# genugate firewall 11.0 p0
# Security Target

genua GmbH — Kirchheim

2026-01-27
Version 11.0.12 (141f1f2)

genua.

# Contents

# 1 ST Introduction

## 1.1 ST Reference

| | ST Reference |
|---|---|
| ST Title | genugate firewall 11.0 p0 Security Target |
| Version | 11.0.12 |
| Developer | genua GmbH |
| Date | 2026-01-27 |

## 1.2 TOE Reference

| | TOE Reference |
|---|---|
| TOE Name | genugate firewall 11.0 p0 |
| Product Name | genugate 11.0 p0 |

## 1.3 TOE Overview

The Target Of Evaluation (TOE) **genugate firewall 11.0 p0** is part of a larger product, the firewall **genugate 11.0 p0**, which consists of hardware and software. The TOE itself is part of the software. The operating system is a modified OpenBSD. The TOE supports IPv4 and IPv6.

To mitigate hardware failures the genugate has a high availability (HA) option where two or more genugate systems are operating in parallel and take over a failing system.

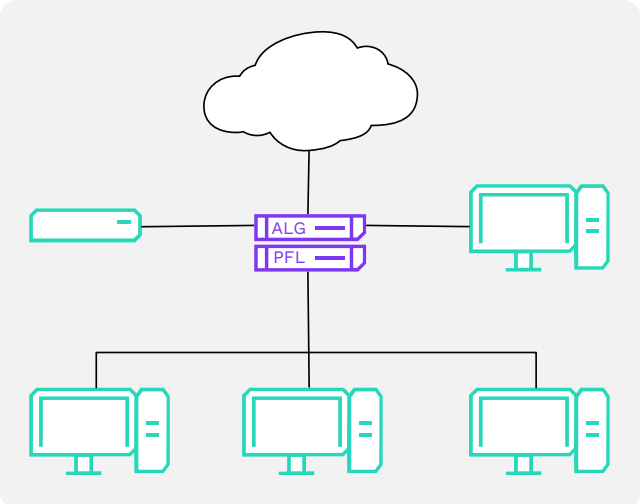



Figure 1: Overview hardware variant **genugate** (PFL not part of TOE)

The product genugate exists in two variants, shown in figures 1 and 2.

- The first simply named **genugate** is a two-tiered application with the physically separated components Application Level Gateway (ALG) and the Packet Filter (PFL) connected in series.

  They run on dedicated hardware provided by genua and other selected suppliers.

- The second variant named **genugate Virtual** is the virtualized ALG without the PFL.

  In virtualized environments, the serial connection of two virtualized components on one hypervisor does not bring any significant security advantage. The total system can only be as secure as the hypervisor.

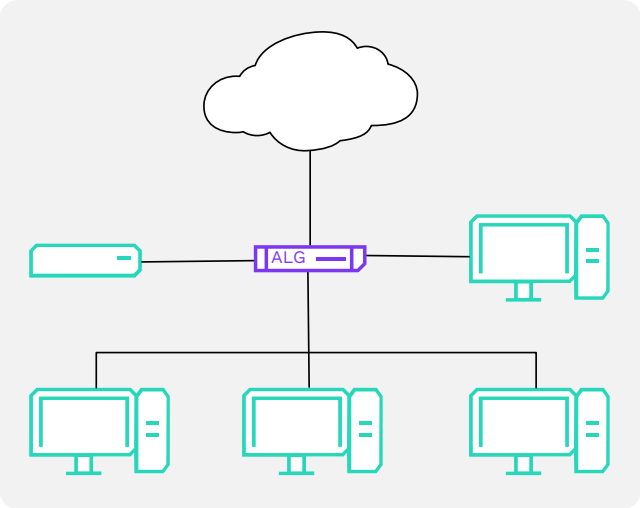



Figure 2: Overview variant **genugate Virtual**

For the genugate Virtual, it is possible to run several instances on one hypervisor (see figure 3). They can run in a HA setup or as independent standalone systems.

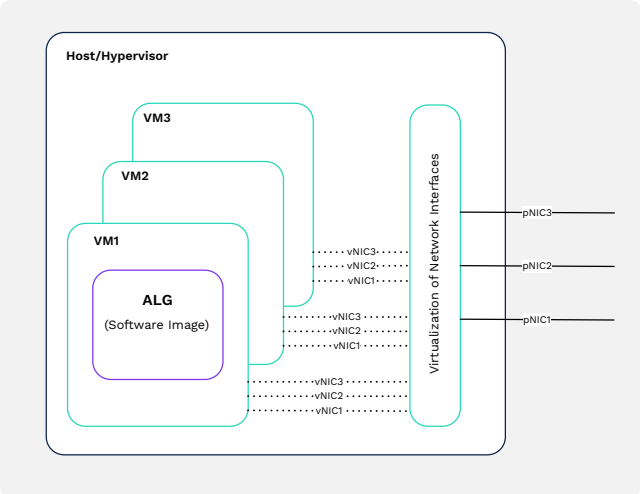



Figure 3: Virtualization setup for **genugate Virtual** with several genugate Virtual instances running on one hypervisor

Besides the network interface to the PFL (or the internal network in case of the genugate Virtual), the ALG has (at least) three more interfaces in order to connect to the external network, the administration network and the demilitarized zone(s) (DMZ). For the high availability option, the ALG needs another network interface for the HA network. The PFL has a second interface which is connected to the internal network, and optional interfaces for further DMZs.

The aim of the firewall is to control the IP traffic between the different connected networks. Therefore the ALG uses proxies that implement filter policies in order to control all data transmitted between the different networks, while the PFL uses packet filtering as an additional means to control all data that is sent to and from the internal network.

The TOE, **genugate firewall 11.0 p0**, consists of the software that implements the IP traffic

control and related functionality of the firewall on the ALG. This includes the proxies, the hardened OpenBSD kernel IP stack, packet filter, but also other supportive functionality as logging of security events (see the next section for a more detailed definition of the TOE scope and boundary).

The PFL software, although the PFL is present in the architecture of the hardware variant **genugate**, is not part of the TOE.

The genugate product family includes the following security features, provided by the TOE:

- The ALG does not perform IP forwarding, but it terminates all connections at the respective proxies.
- IP spoofing checks are performed. The source and destination address of the IP packet are checked against the IP address (and netmask) of the receiving interface.
- Proxies that accept connections from the connected networks run in a restricted runtime environment.
- The modified OpenBSD kernel logs events related to firewall security that occur while checking incoming IP packets, and it keeps statistic counters for other events.
- All central processes of the ALG are monitored by the system and kept running.
- The log files are continuously analyzed and the administrators are notified about security-relevant events.
- The deletion and/or modification of critical files is prohibited by the respective usage of OpenBSD file system flags.
- The TOE has a special maintenance mode. During normal operation IP packets are handled as usual and the file system is secured by the OpenBSD file system flags. In maintenance mode, however, the OpenBSD file system flags can be altered for maintenance operation. In this mode all IP packets are dropped for security reasons (single-user mode).
- To mitigate hardware failures the genugate has a high availability (HA) option where two or more genugate systems are operating in parallel and take over a failing system. The different systems synchronize their configuration with one another. The genugate provides certified mechanisms for both the physical and virtualized variants.

Moreover, as an additional layer of security hardening, for the physical genugate the internal network is protected by a two-tier security architecture that filters at different levels of the network stack. The two tiers (ALG and PFL) run on separated hardware that is separately configurated. The filter rules of the PFL cannot be modified during normal operation. Both ALG as well as PFL enforce packet filter rules.

### 1.3.1 Required non-TOE Hardware/Software/Firmware

The hardware product **genugate** is based on OpenBSD 7.5 that runs on a large scale of hardware using different INTEL-compatible processors. The ALG needs at minimum an Intel Celeron with 1 GB memory and four 1 GBit network interfaces (the high availability option needs at least five interfaces). The PFL needs an Intel Celeron with 512 MB memory and two 1 GBit network interfaces. Nonetheless the hardware is selected by the manufacturer in order to guarantee proper execution of the product.

The currently distributed hardware versions are the genugate S, genugate M and genugate L, revisions 3.0 and 4.0. These hardware versions, as well as the Infodas hardware *SDoT Server V3B*, are in scope for this certification.

The hardware versions use the CPUs listed in table 1.

The hardware revisions 1.0 and 2.0 are not in scope for this evaluation although the software runs on it with the same security functions.

Table 1: CPUs used for the ALG of the hardware genugate

| Hardware version | Revision | CPU |
|---|---|---|
| genugate S | Rev 3 | AMD EPYC 3151 |
| genugate S | Rev 4 | Intel Core i5-10500TE |
| genugate M | Rev 3 | Intel Xeon E-2236 |
| genugate M | Rev 4 | Intel Xeon E-2386G |
| genugate L | Rev 3 | Intel Xeon W-3225 |
| genugate L | Rev 4 | Intel Xeon W-3323, Intel Xeon W-3335 |
| SDoT Server V3B | | Intel Xeon E-2276ML |

Table 2: CPUs used for the ALG of the genugate Virtual

| Virtualiser | CPU |
|---|---|
| VMWare ESXI | AMD EPYC 7452 |
| KVM | AMD EPYC 73F3 |
| KVM | Intel Xeon Silver 4310 |

The proxies and other user space programs on the ALG are based on Perl 5.36.3 which is distributed with the product.

For the high availability option using OSPF a correctly configured OSPF router is needed in the internal network.

If the **genugate Virtual** is used, the server on which the genugate Virtual will be installed needs specific CPU and IO virtualization features. The virtualization servers must provide the same CPU, memory and network capacities as the ALG hardware model.

The following server CPUs and hypervisors are in the scope of the evaluation for the genugate Virtual:

- CPU: Intel or AMD, supporting the required virtualization features VT-x (for Intel CPUs) or AMD-V (for AMD CPUs)
- hypervisor: VMware ESXi or KVM

The CPUs used in the virtualisation hosts are listed in table 2.

All required firmware is delivered with the ALG installation image and need not be provided by the customer.

## 1.4 TOE Description

The TOE **genugate firewall 11.0 p0** is used to control the connections and data transfer between different networks, where each network has different security needs and different threat levels for the other networks.

The TOE can be configured in such a way that the security needs for each network are optimally met. A standard configuration consists of the following networks connected to the TOE:

**internal network:** This is the network that has to be protected against attacks from the external network. Usually only a few services from the internal network are accessible from the external network, secured by user authentication. The internal network is secured by the TOE, using filtering mechanisms at different levels of the IP stack. This network is usually controlled by a defined security policy.

**external network:** This is the most insecure network, e. g. the internet. In general, no security policy exists, and all kind of attacks can occur in this network.

**administrative network:** This network is used to allow a secure administration of the TOE. This network is isolated from all other networks and only administrators have access. The usual access is through the HTTPS web interface, but an SSH access for debugging and maintenance operation is also available.

**demilitarized zones:** These networks allow access to common services from the external network, without the need to open the internal network. Usually, Web servers and FTP servers are installed in this network. This network is usually controlled by a defined security policy.

**HA network:** This internal network is necessary for the high availability option. It is used to synchronize the configuration between the systems.

The internal HA network must use IPv4 addresses. For all other networks IPv4 and IPv6 is supported (with the exception that the proxy mcastudprelay, which is needed for multicast UDP packets, only supports IPv4).

The TOE provides the following security features:

- The ALG does not perform IP forwarding. Instead, all connections are terminated at the respective user-space proxies. For performance optimization, the ALG uses socket splicing for TCP connections and UDP datagrams when appropriate.
- IP spoofing checks are performed. The source and destination address of the IP packet are checked against the IP address (and netmask) of the receiving interface.
- Proxies that accept connections from the connected networks run in a restricted runtime environment.
- The modified OpenBSD kernel logs events related to firewall security that occur while checking incoming IP packets and keeps statistics for other events.
- The TOE has a special maintenance mode. During normal operation IP packets are handled as usual and the file system is secured by the OpenBSD file system flags. In maintenance mode, however, the OpenBSD file system flags can be altered for maintenance operation. In this mode all IP packets are dropped for security reasons.
- All central processes of the ALG are monitored by the system and kept running.
- The log files are continuously analyzed and the administrators are notified about security-relevant events.
- The log files are intelligently rotated (overwriting the oldest data first) so that they avoid filling the available space but the administrator still can see recent log entries. There are two classes of log files, the rotating and the flagged ones. The rotating log files are rotated automatically, based on size and time. The flagged log files can only be rotated in maintenance mode with the acknowledgement of the administrator.
- File configuration of the OpenBSD file system flags prohibit the deletion of the most important log messages, and the modification of all critical files of the system, in particular the modification of executable programs and scripts.
- To mitigate hardware failures the genugate has a high availability option where two or more genugate systems are operating in parallel and take over from a failing system. The different systems synchronize their configuration with one another.

