that is started after the kernel is bootstrapped. It checks the integrity of all system files before it starts any other services. Once the system is running the mcp will perform similar integrity checks of the system files at regular intervals.

The data plane component validates that all interfaces are available and performs a series of encryption tests. The encryption tests, which are based on test vectors from rfc

3686, verify the correct cryptographic operations of "packet encryption" component.

A test validating proper working of the "packet encryption" component is performed on per packet basis as well. For each packet undergoing cryptographic transformation, a part of the packet is encrypted and verified with a reference implementation of crypto algorithm.

This functionality implements the following SFRs:

- FPT_FLS.1
- FPT_TST.1

7.5.4 SF.FAILSAFE – Failsafe

When the USB-memory or smart card holding the RSA certificate/private key is removed the mcp component is signalled, which in turn restarts the system.

This functionality implements FTP_FLS.1.

7.5.5 SF.TIME – Time-stamps

The TOE provides reliable time stamps for its own use, in particular for the generation of audit records and validation of certificates used for VPN and administrator authentication. The time stamp is provided by the TOE environment through the NTP service of the underlying operating system.

This functionality implements FPT_STM.1.

7.5.6 SF.RIP – Residual information protection

Memory segments that are used for cryptographic keys are overwritten as soon as they are no longer needed.

This maps to FDP_RIP.1.

7.5.7 SF.AUTOUPD – Automatic update

The automatic update functionality is implemented using digitally singed updates that are verified using a factory installed certificate. The autoupdate daemon will verify the signature and verify that the software in the update is newer than the running version.

This functionality implements the following SFRs:

- FPT_TUD_EXT.1
- FCS_COP.1e

8 Abbreviations, Terminology and References

8.1 Abbreviations

AES	Advanced Encryption Standard
API	Application Programming Interface
CA	Certificate Authority
CC	Common Criteria
CN	Common Name
CPU	Central Processing Unit

CRL	Certificate Revocation List
DHCP	Dynamic Host Configuration Protocol
EAL	Evaluation Assurance Level
ESP	Encapsulating Security Payload
FIPS	Federal Information Processing Standard
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IP	Internet Protocol
IPSec	Internet Protocol Security
MACSec	Media Access Control Security
NTP	Network Time Protocol
OSI	Open Systems Interconnection
PKCS	Public-Key Cryptographic Standard
RAM	Random Access Memory
RFC	Request For Comment
RSA	Rivest, Shamir and Adleman
SHA	Secure Hash Algorithm
SFP	Security Function Policy
SKUT	Simple Key-exchange Using TLS
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functions
USB	Universal Serial Bus
VPN	Virtual Private Network

8.2 Terminology

8.3 References

CC	Information Technology – Security Techniques – Evaluation Criteria for IT Security, also known as the Common Criteria or CC – Common Criteria for Information Technology Security Evaluation. • Part 1: Introduction and general model, September 2012, Version 3.1 Revision 4, CCMB-2012-09-001 • Part 2: Security functional Components, September 2012, Version 3.1 Revision 4, CCMB-2012-09-002 • Part 3: Security Assurance Components, September 2012, Version 3.1 Revision 4, CCMB-2012-09-003
CEM	Common Methodology for Information Technology Security Evaluation,

	Evaluation Methodology, September 2012, Version 3.1 Revision 4, CCMB-2012-09-004
FIPS180-4	FIPS 180-4, Secure Hash Standard (SHS), 2012 March, FIPS PUB 180-4
FIPS197	Advanced Encryption Standard (AES), 26. November 2001, FIPS-197
IEEE802.1AE	Media Access Control (MAC) Security, IEEE Std 802.1AE-2006, IEEE Computer Society, 18 August 2006
IEEE802.1AEbn	Media Access Control (MAC) Security – Amendment 1: Galois Counter Mode – Advanced Encryption Standard – 256 (GCM-AES-256) Cipher Suite, IEEE Std 802.1AEbn-2011, IEEE Computer Society, 14. October 2011
PKCS1v2.1	PKCS #1 v2.1: RSA Cryptography Standard, RSA Laboratories, 14. June 2002
RFC2246	RFC 2246: The TLS Protocol Version 1.0, The Internet Society, January 1999
RFC4303	RFC 4303: IP Encapsulating Security Payload (ESP), The Internet Society, December 2005
RFC3686	RFC 3686: Using Advanced Encryption Standard (AES) Counter Mode With IPSec Encapsulation Security Payload (ESP)
RFC5280	RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, NIST, May 2008
SKUT3	Skut version 3 protocol description, v 1.0