For the physical genugate, the internal network is moreover protected by a two-tiered security architecture that filters on different levels of the network stack (ALG and PFL, running physically separated on different hardware). ALG and PFL must be separatedly configurated. The filter rules of the PFL cannot be modified during normal operation. All this provides an additional layer of security hardening.

The genugate Virtual consists of the virtualized ALG and implements equivalent filter tasks on IP level as the PFL does, which suits a virtualized environment.

### 1.4.1 The Application Level Gateway (ALG)

The ALG uses policies to provide and control connections between the different networks. The policies are realized by user-space proxies, called relays. The relays examine the data and perform most of the filtering and controlling function. The protocol-specific relays have enough knowledge about the respective protocol in order to filter possible threatening or insecure protocol elements. The relays implement several access control lists that allow a fine-grained control for the usage of services. All relays can be transparent with respect to the source and/or destination address, so that the ALG can be configured transparent with respect to IP addressing. The ALG checks for source or destination spoofing attacks.

The ALG does not perform IP forwarding. Therefore, for performance reasons and in cases where allowed, socket splicing is used to optimize the handling of TCP connections/UDP packets through the ALG: After the initial flow control checks on connection setup, the relays can switch to socket splicing mode. Then the data that would only be copied from kernel mode to application mode and back is kept in kernel memory. The connections are handled by the kernel like all traffic but instead of being copied to user space it is directly directed to the output socket. Socket splicing should be strictly distinguished from IP forwarding. Using IP forwarding, no packet reassembly is done; and all packets are copied verbatim to the outgoing socket including their IP headers, without further checks. With socket splicing, the TCP data stream/UDP contents is extracted out of the IP packets with all associated tests and checks and new IP packets are created by the kernel on output. Socket splicing is *not* applied for protocols where the whole data stream must be checked. So it is not feasible for protocols that use the virus scanner or that filter HTML data.

The generic relays for UDP and TCP can apply a protocol conformance filter (PCF) that matches the protocol data at the beginning of the connection against regular expressions. If the match fails, the relays finish the connection.

The administrator can also configure (meta-)policies which are implemented by one or more relays with predefined attributes for common use cases.

The TOE provides proxy support for the following services/policies implemented by the respective relays:

**IP:** This policy can be used for all IP protocols (besides ICMP ECHO, UDP, or TCP, which are supported by their own proxies). It is a very generic proxy and has no knowledge about any application level protocol.

The iprelay implements this policy.

**PING:** This policy is used if the ALG should transmit ICMP ECHO REQUEST and ICMP REPLY packets from one network into another.

The pingrelay implements this policy.

**UDP:** This policy is implemented by a generic proxy than can be used for almost any service that is based on UDP.

The udprelay implements this policy.

This policy knows the following PCF: DNS, MSSQL.

**TCP:** This policy is implemented by a generic proxy that can be used for services based on TCP. It has no knowledge about application level protocols unless protocol conformance filters are configured that check for a basic protocol conformance by applying regular expressions at the beginning of the communication. It can handle TLS connections.

The tcprelay implements this policy.

This policy knows the following PCF: BGP_v4, DNS, Fernwartungs_App, IMAP_v4, LDAP, MSSQL, MySQL, POP3, PostgreSQL, PostgreSQL_SSL, PPTP, RDP, SMB, SSH, SSH_v2, SSL, SSL_no_v3, TeamViewer, VNC.

**SMTP:** This policy is implemented by an application-specific proxy for the SMTP protocol. All protocol commands are analyzed and can be filtered. The mail header and bodies can be filtered. It contains functionality to filter SPAM mail. It has an interface to an optional virus scanner. SMTP authentication can optionally be configured.

The smtprelay implements this policy.

**SMTP2SMTP:** This policy is implemented by an application-specific proxy for the SMTP protocol. All protocol commands are analyzed and can be filtered. The mail header and bodies can be filtered. It contains functionality to filter SPAM mail. It has an interface to an optional virus scanner. The SMTP2SMTP relay does not authenticate the users itself, but relies on the responses of the remote Mail Transfer Agent (MTA). In contrast to the SMTP relay the SMTP2SMTP relay does not queue the mails to `postfix`, but directly connects to the SMTP server.

The smtp2smtprelay implements this policy.

**IMAP:** This policy is implemented by an application-specific proxy for the IMAP and IMAPS (meta-policy) protocols. All protocol commands are analyzed and can be filtered. It has an interface to an optional virus scanner.

The imaprelay implements this policy.

**POP:** This policy is implemented by an application-specific proxy for the POP and POPS (meta-policy) protocols. All protocol commands are analyzed and can be filtered. It has an interface to an optional virus scanner.

The poprelay implements this policy.

**SIP:** This policy is implemented by an application-specific proxy for the SIP and SIPS (meta-policy) protocols. All protocol commands are analyzed and can be filtered.

The siprelay implements this policy.

**SSH:** This policy is implemented by an application-specific proxy for the SSH protocol. It intercepts SSH connections, can filter selected SSH protocol messages and can authenticate users.

The sshrelay implements this policy.

**WWW:** This policy aims to protect clients when they try to connect to external web servers, i. e. while doing "web surfing". The policy is implemented by an application-specific proxy for the HTTP protocol and its application data. This proxy analyzes the HTTP protocol headers and the application data. The content type of the application data can be used to either filter text data like HTML or to scan binary data for viruses. It can handle TLS connections.

The wwwrelay implements this policy. This relay can also filter request methods.

The **WWWSERVER** meta-policy works similarly like the WWW policy, but it is a meta-policy that specifically aims to protect the own web servers towards requests from the external network. It is also implemented by the wwwrelay.

**HTTP:** This policy is implemented by an application-specific proxy for the Websocket and/or the HTTP protocol and its application data. This proxy analyzes the HTTP protocol headers and the application data. It cannot be configured directly but only via meta-policies.

The httprelay implements this policy.

The **WEBSERVICE** meta-policy can control SOAP and WSDL services by validation against XML schema files that are uploaded onto the genugate. It can handle TLS connections.

The **OPCUAHTTP** meta-policy validates OPC UA messages.

**FTP:** This policy is implemented by an application-specific proxy for the FTP protocol. All protocol commands are analyzed and can be filtered. and users can be authenticated. It can handle TLS connections. It has an interface to an optional virus scanner.

The ftprelay implements this policy.

**MCASTUDP:** This policy is implemented by a generic proxy for UDP multicast packets using IPv4. It filters IGMP packets based on the multicast group and allows or blocks multicast UDP packets according to the current group membership. The relay needs support from the `igmpproxy` at the PFL which is needed to properly route the multicast UDP packets on the PFL.

The mcastudprelay implements this policy.

**Further Meta-Policies:** Besides the already mentioned meta-policies there are further meta-policies that use the low-level TCP, UDP and IP policies. These are BGP, DNS, DNSSERVER, IMAPFILTER, IPSEC, LDAP, MSSQL, MYSQL, NTPSERVER, OPCUATCP, POSTGRESQL, PPTP, RDB, SMB, SNMPSERVER, SNMPTRAP, TEAMVIEWER, VNC, and WAF. These meta-policies are combinations of different policies preconfigured for the respective service.

Depending on the policy, the tcprelay, udprelay and/or iprelay implement the policy.

All relays are highly configurable. The preferred configuration method is through HTML forms at the administrative interface (Web-GUI) that are transported by secure HTTPS connections in the administration network. An alternative is a REST-oriented interface (REST-API) that can also be used by automated procedures.

Some relays have support for authentication in the respective protocol. These relays can authenticate their users against authentication servers (LDAP or RADIUS).

Unencrypted SNMP monitoring should only be made from sufficiently secure networks, because the SNMP packets may contain sensitive information. An alternative is strongly encrypted SNMPv3.

The ALG has an interfaces to communicate with a virus scanner (either local or remote). It also has an interface to a web site classification service (Advanced Web Categories, AWC). (The virus scanner or the classification service itself are not part of the TOE.)

### 1.4.2 The Packet Filter (PFL)

The internal network has high security needs and, for the physical genugate, is therefore not directly connected to the ALG, but to the PFL. The PFL has at least two network interfaces: One of them is connected to the ALG with a cross cable. The (small) network is called the transit network. The other interface connects to the internal network.

The PFL works as packet filter with a set of filter rules. Only configured TCP connection requests from the transit network are allowed, but there is no default restriction for packets from the internal network. In order to allow connections into the internal network, extra rules have to be added by administrators.

The PFL is a minimalistic system. In the certified mode it boots from a removable USB stick and has no other permanent memory. The medium is configured and created at the ALG. Physical access is needed to write the medium at the ALG, transfer it from the ALG to the PFL, and reboot the PFL with the new configuration.

The configuration of the PFL is done through the administrative interface at the ALG.

### 1.4.3 High Availability (genugate cluster)

For a high availability (HA) setup, the HA option is installed on two or more genugates (HA peers) and they are connected by a separate HA network that is used to synchronize the configuration and negotiate the active HA nodes. If a system fails some other system takes over its services and IP addresses.

For the variant using OSPF an external OSPF router is needed in the internal network. Figures 4 and 5 give an overview for two parallel systems, although more than two are possible.

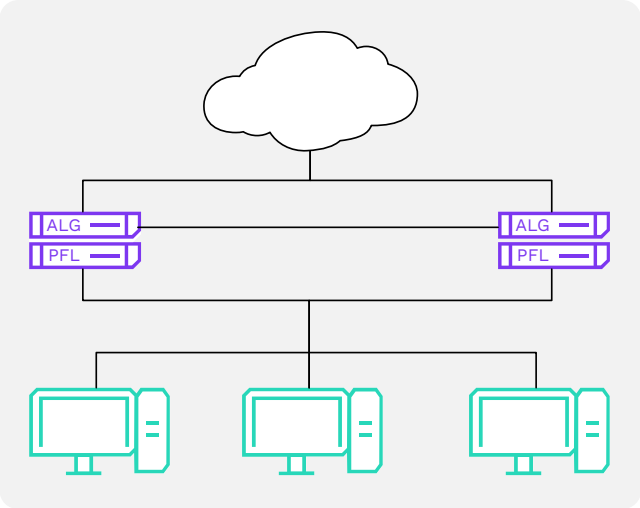



Figure 4: Overview HA setup for hardware variant **genugate**

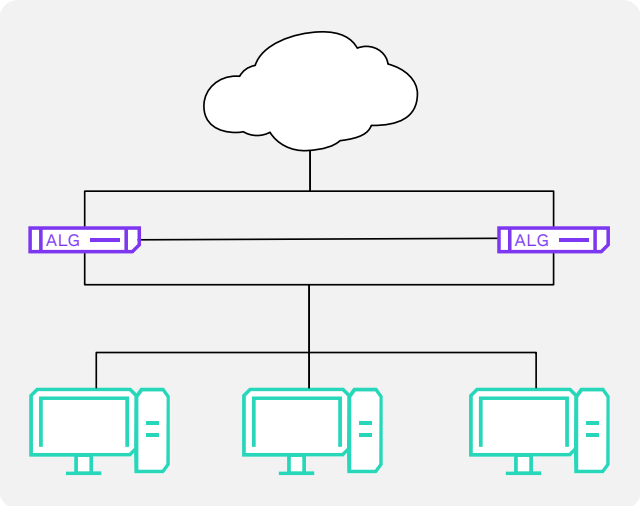



Figure 5: Overview HA setup for variant **genugate Virtual**

The synchronization of the configuration in the HA network uses IPsec using IKEv2 with preshared keys to encrypt the communication and to ensure integrity and authenticity.

Optionally the transit networks of the genugate peers can be united into one transit network. Then cross cables can no longer be used and switches must be incorporated. This setup avoids a full HA take over if only one PFL fails.

The CARP setup can also be used in a PAP configuration where an additional packet filter is placed before the ALG. The CARP PAP configuration is not part of this certification.

### 1.4.4 Physical Scope

Both ALG and PFL run on Intel-compatible hardware in 64 bit mode (architecture x86_64). The hardware components are selected by genua.

For the physical variant **genugate** the end user has no need to check for compatibility. The

Table 3: Scope of delivery. The row *Hardware* only applies to the hardware variant **genugate**.

| **Type** | **Name** | **Release** | **Hardware** | **Medium** |
|---|---|---|---|---|
| Hardware | genugate | 11.0 p0 | Server hardware:<br>• genugate S, genugate M, or genugate L, with hardware revision 3.0 or 4.0;<br>• *SDoT Server V3B*[1] | hardware (no software pre-installed) |
| Software | genugate | 11.0 p0 | N/A | installation image (see text for details) |
| Documentation | English and German manuals (only German versions are part of TOE):<br>• administrator and user guidance (genugate Administration and Configuration Manual, genugate Administrations- und Konfigurationshandbuch);<br>• installation manual (for hardware variant: genugate Installation Manual, genugate Installationshandbuch; for virtual variant: genugate Virtual Installation Manual, genugate Virtual Installationshandbuch);<br>• certification guidance (genugate Certification: General Information, genugate Erläuterungen zur Zertifizierung) | 11.0 p0 | N/A | manuals, downloaded separately from genua customer support server |

scope of delivery can be seen in table 3. For the **genugate Virtual** the end user has to check the hardware if the needed virtualization features (VT-x for Intel CPUs and AMD-V for AMD CPUs) are supported, see chapter 1.3.1 for the requirements on hardware and hypervisor in the scope of the evaluation. In both cases, for the physical as well as for the virtual variant of the genugate, the hardware is not part of the TOE.

The genugate software is distributed in form of an installation image (ISO image, USB image or

[1]As an exception, *SDoT Server V3B* is not delivered by genua: The *SDoT Server V3B* hardware is an Infodas product and is supplied and delivered by Infodas. If the genugate hardware variant with Infodas hardware *SDoT Server V3B* shall be used, the end customer must order and purchase the hardware *SDoT Server V3B* separately from company Infodas.

QCOW2 image format). This installation image can either be shipped together with the hardware on a separate USB flash drive, or it can be downloaded from the genua support server, whereby the second option is the only one possible for customers buying the genugate Virtual product variant. The documentation is contained in the installation image, but it can also be downloaded separately from the genua support server. The TOE as part of the software is contained in the installation image for the product genugate 11.0 p0.

**Application Note**: While the evaluation was performed only with genugate hardware of revision 4.0, 3.0 and *SDoT Server V3B*, the software is expected to run with all security features also on older still supported and newer hardware revisions, provided the hardware requirements of the preceding paragraph are met.

The physical connections for the hardware variants are:

- the network interfaces to the external, internal, HA (optionally), administration networks and the DMZ
- connections for the keyboard, monitor, and serial interfaces at the ALG and PFL
- power supply

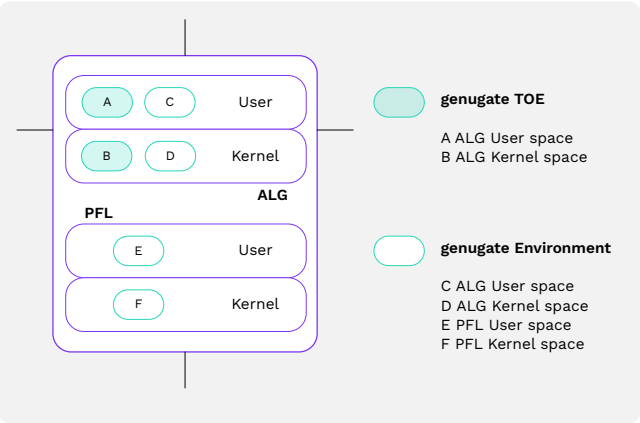



Figure 6: Scope and boundary for the hardware variant **genugate**

The following list gives a schematic overview on the TOE and its environment. It divides the software on ALG and PFL into user and kernel space parts. On ALG, the user and the kernel space contain parts of the TOE, and parts of the environment, whereas on PFL, user as well as kernel space belong to the environment. The following table lists the components in each part. The components for the parts **A** and **B** are part of the TOE. The components for **C**, **D**, **E**, and **F** are part of the environment.

Schematic overviews for both genugate variants are shown in figures 6 and 7

- **A** ALG TOE User space
  relays, logging, admin web server, configuration commands, system startup
- **B** ALG TOE Kernel space
  network layer, logging, system call interface
- **C** ALG Environment User space
  `squid`, `postfix`, DNS server, `ntpd`, SNMP server, CARP PAP configuration, genugate options: AWC, virus scanner; authentication methods, OS environment

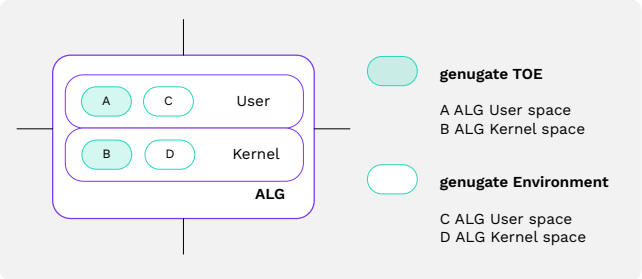



Figure 7: Scope and boundary for variant **genugate Virtual**

- **D** ALG Environment Kernel space
  process management, memory management, device drivers, socket layer, tty driver, I/O system, IPC operation, file systems
- **E** PFL Environment User space
  logging, system startup `igmpproxy`, `ospfd`, `ospf6d`, OS environment
- **F** PFL Environment Kernel space
  network layer, logging, system call interface process management, memory management, device drivers, socket layer, tty driver, I/O system, IPC operation, file systems

The different parts have the following interfaces with one another:

| | | |
|---|---|---|
| **A** | **B** | system call interface |
| **A** | **C** | interprocess communication (via system call interface) |
| **B** | **D** | kernel interfaces between the kernel components |
| **E** | **F** | system call interface |
| **ALG** | **PFL** | serial connection |
| **ALG** | **PFL** | network connection |
| **ALG** | **PFL** | USB boot medium |

### 1.4.5 Logical Scope

The genugate is a complex product with many options that are not directly connected to the TSF. Some of these options do not interfere with the security functions of the TOE and are assumed to be configurable and usable in certified mode.

On the other hand there are options that interfere with the security functions and therefore must not be used in certified mode.

The following paragraphs describe the features that are in scope of the TOE and for which SFRs exist. The list of excluded options consist of those options that interfere with the secure operation of the TOE. These options must not be configured.

The TOE has the following logical scope:

- The kernel components “network”, “packet filter”, and “restricted runtime” for the ALG. These components perform the spoofing checks, packet filtering and access control for incoming data. The spoofing checks contain detecting any mismatch between the source and destination address of the IP packet and the IP address and netmask of the receiving interface.
- All relays that implement the policies and meta-policies referenced in chapter 1.4.1. These components perform the filtering on application level, ACL checks, and calls to the optional virus scanner. The virus scanning functionality is not part of the TOE. The sshrelay, ftprelay,

wwwrelay and smtprelay allow for user authentication. For the smtprelay the authentication at the relay is optional.[2] The authentication methods themselves are not part of the TOE. Moreover, there is no security claim for the OPC UA functionality.

- The TCP and UDP relays can filter protocol conformance by applying regular expressions at the beginning of the communication. There are several predefined protocol conformance filter.
- System startup. This component performs the secure startup of the system and the conversion to maintenance mode.
- The logging and self-monitoring tools. These components perform the accounting and auditing functions.
- Admin web interface including the REST web interface (REST-API). This component allows the configuration by administrators.
- The configuration for the users, network, relays, DNS server, mail server, packet filter, Squid web proxy, virus scanner, AWC, audit, SNMP server, and igmpproxy.
- The option HA.
- A patch installation mechanism, that only installs software patches signed by genua. The mechanism also checks that the patch is appropriate for the given software version and patch level.

  The public key for the signature verification is contained in the installation image and part of the secure installation process. During operation it is secured by the self protection measures.

The TOE has the following logical boundaries:

- virus scanner interface: delivering the data to the virus scanner and obtaining the scanner result. The virus scanner itself is not part of the TOE.
- Advanced Web Categories (AWC): interface to a web site classification service. The classification service itself is not part of the TOE.
- Web Application Firewall (WAF): interface to a local web site protection service using the Modsecurity Project of the OWASP foundation.
- external authentication methods: interaction with the authentication service. The authentication methods themselves are not part of the TOE.
- configuration interface: sending forms to and receiving form data from a web browser. The JavaScript code delivered from the genugate to the administrative web browser for easy navigation within the browser are not part of the TOE (e.g. *swagger-ui*).

This Security Target claims SFRs for cryptographic operations only for some parts:

- integrity and authenticity check on software patch installation
- HTTPS access incl. TLS to the administrative interface
- SSH server implementation for shell access to the ALG
- Communication in the HA network with IPsec

Excluded are especially the following operations:

[2] If authentication at the smtprelay is not configured, the SMTP Server will perform the authentication.

- although some relays support encryption with TLS, this Security Target does not contain SFRs for the class FCS (Cryptographic Support) for TLS usage of the relays. Therefore these cryptographic operations are not part of the TSF.
- the cryptographic operations of the sshrelay. This Security Target does not contain SFRs for the class FCS (Cryptographic Support) for the sshrelay. Therefore these cryptographic operations are not part of the TSF.
- DNSSEC for DNS-based ACLs.

The TOE explicitely excludes the following options or services from its logical scope:

- the Custom HA mode
- the CARP PAP mode
- the Advanced Web Categories (AWC)

## 1.5 Conformance to VS-AP Firewall

In the context of an evaluation conforming to the German national requirements catalogue for the evaluation of IT security products suitable for processing Classified Information (CI) up to the level of VS-NfD [9], this Security Target aims for conformance to the product type Firewall using the CI Requirements Profile (VS-Anforderungsprofil, VS-AP) Firewall [2]. For practical reasons, this document refers to the English version [8] of the document. It is VS-NfD and VS-QVf conformant for the packages defined in the VS-AP.

The VS-AP claims are met, because each Security Assurance and Functional Security Component is at the same level or higher than the requirements of VS–Anforderungsprofil Firewall [8] and for VS-NfD [9] evaluations, including the packages NfD and QVf defined in the referenced document.

With editorial enhancements for clearer understanding and modifications for usability outside of the VSA context, this Security Target also uses the *Security Problem Definition* and augmented *Security Objectives* from [8]. The augmentations that have been added to the Security Objectives in order to formulate more specific requirements for the concrete product are compatibel with the Security Problem Definition, so that the original rationale is still valid. More details about the applied adaptations are provided in the respective Chapters 3 and 4.

# 2 Conformance Claims

This section contains all claims necessary for an evaluation conforming to a Common Criteria (CC) certification.

## 2.1 CC Conformance Claim

This Security Target is Common Criteria Part 2 extended and Part 3 extended conformant to the Common Criteria Version 2022, Revision 1 (November 2022) [11, 12, 14].

## 2.2 PP Claim, Package Claim

There are no Protection Profile claims.

This Security Target claims to be conformant to the Assurance Packet EAL4 augmented with ALC_FLR.2, ASE_TSS.2 and AVA_VAN.5. These components are defined in CC Part 3 [12]. The Security Target also claims ALC_PAM.1, which is an extended security assurance component defined in this Security Target.

## 2.3 Conformance Rationale

The Security Target has no Protection Profile claim, therefore no conformance rationale has to be given.

This Security Target uses extended functional component definitions (see section 5.1). Therefore it is Common Criteria Part 2 extended. It uses extended assurance requirements (see section 5.2). Therefore it is Common Criteria Part 3 extended.

# 3 Security Problem Definition

In order to clarify the nature of the security problem that the TOE is intended to solve, this section describes the following:

- Any assumptions about the security aspects of the environment and/or of the manner in which the TOE is intended to be used.
- Any known or assumed threats to the assets against which specific protection within the TOE or its environment is required.
- Any organizational security policy statements or rules with which the TOE must comply.

The modelling is directly taken from [8], with editorial adaptions in some cases for better clarity, or for adaptation to the specific case of the genugate, for example, by formulating specific assumptions regarding the use of LDAP or RADIUS servers. For the usage outside of VSA context, all references to the Security Operation Procedures (short: SecOPs) have been removed.[3]

## 3.1 Roles/Subjects

The Security Target defines the following users:

- **Administrator**
- **Auditor**
- **Attacker**

For details see [8], Chapter 2.1.

## 3.2 Assets/Objects

The Security Target defines the following primary assets:

**Data, functions and resources of the sensitive network**
: The primary asset is the data and functions of the sensitive network (i. e. of the internal network that shall be protected by the Firewall). This includes not only the data transported in the network and stored in the network components, whose confidentiality and integrity must be maintained, but also metadata such as those data which describe the network topology. Access to this data must be controlled and knowledge and unauthorized manipulation of the data must be prevented. Likewise, the manipulation of network functions and resources and their unauthorized use by external users must be prevented.

Secondary assets are data or resources required for technical reasons when operating the Firewall and that, if compromised, indirectly endanger the primary assets. The Security Target defines the following secondary assets:

**Firewall software and its configuration**
: All data and components whose manipulation would endanger the secure function of the Firewall must be protected with regard to their authenticity and integrity. In addition, the configuration of the Firewall must be protected with regard to its confidentiality if, for example, it allows conclusions to be drawn about the topology of the network to be protected. In addition dynamic configuration parts (settings about routing protocols) must be checked for integrity. In particular the software and the configuration of the Firewall are part of this secondary asset.

[3] Instead, on the level of the security objectives for the environment some specific requirements that need to be followed have been added, see Chapter 4.

**Firewall notifications and alerts**
If the Firewall detects security relevant events of its filter functionality, it generates notifications and alerts according to the configuration. These are stored in an audit trail.

The audit trail may only be read or exported by an **Administrator** or an **Auditor**. Only an **Administrator** is allowed to delete the audit trail. Since the audit trail could provide a possible external **Attacker** with information about potential vulnerabilities of the system or about the topology of the sensitive network, the confidentiality of the audit trail must be protected in addition to authenticity and integrity.

**Authentication data**
Authentication data are, for example, passwords used for the authentication of an **Administrator** or an **Auditor**, check values for those passwords, or asymmetric key pairs used for authentication. They must be kept confidential (in case of secret authentication data like passwords) and with integrity.

**Firewall log entries**
The Firewall generates log entries for all security-relevant administrator activities.

Log entries may only be read or exported by an **Administrator** or an **Auditor**, and they may only be deleted by an **Administrator**. Here, too, the protection requirement therefore includes authenticity, integrity and confidentiality.

**Cryptographic keys**
These are all cryptographic keys that may be needed for any of the following purposes:

- Authentication
- Secure storage of authentication data
- Securing a secure channel, for example for remote administration
- Securing VPN connections (as provided by the Firewall)
- Trust Anchor for public key infrastructures, e. g. CA certificates for TLS client certificate verification

For symmetric keys and private asymmetric keys, their authenticity, confidentiality and integrity must be protected; for public keys, authenticity and integrity must be protected.

All assets are taken from [8], Chapter 2.2, with some editorial adaptions for clarification. In the description of the secondary assets **Firewall notifications and alerts** and **Firewall log entries** the administrative access rules for log data are slightly changed, in line with the concrete TOE implementation and with the derived security objectives **O.AuditTrail** and **O.Logging**. In [8], the asset description is in this respect more specific than the security objectives, and requires rather a specific implementation than a security policy, which is seen as a deficiency of [8].

## 3.3 Threats

The Security Target defines the following threats:

**T.DataTraffic**[4]
This threat refers to any attempts by an external **Attacker** to send such data to the Firewall or through it to the sensitive network (i. e. the internal network to be protected) that compromises the security of the Firewall itself (by compromising any of the secondary assets **Firewall software and its configuration**, **Firewall notifications and alerts**, **Authentication data**, **Firewall log entries** and **Cryptographic keys**) or of the sensitive network (i. e. asset **Data, functions and resources of the sensitive network** within the sensitive network).

[4] misspelled **T.Datatraffic** in [8]

For example, an attacker sends manipulated packets to the Firewall or into the sensitive network to exploite vulnerabilities (such as buffer overflow) within the Firewall or in the network it protects and thereby attacking any of the listed assets. This attack includes fragmentation attacks.

At higher protocol levels, this includes the threat of malware entering into the sensitive network or the Firewall itself. The threat also includes attacks that aim to bypass detection and filtering measures of the Firewall and thereby penetrate the network (firewall evasion), for example through unusual or incorrect header or data encoding.

Also included in this threat is if an **Attacker** manages to gain an advantage through non-standard, or non-state of the art protocols. In doing so, he could exploit such protocols or poorly implemented key management to carry out attacks. A successful attack destroys the integrity and confidentiality of the data flow in the network and may lead to compromise of the sensitive network or the Firewall itself.

**T.Network**

This threat includes all attempts to gain unauthorized information from or about the sensitive network to be protected (i. e. the internal network), or to modify functions and resources (asset **Data, functions and resources of the sensitive network**) in this network without authorization, or to use them without authorization.

The primary aspect of this threat is an **Attacker** attempting to read sensitive data from the sensitive network.

An example of further scenarios is that an **Attacker** succeeds in using services of components of the sensitive network from the outside in a way that is not intended. This also includes the analysis or observation of network traffic.

Another attack scenario is that an **Attacker** manages to create a network topology to find out which computers are included in the network or find out the IP addresses and corresponding service ports. The **Attacker** can use this information to perform more targeted attacks against components in the network.

**T.Compromise**

This threat includes all attempts by an **Attacker** to compromise the Firewall itself in order to achieve further attack targets. This includes making unauthorized changes to the software or the configuration of the software, as well as making unauthorized changes to log data or the audit trail. This also includes attempts to gain unauthorized knowledge of the aforementioned objects (assets **Firewall software and its configuration**, **Firewall notifications and alerts** and **Firewall log entries**) of the Firewall that require protection.

**T.Access**

An **Attacker** manages to gain unauthorized access to the Firewall and modify the security functionality unnoticed. This results in a compromised Firewall (with possibly compromised secondary assets **Firewall software and its configuration**, **Firewall notifications and alerts**, **Authentication data**, **Firewall log entries** and **Cryptographic keys**) with no knowledge of the compromise by the authorized individuals.

An **Attacker** succeeds in compromising access permissions (e. g. by compromising **Authentication data** or access control lists as part of the **Firewall software and its configuration**) to Firewall data and thus having permanent access to the Firewall and the network's sensitive data. Compromising access permissions can happen, for example, by replacing existing permissions, modifying them, or gaining administrator privileges.

An **Attacker** succeeds in gaining access to the Firewall as an **Administrator** or **Auditor**, for example, by masquerading as an **Administrator** or **Auditor** to the Firewall, by masquerading as a Firewall to the **Administrator** or **Auditor**, by a replay attack, or a man-in-the-middle attack.

Unauthorized access allows the **Attacker** to perform malicious actions that would compromise the security functionality of the Firewall and all networks it protects.

**T.AuthenticationData**[5]
This threat includes all attacks against the asset **Authentication data** managed by the Firewall.

For example, an **Attacker** may be able to guess a password that is not secure or is too weak and thus gain successful access to the Firewall. In this way, the **Attacker** can impersonate as a known **Administrator** or **Auditor** and take advantage of this trust.

An **Attacker** manages to obtain the private **Cryptographic keys** used by the Firewall. This includes third party cryptographic material that is used as a trust anchor in the Firewall, like e. g. CA certificates.

An **Attacker** manages to successfully masquerade as an **Administrator** or other end device, for example, despite secure protocols but weak authentication endpoints (shared password, easy-to-guess password, password transmitted in plain text). The success of such an attack results in the loss of confidentiality and integrity of the data flow of the sensitive network and may lead to compromise of the Firewall itself.

**T.Update**
An **Attacker** manages to manipulate an update to the Firewall, compromising the Firewall's security functionality by compromising the asset **Firewall software and its configuration**.

The threats are taken from [8], Chapter 2.3, with some few editorial enhancements for clearer understanding. In **T.Access** the **Auditor** was added as potential victim of an attack where the **Attacker** is masquerading as a Firewall, which seems to be simply forgotten but in line with the intention of [8].

Previously defined assets and roles/subjects are set in a **bold** font. Threat agent is in all cases the **Attacker**.

## 3.4 Assumptions

The Security Target defines the following assumptions that are made on the operational environment:

**A.FirewallSystem**[6]
It is assumed that the Firewall operator has a concept for the Firewall system, of which the Firewall product is a component.

It is assumed that the concept for the Firewall system considers the relevant and applicable national or international requirements, like in Germany in particular the requirements of the BSI from the IT-Grundschutz-Kompendium [3]. For the latter, reference is made in particular to the modules NET.1.1 and NET.3.2.

It is assumed that the Firewall is used in a Firewall system implemented according to such a concept.

**A.PhysicalProtection**[7]
If the physical variant of the Firewall is used, it is assumed that the Firewall is operated in its operational environment with sufficient physical protection so that physical attacks, manipulations and disturbances of the (physical) interfaces of the Firewall or its functionality are prevented.

If the virtualized variant of the Firewall is used, i. e. if genugate Virtual is used, it is assumed that it is sufficiently protected in the virtualized operating environment so that attacks, manipulations and interruptions of the virtual interfaces of the Firewall respectively its functionality are excluded.

[5] misspelled **T.Authenticationdata** in [8]
[6] misspelled **A.Firewall_system** in [8]
[7] misspelled **A.Physical_protection** in [8]

**A.DataTraffic**[8]

It is assumed that all traffic both into and out of the sensitive network to be protected (i. e. the internal network) passes through the Firewall or any other equivalent product, like e. g. products with the same level of certification or approval, and which are compliant to the Firewall operator's concept for the Firewall system.

If the virtualized variant of the Firewall is used, i. e. if genugate Virtual is used, it is assumed that the hypervisor reliably forwards the packets to the virtual machine in which the Firewall is installed.

**A.Administrator**

It is assumed that an **Administrator** for the management tasks of the Firewall is trustworthy, competent and is trained accordingly.

It is also assumed that the security policy of the internal network allows **Administrators** only to view and modify the network components and network configurations. This does not apply to read access to the log entries of the Firewall in accordance with the assumption **A.Auditor**.

If the Firewall is removed, the **Administrators** ensure that unauthorized access to sensitive residual information (such as log entries, authentication data, cryptographic keys, etc.) is no longer possible, for example by securely deleting the data.

In addition, it is assumed that the **Administrator** performs his tasks only via a secure workstation where it is ensured that unauthorized persons cannot spy on the **Administrator's** activities.

**A.Auditor**

It is assumed that an **Auditor** of the Firewall is trustworthy and is competent and appropriately trained for his tasks.

Further, it is assumed that the security policy of the internal network allows **Auditors** (and **Administrators**) only to view and export the log entries.

In addition, it is assumed that the **Auditor** performs his tasks only via a secure workstation where it is ensured that unauthorized persons cannot spy on the **Auditor's** activities.

**A.Platform**

It is assumed that the hardware and software platform on which the Firewall runs is appropriately selected, installed and configured, and in case of security alerts security updates of affected platform components are installed promptly. If parts of the platform perform security functionality for the Firewall, like e. g. if LDAP or RADIUS servers are used to support the authentication process, it must adequately protect this process and the associated authentication data. In particular, these servers are subject to the same requirements on physical protection as the Firewall itself.

If the virtualized variant of the Firewall is used, i. e. if genugate Virtual is used, the virtualized environment of the Firewall must also be appropriately selected, installed and configured. In particular it is assumed that unused services and interfaces are disabled in the virtualized environment.

**A.Update**

It is assumed that the Firewall software is updated by an **Administrator** on a regular basis, but especially on an ad hoc basis, as soon as new updates are available in response to vulnerabilities that have become known.

**A.Time**

It is assumed that a trustworthy time is provided to the Firewall.

**A.Functionality**

If the physical variant of the Firewall is used, i. e. if the physical genugate is used, it is assumed

[8] misspelled **A.Datatraffic** in [8]

that the platform on which the Firewall is operated only runs those services or applications that

- are required for the operation or administration of the Firewall or
- provide additional security services defined in the concept for the Firewall system (such as an intrusion detection system or an intrusion prevention system).

It is assumed that the platform on which the Firewall is running is designed for providing its security functionality and that unused services and interfaces are disabled. Security alerts also for the components of the platform must be tracked, and security updates of these components must be installed promptly.

**Application Note:** If the virtualized variant of the Firewall is used, i. e. if genugate Virtual is used, the assumption **A.Functionality** is not applicable. Instead, the virtualization parts in the assumption **A.Platform** then completely take over the assumptions from **A.Functionality**.

The assumptions are taken from [8], Chapter 2.5, with some few editorial adaptions for clarification, in particular with respect to the physical or virtualized variant of the Firewall, and with adaptations to the specific case of the genugate. In **A.Platform** the reference to the Security Operation Procedures (which is not applicable outside of VSA context) has been removed, instead, **A.Platform** as well as (with more details) the security objective **OE.Platform** have been enhanced by the specific requirements for the platform for the concrete genugate product (see also Chapter 4). In **A.Platform** as well as in **A.Functionality** a formulation was added which requires that, in case of relevant security alerts, security updates of affected platform components are installed promptly. This seems sensible and in line with the intention of [8] with regard to the respective requirement in security objective **OE.Update**. The administrative access rules for log data as formulated in **A.Administrator** and **A.Auditor** are adapted, in line with the concrete TOE implementation and with the derived security objectives **O.AuditTrail** and **O.Logging**.[9] For broader international use, in **A.FirewallSystem** the reference to requirements of BSI was replaced by more general applicable national or international requirements, with BSI requirements as specific examples for Germany.

Previously defined roles/subjects are set in a **bold** font.

## 3.5 Organizational Security Policies

The Security Target defines the following organizational security policies that have to be met:

**OSP.Management**

The Firewall provides management options for the authorized **Administrator** to configure the Firewall in a secure manner.

The set of rules of the Firewall corresponds to the security requirements of the sensitive network (i. e. of the internal network that shall be protected by the Firewall). This involves analyzing which accesses are needed from the inside to the outside and which from the outside to the inside (if such accesses are allowed at all) and designing the rules such that only the permitted accesses and data streams are allowed and all others are prevented. This also includes completely blocking unneeded protocols.

**OSP.Audit**

All security-related activities and events are recorded, processed and audited.

On the one hand, this concerns the notifications and warnings after application of the configured set of Firewall rules, which are kept in a so-called audit trail, as well as the log entries regarding the administrator activities.

[9] In the VS-AP Firewall [8] the formulation of Assumptions **A.Administrator** and **A.Auditor** regarding administrative access rules for log data are more specific than the formulations in security objectives **O.AuditTrail** and **O.Logging**, and they do not only require a defined security policy but determine a concrete implementation. This is seen as a deficiency of the VS-AP Firewall [8].

**OSP.Crypto**
The cryptographic functions or protocols that are implemented in the Firewall use only cryptographic algorithms and procedures that are approved by the BSI.

**OSP.TrustworthyVirtualizationEnvironment**
If the virtualized variant of the Firewall is used, i. e. if genugate Virtual is used, then the operator must provide a secure virtualization environment to implement virtualization risk mitigation measures, such as mitigation measures of hypervisor risks or virtualized network risks. This also includes a sufficiently protected environment for administration and management of the virtualization environment in order to reduce the risks of attacks on the virtualization host.

**Application Note:** This organizational security policies does not apply to the physical genugate.

The organizational security policies are taken from [8], Chapter 2.4, with some few editorial adaptions for clarification, in particular with respect to the physical or virtualized variant of the Firewall. For the usage outside of VSA context, in **OSP.TrustworthyVirtualizationEnvironment** the reference to the Security Operation Procedures has been removed.[10] Previously defined roles/subjects are set in a **bold** font.

[10] Instead, specific requirements that need to be followed have been added to the security objective **OE.Platform**, see Chapter 4.

# 4 Security Objectives

The purpose of the security objectives is to describe the planned response to a security problem or threat. Threats can be directed against the TOE or the security environment. The CC identifies two categories of security objectives:

- security objectives for the TOE
- security objectives for the operational environment

Both categories of security objectives are intended to be the same or exceeding as in the requirements document “VS–Anforderungsprofil Firewall” [8] of BSI. The modelling is therefore mainly taken from [8], Chapters 3.1 and 3.2, with editorial adaptions for clarification and partial shortage where this fits, e. g. by removing general examples and some explanations, and with adaptions to the specific case of the genugate. This includes adding more specific requirements in some cases, e. g. more specific administrative access rules for log data and/or configuration data as formulated in **O.Rules**, **O.AuditTrail**, **O.AccessControl** and **O.Logging**. In **O.Crypto** all TOE security functionality that uses cryptographic operations is explicitly listed. In **OE.Administrator** some specific rules regarding authentication data have been added, and the requirement for secure workstations has been enhanced by a requirement for a separate administration network in which the secure workstation must be placed. Some adaptations for clarification have been done with respect to specific requirements on the physical or virtualized variant of the Firewall, in particular in **OE.Platform** and **OE.Functionality**.[11]

For the usage outside of VSA context, in several security objectives for the operational environment the specific formulations regarding the Security Operation Procedures or handling of Classified Information (CI) has been removed or replaced by more general formulations. For instance, in **OE.Administrator** and **OE.Auditor** the requirement that the Administrator/Auditor must be authorized to handle CI of the required level is replaced by the requirement to be authorized for his tasks. In **OE.Rules** the term CI network has been replaced by the term sensitive network (which was also used throughout the remainder of the VS-AP Firewall [8]). In **OE.Platform**, the requirement that the Security Operating Procedures of the Firewall must describe the specific requirements for the selection and secure configuration of the platform was replaced by the direct formulation of the specific requirements that need to be followed, like, for example, by requirements regarding the use of LDAP or RADIUS servers or OSPF routers.

In all cases these changes are in line with the changes that had been done to the formulations in the security problem definition, where e. g. also the text was adapted for the usage outside of VSA context.

## 4.1 Security Objectives for the TOE

The Security Target defines the following security objectives for the TOE:

**O.Rules**

The TOE must enforce filter rules to control the flow of all data between all connected networks and computers, in particular to secure the sensitive network (i. e. internal network to be protected). This requirement applies to encrypted as well as unencrypted connections.

The TOE must be able to implement “Stateful Packet Inspection”, especially also for stateless protocols such as ICMP and UDP.

[11] The Application Note on **OE.Functionality** in [8] claims that for a virtualized Firewall **OE.Functionality** is not applicable and that instead **OE.Platform** takes over the respective objectives. But this is not consistently implemented in [8], for instance because **OE.Functionality** still contains explicit requirements on virtualized Firewalls, which is seen as a deficiency of [8]. That is why in this Security Target the requirements on a virtualized firewall formulated in **OE.Functionality** in [8] had been explicitly shifted to **OE.Platform**. An exception to this is the prohibition to operate approved and non-approved systems on the same hardware, which was dropped entirely because it is not applicable outside of VSA context.

The filter rules must include a Default-Deny-Policy, i. e. drop every packet that cannot be explicitly assigned to a permitting rule. It must be possible for an Administrator or Auditor to see the default settings of all filter rules.

If the TOE detects that it cannot reliably enforce the set of rules (for example, because certain required processes could not be initialized correctly), it must ensure that the data flow between the different networks is blocked and that the TOE cannot be bypassed. However, this should not restrict the local administration access.

**O.AuditTrail**
The TOE must provide a function to generate and store notifications and alerts depending on the previously configured filter rules, and must provide means to present a readable and searchable view on these audit log entries, access-protected to Administrators and Auditors.

**O.AccessControl**
The TOE has to establish a user and rights management for the execution of management functions (including configuration of the TOE and software update) and for the evaluation of audit log entries. The TOE must ensure that only Administrators can change the TOE configuration.

**O.Authentication**
Administrators and Auditors must be successfully authenticated before they can perform any management operation according to their role.

**O.AuthenticationData**
All authentication data is protected according to its protection needs.

**O.Management**
The TOE must provide at least the following management functions to the Administrator:

- secure installation of the TOE,
- software update,
- TOE configuration, including configuration of its filter rules,
- audit log management,
- user and rights management.

The TOE functionality can be limited by an Administrator to the necessary extent.

**O.Logging**
The TOE must provide functions for the generation and integrity-protected storage of audit log entries, and means to present a readable and searchable view on the audit log entries, access-protected to Administrators and Auditors.

**O.Update**
The TOE must implement a secure software update mechanism that ensures the authenticity and integrity of installed software updates.

**O.Crypto**
All cryptographic operations that are implemented by the TOE in the context of software update, Administrator or Auditor authentication or to provide a secure channel for remote administration, or for synchronization between HA peers in the HA network, must use only secure cryptographic methods.

Key management functions must be implemented such that the keys are protected against tampering and unauthorized access. In the case of key generation or secure deletion of keys and intermediate cryptographic values, the keys and operations must fulfil the requirements from BSI TR 02102-1 [4].

## 4.2 Security Objectives for the Environment

The Security Target defines the following security objectives for the operational environment:

**OE.FirewallSystem**

The Firewall operator must create an overarching concept for the Firewall system. This concept must describe how the Firewall product, in interaction with other products, ensures the security of the sensitive internal network(s) to be protected.

The concept for the Firewall system must take into account the applicable national or international recommendations, like in Germany especially the BSI requirements as specified in the IT-Grundschutz-Kompendium [3], and here in particular the modules NET.1.1 and NET.3.2.

The concept for the Firewall system should demonstrate how appropriate filtering is performed at all protocol layers and for all protocol types, or why this is not considered necessary for certain protocol layers. The omission of filtering must not create a channel through which sensitive data can flow. The possible filtering thus ranges from the network header level of the packets to malware detection at the application layer level.

In addition, the concept should define whether the topology of the sensitive network must be obfuscated against external networks or why this may not be necessary in exceptional cases. If obfuscation is necessary, it must be realized by a proxy that can make the internal structure completely invisible to external networks by representing all requests to the outside as its own.

A typical configuration for a Firewall system would be something like the so-called P-A-P structure, where one product implements a packet filter (P) and the genugate firewall implements ALG and packet filter (A-P). The concept for the Firewall system must specify which component performs each function. If the virtualized form of the TOE is used, and if a multi-level Firewall system is intended (P-A-P structure), each level should run on dedicated hardware.

The concept must also include specifications on whether, for example, a “demilitarized zone” (DMZ) will be established.

Note: If there are already explicit specifications, for example from the BSI or a comparable authority, for a Firewall system, or if the operator’s information security concept already adequately describes the Firewall system, there is generally no need for a separate concept.

**OE.Rules**

The set of firewall rules configured by the Administrator for the Firewall ensures that only permitted connections are possible, while all accesses and data streams that are not required are prevented. Depending on the concept for the Firewall system, it must be ensured that no direct accesses from outside into the sensitive internal network are possible. The concept may also result in data from a “demilitarized zone” (DMZ) having to be fetched from the sensitive network (push-pull principle) and connections that cross the Firewall system having to be initialized from the sensitive network. All protocols that are not required are completely deactivated. All configurations must be conformant to the overarching concept for the Firewall system.

**OE.PhysicalProtection**

Physical security of the Firewall product must be ensured. If the virtualized form of the TOE is used, the security of the virtualized operating environment of the Firewall must be ensured. The protection requirements of this must be derived from the protection requirements of the network(s) to be protected by the Firewall. Optional add-on modules (e. g. SIP) must always be kept up to date in order to exclude known vulnerabilities.

**OE.DataTraffic**

All data traffic into and out of the network to be protected passes through the Firewall. The only exceptions to this may be connections that are routed via products that are also approved for the network classifications and do not override the networks’ defined security policies.

**OE.Administrator**

An Administrator of the Firewall must be trustworthy and in particular authorized for his tasks. In addition, he must be appropriately trained in the operation of the Firewall and be appropriately qualified to perform the management tasks.

Passwords and comparable authentication data for Administrators are chosen and managed sufficiently secure according to appropriate guidelines for passwords and authentication token. The following rules must be included in these guidelines: If remote administration via SSH is used, usage of RSA keys (SSH user key or host key) is not allowed, ECDSA keys must be used instead. If RSA keys are used for TLS client or server certificates for remote administration, the RSA key size must be at least 3072 bits. The allowed maximum for consecutive failed authentication attempts for remote administration with password-based authentication mechanism must not be configured to a value that exceeds the default value of 5.

If the Firewall or components of the sensitive network are decommissioned or removed from the operational environment, the operator on whose behalf the Administrator is acting must ensure that there can be no unauthorized access to any sensitive residual information that may still be present (such as audit log entries, authentication data, cryptographic keys, etc.). There must be guidance that explains how this data is to be handled securely (in terms of deletion, destruction, storage). Existing applicable national or international disposal requirements for the product, like in Germany especially BSI disposal requirements, must be observed.

The operational environment must provide a separate administration network to access the management functions of the Firewall and, if the virtualized form of the TOE is used, of the virtualization host. The Administrator shall perform his tasks only via a secure workstation that is placed in the administration network and where it is ensured that unauthorized persons cannot spy on the Administrator's activities.

**OE.Auditor**

An Auditor of the Firewall must be trustworthy and in particular authorized for his tasks. In addition, he must be appropriately trained and qualified for his tasks.

An Auditor must also fulfil the requirements in **OE.Administrator** on a secure workstation, and on passwords and comparable authentication data.

**OE.Platform**

The platform on which the Firewall software runs must be appropriately selected, installed, and configured:

If the physical variant of the Firewall is used, the Firewall software must be installed on one of the hardware servers genugate S, genugate M or genugate L, revisions 3.0 or 4.0, or on the Infodas hardware *SDoT Server V3B*. If the virtualized variant of the Firewall is used, the Firewall software shall only be installed in a trustworthy virtualization environment. Only the hypervisors VMware ESXi or KVM are allowed, and the virtualization server must use an Intel or AMD CPU, supporting the virtualization features VT-x (for Intel CPUs) or AMD-V (for AMD CPUs). Mitigation measures of hypervisor risks or virtualized network risks must be implemented. Only those services or applications may run on the virtualization environment that are required for the operation or administration of the Firewall or that provide additional security services defined in the concept for the Firewall system (such as an Intrusion Detection System or an Intrusion Prevention System).

The platform must be securely configured. This includes both the hardening of the platform (e. g., disabling unneeded services in an operating system) and the appropriate use of any rights management (different processes should run with different user permissions and should communicate only via defined interfaces).

If the Firewall is in long-term operation (e. g. 24/7 continuous operation), the time interval for re-authentication must be configured to a reasonable value (as e. g. already specified in the

default configuration). In particular, the obligation for re-authentication must not be switched off.

If an LDAP or a RADIUS authentication server is used to support the authentication process, the server must be installed and configured securely and appropriately. The authentication server must adequately protect the authentication operations and the associated authentication data. In particular, the server is subject to the same requirements on physical protection as the Firewall itself. Furthermore, the environment must provide a secure network connection between the authentication server and the TOE.

If the security of the Firewall is supported by further external servers, e. g., by NTP time servers for a reliable time source, or by log servers (audit servers) for storing log files, these must also be connected to the Firewall via a protected network connection. Also these servers are subject to the same requirements on physical protection as the Firewall itself.

If OSPF routers are used, they must be set up in a protected area. Separate network areas must be set up for connections to the Firewall and connections to the internal network.

If the high availability (HA) option of the TOE with OSPF is used, correctly configured OSPF routers must be provided in the internal network, which are suitably secured against attacks from the internal network.

**OE.Update**
All TOE software must be regularly checked by an Administrator to ensure that it is up to date. Necessary updates due to known vulnerabilities must be installed in a timely manner after being published.

The same applies to the software portions of the platform on which the Firewall is running, like e. g. an associated operating system, or virtualization environment, and to optional add-on modules that are installed on the Firewall platform, like e. g. virus scanners. Also these must always be kept up to date in order to eliminate known vulnerabilities.

**OE.Time**
It must be ensured that the operational environment provides trustworthy time to the Firewall.

**OE.Functionality**
If the non-virtualized form of the TOE is used, i. e. if the physical genugate is used, only those services or applications may run on the platform on which the Firewall is operated that are required for the operation or administration of the Firewall or that provide additional security services defined in the concept for the Firewall system (such as an Intrusion Detection System or an Intrusion Prevention System). A general (non-security-related) web server, on the other hand, would not be permitted on the same platform.

The platform on which the Firewall runs is designed for its security functionality to be performed. Unused services and interfaces must be disabled.

**Application Note:** If the virtualized variant of the Firewall is used, i. e. if genugate Virtual is used, the security objective **OE.Functionality** is not applicable. Instead, the virtualization parts in the security objective **OE.Platform** then completely take over the objectivs from **OE.Functionality**.

**OE.Crypto**
In all cases where the Firewall requires cryptographic operations, for example in the context of Administrator or Auditor authentication or to provide a secure channel for remote administration, then all IT components in the operational environment with which the Firewall therefore communicates also use only secure cryptographic procedures, e. g. as described in BSI TR 02102 part 1 to 4 [4, 5, 6, 7].

If parts of key management functions take place in the operational environment of the Firewall, these parts must be implemented such that the keys are protected against tampering and unauthorized knowledge. In the case that parts of the key generation take place in the operational environment, or secure deletion of keys and intermediate cryptographic values, the

keys and operations must satisfy the relevant national or international requirements, like e. g. the requirements from BSI TR 02102-1 [4].

**OE.LifecycleConcept**

The manufacturer must create a concept for the Firewall that covers the entire life cycle of the Firewall.

This concept has to include the security of all processes during production and delivery of the devices or software, for ensuring that the operator receives authentic devices or software. The concept also applies to possible suppliers of security-relevant components and other subcontractors involved in the manufacturing process and configuration of the devices.

For software components of the Firewall, the operator must be able to verify authenticity by, for example, transmitting the hash value or signature via these components in a secure manner.

The way in which the manufacturer of the software keeps the security of the Firewall constantly up to date, for example by providing software updates, must also be described in the concept. It must also be clear how the manufacturer considers current security alerts that relate to the Firewall or its components (TOE and non-TOE parts).

Furthermore, the manufacturer describes the procedure if the operator decides to decommission the Firewall. In particular, it must be explained how the operator removes sensitive data from the Firewall and which components must be physically destroyed.

The operator must track any security alerts also for third-party hardware and software platform where TOE is installed on (like the virtualization environment in case of the virtualized variant of the Firewall), or other parts of the platform that perform security functionality for the Firewall (like e. g. LDAP or RADIUS servers that might be used to support the authentication process), and install security updates of these components promptly.

## 4.3 Security Objectives Rationale

In this chapter it must be shown that the security objectives are consistent with the security problem definition.

Since the security modelling of this ST follows mainly the modelling of [8], the rationale for security objectives in this ST can also mainly be taken from [8], Chapter 3.3.

The following modifications must be considered:

- Note that the misspelled names of threats and assumptions in [8] need to be corrected: In order to comply to the naming in the original, German version of the requirements document “VS-Anforderungsprofil Firewall” [2] and hence also to the naming in this Security Target, throughout Chapter 3.3 in [8], the reader must assume that the term **T.Datatraffic** is replaced by **T.DataTraffic**, **T.Authenticationdata** is replaced by **T.AuthenticationData**, **A.Firewall_system** is replaced by **A.FirewallSystem**, **A.Physical_protection** is replaced by **A.PhysicalProtection**, and **A.Datatraffic** is replaced by **A.DataTraffic**.
- Table 3 in [8], Chapter 3.3, provides also a rationale why **OE.DataTraffic** contributes to counter the threat **T.Compromise**, but a respective cross in Table 2 is missing by mistake. Hence the Table 2 in [8] must be corrected by marking also Objective **OE.DataTraffic** with a cross in the row for **T.Compromise**.
- A small editorial correction needs to be done in the justification for the part of **T.Update** that is covered by **OE.FirewallSystem** and **OE.Rules**: the last word “at” in the row for **T.Update** in Table 3, Threat rationale, needs to be deleted. See the original German text in [2] for comparison.
- Another small editorial correction needs to be done in the justification for the part of **OSP.Audit** that is covered by **OE.FirewallSystem**: the last word “at” in the row for **OSP.Audit** in Table 4, OSP rationale, needs to be deleted. See the original German text in [2] for comparison.

- In Table 5, Assumptions rationale, in [8] the rationale for the coverage of **A.PhysicalProtection** by **OE.PhysicalProtection** must be shortened to e. g. “OE.PhysicalProtection contributes to fulfilling the requirement”. The exception formulated in the first half of that sentence is no longer required since the references to the VSA had in this ST been removed from the text of **OE.PhysicalProtection**.

In this Security Target, some security objectives for the operational environment have been enhanced by more specific requirements. For instance, **OE.Platform** has been enhanced by specific requirements for the platform for the concrete product, e. g. requirements on the hardware or virtualization environment, and requirements on the usage and configuration of external servers that support the security of the Firewall system (like LDAP, RADIUS, NTP time servers or audit servers), and on OSPF routers. These enhancements directly contribute to fulfilling the respective assumptions and policies and threats, and the explanations in the existing rationale can remain unchanged in this regard.

The changes that had been applied in this ST to the formulation of the security problem definition for the usage outside of VSA context are in line with the changes that had been done, with the same intention, to the formulations of the security objectives. Also in this regard the existing rationale can remain unchanged.

# 5 Extended Components Definition

## 5.1 Class FPT: Protection of the TSF

The class has been augmented by two families.

### 5.1.1 Simple Self Test (FPT_SST)

#### Family behaviour

The family defines the requirements for the self-testing of the TOE with respect to some expected correct operation. Examples are expected running processes or expected files at some location in the file system. These tests can be carried out at start-up, periodically, at the request of the authorized 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.
The requirements of this family are also needed to detect the corruption of TOE executable code (i. e. TOE software) and TOE data by various failures that do not necessarily stop the TOE's operation (which would be handled by other families). These checks must be performed because these failures may not necessarily be prevented. Such failures can occur either because of unforeseen failure modes or associated oversights in the design of hardware, firmware, or software, or because of malicious corruption of the TOE due to inadequate logical and/or physical protection.

#### Component levelling


FPT_SST.1EX TOE testing, provides the ability to test the TOE's correct operation. These tests may be performed at start-up, periodically, at the request of the authorized user, or when other conditions are met. It also provides the ability to verify the integrity of TOE data and executable code.

#### Management: FPT_SST.1EX

The following actions could be considered for the management functions in FMT:

a) management of the conditions under which TOE self testing occurs, such as during initial start-up, regular interval, or under specified conditions;

b) management of the time interval if appropriate.

#### Audit: FPT_SST.1EX

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

a) Basic: Execution of the TOE self tests and the results of the tests.

**FPT_SST.1EX TOE testing**

Hierarchical to: No other components.

Dependencies: No dependencies.

**FPT_SST.1EX.1 The TSF shall run a suite of self tests [selection: *during initial start-up*, *periodically during normal operation*, *at the request of the authorized user*, *at the***

***conditions*** **[assignment: *conditions under which self test should occur*]] to perform the following checks: [assignment: *list of self tests*].**

**FPT_SST.1EX.2 The TSF shall provide authorized users with the capability to query the results of the following checks: [assignment: *list of self tests*].**

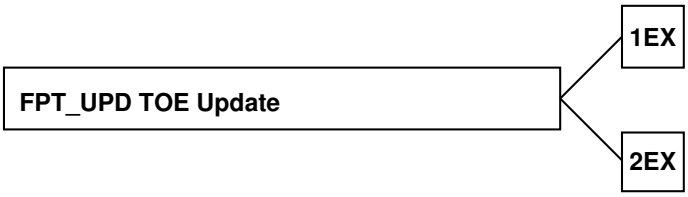
### 5.1.2 TOE Update (FPT_UPD)

The family specifies the secure and correct installation of patches from the developer by an administrator. The new family is added to the class FPT, because it protects the TSF from manipulation through the installation of malicious patches.

Family behaviour

The requirements of this family assure that only authorized patches from the developer can be installed in a secure and correct way by an administrator. The family has two components, which are independent from each other.

Component levelling




FPT_UPD.1EX Trusted update, requires that patches are signed using the specified cryptographic standards.
FPT_UPD.2EX Update identification data, requires that the patches have a unique patch level that is updated at the same time.

Management: FPT_UPD.1EX.1, FPT_UPD.1EX.2

The following actions could be considered for the management functions in FMT:

a) determining the time when to apply the patches.

Audit: FPT_UPD.1EX, FPT_UPD.2EX

The following actions should be auditable if FAU_UPD.1EX TOE Update is included in the PP/ST:

a) Basic: The result of the patch update.

#### FPT_UPD.1EX Trusted update

Hierarchical to: No other components.

Dependencies: FCS_COP.1 Cryptographic operation.

**FPT_UPD.1EX.1 The TOE shall cryptographically verify additional code/patches to itself using a digital signature prior to installation using schemes specified in [assignment: *FCS_COP.1 SFR*].**

**FPT_UPD.1EX.2 A modification of the TOE shall only be allowed if the software update**

- **is intended for the current software version,**
- **has the correct patch level and**
- **has been cryptographically verified with regard to integrity and authenticity.**

#### FPT_UPD.2EX Update identification data

Hierarchical to: No other components.

Dependencies: No dependencies.

**FPT_UPD.2EX.1 The TSF shall verify if the activation of the patch and the update of the identification data have been both completed.**

**FPT_UPD.2EX.2 The TSF shall update the active identification data when the patch is applied in order to keep the system in a defined state.**

**FPT_UPD.2EX.3 The TSF shall use the maintenance mode to activate the final TOE.**

## 5.2 Class ALC: Life-cycle support

The class ALC has been augmented by one family ALC_PAM that specifies the secure and correct generation of patches by the developer. The new family is added to the class ALC because it deals with the patch aspect of the product life-cycle, which is not explicitly covered by the other families of class ALC. The definition of ALC_PAM is taken from the ISO Technical Specification ISO/IEC TS 9569 "Information security, cybersecurity and privacy protection — Evaluation criteria for IT security — Patch Management Extension for the ISO/IEC 15408 series and ISO/IEC 18045" [16], including the related evaluation work units for this family.

Since this ISO Technical Specification [16] is not yet freely available e. g. on the Common Criteria portal https://commoncriteriaportal.org/, the definition of the new family is repeated here for convenience.

### 5.2.1 Patch Management (ALC_PAM)

#### Objectives

The objective of this family is to identify the policies and procedures to be implemented in the development process, which will be applied after the initial release of a TOE by the developer.

The application of the patch management (PAM) process cannot be always determined at the time of the initial evaluation. Nevertheless, it is possible to evaluate the policies and procedures that a developer has in place to perform the PAM process for a future patch release. It is also possible to obtain some evidence of the correct application of the procedures during the patching of the problems which are found during the evaluation of other assurance classes like AVA (vulnerability assessment) and ATE (tests).

The written PAM policies, processes and procedures are internal documents for the developer. These shall include instructions, among others, on how developers securely provide guarantees of authenticity to distribute and apply patches and how the life cycle of the keys, used for providing authenticity of new patches, is handled.

These procedures shall guarantee the secure development, the secure deployment, installation and activation for patches. Moreover, the procedures and the set of commands supporting them shall be described in the AGD (guidance) family.

#### Component levelling

This family contains only one component.

#### Application notes

None.

#### ALC_PAM.1 Patch management

Dependencies: ALC_FLR.2 Flaw reporting procedures.

Application note: The purpose of ALC_FLR is to build assurance of the flaw remediation procedures which are applied after security flaws were discovered. Separately, the purpose of ALC_PAM is to build assurance of the patch management processes which are applied when the behaviour of the initial TOE is changed independent of the type of change.
Therefore, the relationship of ALC_FLR to ALC_PAM is justified by the need to release patches to distribute flaw corrections.

Developer action elements:

**ALC_PAM.1.1D The developer shall provide patch management documentation (PMD) for the TOE.**

**ALC_PAM.1.2D The developer shall provide end-of-support information to the TOE users.**

**ALC_PAM.1.3D The developer shall follow the PMD on a regular basis.**

**ALC_PAM.1.4D The developer shall record evidence of the application of the PMD.**

**ALC_PAM.1.5D The developer shall release patches as defined in the PMD until the end-of-support of the TOE.**

**ALC_PAM.1.6D The developer shall follow the PMD to produce an updated set of evaluation evidence for each released patch at least until the stated end-of-support of the TOE.**

**ALC_PAM.1.7D The developer shall provide a channel used to check for the availability and/or download of patches with means to protect the channel according to the specified security capabilities of the TOE.**

**ALC_PAM.1.8D The developer shall create a security relevance report (SRR) for each patch release.**

Content and presentation elements:

**ALC_PAM.1.1C The PMD shall state the criteria used for the decision that a patch shall be released.**

**ALC_PAM.1.2C The PMD shall require the generation of an SRR and shall identify any applicable procedure.**

**ALC_PAM.1.3C The SRR shall describe the flaws, changes and impact that are related to the patch.**

**ALC_PAM.1.4C The PMD shall describe how to update the initial TOE evidence for any applicable SAR.**

**ALC_PAM.1.5C The PMD shall define how to record any PAM-related decision.**

**ALC_PAM.1.6C The PMD shall describe the mandatory patch-specific content for the preparative procedures and the operational user guidance.**

**ALC_PAM.1.7C The PMD shall describe the mandatory procedures during patch release.**

**ALC_PAM.1.8C The PMD shall contain rules regarding testing (using internal resources or using external third party) before a patch is released.**

**ALC_PAM.1.9C The PMD shall describe how end users are notified of a new patch and corresponding installation instructions.**

**ALC_PAM.1.10C The PMD shall describe all necessary developer procedures to support the patch functionality of the TOE.**

Evaluator action elements:

**ALC_PAM.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.**

# 6 Security Requirements

This section contains the security functional requirements, the security assurance requirements, and the rationale.

## 6.1 Security Functional Requirements

Most of the security functional requirements in this subsection have been drawn from the CC Part 2.

The functional requirements for FPT_SST.1EX, FPT_UPD.1EX and FPT_UPD.2EX are not drawn from CC Part 2 but have been defined as extended components.

In the following, the unmodified text from the functional requirement templates is displayed in a sans serif font. The operation ***assignment*** is set in a bold italic serif font. The operation *selection* is set in an italic serif font The operation **refinement** is set in a bold font. The iterations of a Part 2 requirement (if any) are denoted by repeating the requirements and adding an underscore and an iteration indicator in round brackets, e. g. FCS_COP.1_(RSA) as an iteration of FCS_COP.1. In a few occasions, the text has been modified slightly as an refinement operation.

### 6.1.1 Class FAU: Security audit

#### 6.1.1.1 Security audit data generation (FAU_GEN)

##### FAU_GEN.1 Audit data generation

**FAU_GEN.1.1** The TSF shall be able to generate audit data 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;
c) ***Starting and stopping of the system;***
***d) Failures detected by the TOE (FPT_FLS.1);***
***e) Changing operation modes (into maintenance mode and back to normal operational mode);***
***f) Relay configuration;***
***g) Loading of packet filter rules;***
***h) Relay usage;***
***i) Administration activity that changes the TOE configuration;***
***j) Authentication attempts.***

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

a) Date and time of the auditable event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event;
b) For each auditable event type, based on the auditable event definitions of the functional components included in the ~~**PP, PP-Module, functional package or**~~ ST, ***unspecified log data***.

#### 6.1.1.2 Security audit analysis (FAU_SAA)

##### FAU_SAA.1 Potential violation analysis

**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:

a) Accumulation or combination of ***configurable events (packet filter violations, selected messages of daemons, selected messages of the relays, ARP spoofing messages, time synchronization errors, usage of duplicate IP addresses, selected kernel messages and messages from the processes that implement the self-tests)*** known to indicate a potential security violation;
b) ***none***.

#### 6.1.1.3 Security audit automatic response (FAU_ARP)

##### FAU_ARP.1 Security alarms

**FAU_ARP.1.1** The TSF shall take ***configurable actions (log, wall, exec, mail, down, halt)*** upon detection of a potential security violation.

#### 6.1.1.4 Security audit review (FAU_SAR)

##### FAU_SAR.1 Audit review

**FAU_SAR.1.1** The TSF shall provide ***Administrators and Auditors*** with the capability to read ***all audit information*** from the audit data.

**FAU_SAR.1.2** The TSF shall provide the audit data in a manner suitable for the user to interpret the information.

##### FAU_SAR.2 Restricted audit review

**FAU_SAR.2.1** The TSF shall prohibit all users read access to the audit data, except those users that have been granted explicit read access.

##### FAU_SAR.3 Selectable audit review

**FAU_SAR.3.1** The TSF shall provide the ability to apply ***searches*** of audit data based on ***time, date, process id, additional log data (for relay audit data: relay type, connection state, IP addresses and ports, status of logged event, bytes transferred)***.

#### 6.1.1.5 Security audit event storage (FAU_STG)

##### FAU_STG.1 Audit data storage location

**FAU_STG.1.1** The TSF shall be able to store generated audit data on the *TOE itself, and* ***to transmit the generated audit data to an external audit server***.

##### FAU_STG.2 Protected audit data storage

**FAU_STG.2.1** The TSF shall protect the stored audit data in the audit trail from unauthorized deletion.

**FAU_STG.2.2** The TSF shall be able to *prevent* unauthorized modifications to the stored audit data in the audit trail.

##### FAU_STG.4 Action in case of possible audit data loss

**FAU_STG.4.1** The TSF shall ***execute a configurable action (default: inform the Administrators)*** if the audit data storage exceeds ***a configurable limit (default: 80 % of the storage capacity)***.

### FAU_STG.5 Prevention of audit data loss

**FAU_STG.5.1** The TSF shall *overwrite the oldest stored audit records*, and ***execute a configurable action (default: inform the Administrators)*** if the audit data storage is full.

**Application Note**: In addition, the audit data of specific critical event types are protected by OpenBSD file system flags: Flagged audit data are rotated only with the acknowledgement of the Administrator and only during maintenance mode.

## 6.1.2 Class FCS: Cryptographic support

### 6.1.2.1 Cryptographic key management (FCS_CKM)

### FCS_CKM.1_(RSA) Cryptographic key generation

**FCS_CKM.1.1_(RSA)** The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm ***RSA key pair generation*** and specified cryptographic key sizes ***minimum 3072 bit*** that meet the following: ***generation of RSA key pairs according to BSI TR 02102-1 [4] chapter 2.3.2, with random number generation according to FCS_RNG.1_(DRT.1)***.

### FCS_CKM.1_(ECDSA) Cryptographic key generation

**FCS_CKM.1.1_(ECDSA)** The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm ***ECDSA key pair generation*** and specified cryptographic key sizes ***of minimum 256 bit*** that meet the following: ***generation of ECDSA key pairs according to RFC 5656 [34], chapter 4, and SEC 1 [33], chapter 3.2.1, for ECC domain parameters***

***a) NIST curves P-256, P-384 and P-521 according to NIST SP 800-186 [30] chapter 3.2.1;***

***and with random number generation according to FCS_RNG.1_(DRT.1).***

### FCS_CKM.1_(PSK) Cryptographic key generation

**FCS_CKM.1.1_(PSK)** The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm ***pre-shared key (PSK) generation*** and specified cryptographic key sizes ***32 Byte*** that meet the following: ***generation of random PSK with random number generation according to FCS_RNG.1_(DRT.1)***.

### FCS_CKM.1_(DH) Cryptographic key generation

**FCS_CKM.1.1_(DH)** The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm ***Diffie-Hellman (DH) key pair generation*** and specified cryptographic key sizes ***4096 or 8192 bit*** that meet the following: ***generation of ephemeral DH key pairs according to RFC 4253 [38] chapter 8, for DH domain parameters***

***a) 4096 bit MODP group16 according to RFC 3526 [20], chapter 5;***
***b) 8192 bit MODP group18 according to RFC 3526 [20], chapter 7;***

***and with random number generation according to FCS_RNG.1_(DRT.1).***

## FCS_CKM.1_(ECDH) Cryptographic key generation

**FCS_CKM.1.1_(ECDH)** The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm ***Elliptic Curve Diffie-Hellman (ECDH) key pair generation*** and specified cryptographic key sizes ***of minimum 256 bit*** that meet the following: ***generation of ephemeral ECDH key pairs according to RFC 5656 [34], chapter 4, and SEC 1 [33], chapter 3.2.1, for ECC domain parameters***

***a) NIST curves P-256, P-384, P-521 according to NIST SP 800-186 [30] chapter 3.2.1;***
***b) Brainpool curves brainpoolP256r1, brainpoolP384r1 and brainpoolP512r1 according to RFC 5639 [23], chapter 3;***

***and with random number generation according to FCS_RNG.1_(DRT.1).***

## FCS_CKM.2_(PSK) Cryptographic key distribution

**FCS_CKM.2.1_(PSK)** The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method ***import or export of the pre-shared key (PSK) for IPsec for the purpose of secure communication between HA peers in the HA network*** that meets the following:

- ***commands for PSK import or export are only available at the Web-GUI, protected by a TLS session according to FTP_TRP.1_(TLS), are restricted to the Administrator, and via the TLS session, the PSK import and export is protected in confidentiality and integrity by the AEAD encryption algorithm AES GCM according to FCS_COP.1_(AES) (key sizes 128 or 256 bit).***

## FCS_CKM.5_(SSH) Cryptographic key derivation

**FCS_CKM.5.1_(SSH)** The TSF shall derive cryptographic keys

***a) AES encryption keys***
***b) HMAC integrity protection keys***

from ***ephemeral shared secret as agreed for SSH according to FCS_COP.1_(ECDH) or FCS_COP.1_(DH) (all listed domain parameter), and SSH protocol data from client and server as detailed in RFC 4253 [38] chapter 8*** in accordance with a specified key derivation algorithm ***SSH key derivation algorithm for encryption keys (AES) and integrity keys (HMAC)*** and specified cryptographic key sizes

***a) 128, 192 or 256 bit (for AES keys)***
***b) 256 or 512 bit (for HMAC keys)***

that meet the following: ***SSH key derivation according to RFC 4253 [38] chapter 7.2, with hash algorithms SHA-256, SHA-384, SHA-512 according to FCS_COP.1_(Hash)***.

## FCS_CKM.5_(TLS) Cryptographic key derivation

**FCS_CKM.5.1_(TLS)** The TSF shall derive cryptographic keys

***a) AES encryption keys***
***b) HMAC integrity protection keys***

from ***ephemeral shared secret as agreed for TLS according to FCS_COP.1_(ECDH) (ECDH domain parameter NIST P-256, P-384, brainpoolP384r1 or brainpoolP512r1), exchanged random data and TLS protocol data from client and server as detailed in RFC 8446 [32] chapter 7***, in accordance with a specified key derivation algorithm ***TLS key derivation algorithm for encryption keys (AES) and MAC keys (HMAC)*** and specified cryptographic key sizes

***a) 128 or 256 bit (for AES keys)***
***b) 256 or 384 bit (for HMAC keys)***

that meet the following:

- ***TLS key derivation according to RFC 8446 [32] chapter 7.1, with HKDF as specified in RFC 5869 [22] based on HMAC according to FCS_COP.1_(HMAC), with hash algorithms SHA-256 (for AES keys of 128 bit key size or HMAC keys of 256 bit key size) or SHA-384 (for AES keys of 256 bit key size or HMAC keys of 384 bit key size) according to FCS_COP.1_(Hash), and with random number generation according to FCS_RNG.1_(DRT.1).***

## FCS_CKM.5_(IPsec) Cryptographic key derivation

**FCS_CKM.5.1_(IPsec)** The TSF shall derive cryptographic keys

***a) AES encryption keys***
***b) HMAC integrity protection keys***

from ***ephemeral shared secret as agreed for IPsec according to FCS_COP.1_(ECDH) (ECDH domain parameter brainpoolP512r1), exchanged nonces and IPsec protocol data (Security Parameter Indexes, SPIs) from initiator and responder as detailed in RFC 4306 [17]*** in accordance with a specified key derivation algorithm ***IPsec key derivation algorithm for encryption keys (AES) and integrity protection keys (HMAC)*** and specified cryptographic key sizes

***a) 256 bit (for AES keys)***
***b) 512 bit (for HMAC keys)***

that meet the following:

- ***IPsec key derivation according to RFC 4306 [17] chapter 2, with pseudo-random function (PRF) based on HMAC according to FCS_COP.1_(HMAC), with hash algorithm SHA-512 according to FCS_COP.1_(Hash), and with random number generation according to FCS_RNG.1_(DRT.1).***

## FCS_CKM.5_(IPsecAuth)Cryptographic key derivation

**FCS_CKM.5.1_(IPsecAuth)** The TSF shall derive cryptographic keys ***HMAC authentication key*** from ***a pre-shared key (PSK) as generated for IPsec according to FCS_CKM.1_(PSK), and fixed string as detailed in RFC 4306 [17] chapter 2.15*** in accordance with a specified key derivation algorithm ***IPsec key derivation algorithm for authentication keys*** and specified cryptographic key sizes ***512 bit*** that meet the following:

- ***IPsec key derivation for authentication keys from pre-shared key according to RFC 4306 [17] chapter 2.15, with pseudo-random function (PRF) based on HMAC according to FCS_COP.1_(HMAC), with hash algorithm SHA-512 according to FCS_COP.1_(Hash).***

## FCS_CKM.6 Timing and event of cryptographic key destruction

**FCS_CKM.6.1** The TSF shall destroy ***all cryptographic keys generated according to FCS_CKM.1 (all iterations) or derived according to FCS_CKM.5 (all iterations), including all intermediate secret key values, and all DRNG entropy input, seed input, and internal state of the DRNG*** when *no longer needed*.

**FCS_CKM.6.2** The TSF shall destroy cryptographic keys and keying material specified by FCS_CKM.6.1 in accordance with a specified cryptographic key destruction method ***deletion*** that meets the following:

***a) overwriting the key by zeroizing in case of ephemeral plaintext secret or private keys or intermediate secret key values;***
***b) logical deletion in case of public keys or permanently stored secret or private keys.***

#### 6.1.2.2 Cryptographic operation (FCS_COP)

##### FCS_COP.1_(SWVerify) Cryptographic operation

**FCS_COP.1.1_(SWVerify)** The TSF shall perform ***signature verification for the integrity check of software update packages*** in accordance with a specified cryptographic algorithm ***RSA or ECDSA signature verification*** and cryptographic key sizes ***4096 bit (for RSA signatures) or 256 bit (for ECDSA signatures)*** that meet the following:

***a) RSA signature verification according to PKCS #1 v2.2 [24] using RSASSA-PKCS1-v1_5, and with hash algorithm SHA-512 (default), SHA-384 or SHA-256 according to FCS_COP.1_(Hash);***
***b) ECDSA signature verification according to FIPS 186-5 [29] chapter 6, with signature keys based on ECC domain parameters NIST curve P-256 according to NIST SP 800-186 [30], chapter 3.2.1, and with hash algorithm SHA-256 according to FCS_COP.1_(Hash).***

##### FCS_COP.1_(Hash) Cryptographic operation

**FCS_COP.1.1_(Hash)** The TSF shall perform ***hash value calculation*** in accordance with a specified cryptographic algorithm ***SHA-256, SHA-384, SHA-512*** and cryptographic key sizes ***none*** that meet the following: ***FIPS 180-4 [27] chapter 6***.

##### FCS_COP.1_(HMAC) Cryptographic operation

**FCS_COP.1.1_(HMAC)** The TSF shall perform ***HMAC calculation and verification*** in accordance with a specified cryptographic algorithm ***HMAC*** and cryptographic key sizes ***256 bit, 384 bit and 512 bit*** that meet the following: ***RFC 2104 [21], with hash algorithm SHA-256, SHA-384 or SHA-512 according to FCS_COP.1_(Hash)***.

##### FCS_COP.1_(AES) Cryptographic operation

**FCS_COP.1.1_(AES)** The TSF shall perform ***encryption and decryption*** in accordance with a specified cryptographic algorithm ***AES block cipher in CBC mode, CTR mode or GCM mode*** and cryptographic key sizes ***128 bit, 192 bit and 256 bit (however 128 bit is only possible in GCM or CTR mode, and 192 bit is only possible in CTR mode)*** that meet the following: ***FIPS 197 [28] (for AES block cipher), and with applicable standard for the block cipher mode:***

***a) CBC mode: NIST SP 800-38A [25] chapter 6.2;***
***b) CTR mode: NIST SP 800-38A [25] chapter 6.5;***

c) ***GCM mode: NIST SP 800-38D [26].***

### FCS_COP.1_(RSA) Cryptographic operation

**FCS_COP.1.1_(RSA)** The TSF shall perform ***signature generation and verification*** in accordance with a specified cryptographic algorithm ***RSA signature schemes RSASSA-PKCS-v1_5 or RSASSA-PSS (however RSASSA-PKCS-v1_5 is only possible for signature verification)*** and cryptographic key sizes ***minimum 3072 bit*** that meet the following: ***PKCS #1 [24] chapters 8.1, RSASSA-PSS, or 8.2, RSASSA-PKCS-v1_5, with hash algorithm SHA-256, SHA-384 or SHA-512 according to FCS_COP.1_(Hash)***.

### FCS_COP.1_(ECDSA) Cryptographic operation

**FCS_COP.1.1_(ECDSA)** The TSF shall perform ***signature generation and verification*** in accordance with a specified cryptographic algorithm ***ECDSA signature scheme*** and cryptographic key sizes ***of minimum 256 bit*** that meet the following: ***signature generation and verification according to FIPS 186-5 [29] chapter 6, with signature keys based on ECC domain parameters***

a) ***NIST curves P-256, P-384, P-521 according to NIST SP 800-186 [30], chapter 3.2.1;***

***with hash algorithm SHA-256, SHA-384 or SHA-512 according to FCS_COP.1_(Hash) (however SHA-384 is only possible with P-384 curve and SHA-512 only with curve P-521), and with random number generation according to FCS_RNG.1_(DRT.1).***

### FCS_COP.1_(ECDH) Cryptographic operation

**FCS_COP.1.1_(ECDH)** The TSF shall perform ***shared secret agreement*** in accordance with a specified cryptographic algorithm ***Elliptic Curve Diffie-Hellman (ECDH) shared secret agreement*** and cryptographic key sizes ***of minimum 256 bit*** that meet the following: ***shared secret agreement ECDH with cofactor according to RFC 5656 [34], chapter 4, and SEC 1 [33], chapter 3.3.2, with ECDH keys based on ECC domain parameters***

a) ***NIST curves P-256, P-384, P-521 according to NIST SP 800-186 [30] chapter 3.2.1;***
b) ***Brainpool curves brainpoolP256r1, brainpoolP384r1 and brainpoolP512r1 according to RFC 5639 [23], chapter 3.***

### FCS_COP.1_(DH) Cryptographic operation

**FCS_COP.1.1_(DH)** The TSF shall perform ***shared secret agreement*** in accordance with a specified cryptographic algorithm ***Diffie-Hellman (DH) shared secret agreement*** and cryptographic key sizes ***4096 or 8192 bit*** that meet the following: ***shared secret agreement according to RFC 4253 [38] chapter 8, with DH keys based on DH domain parameters***

a) ***4096 bit MODP group16 according to RFC 3526 [20], chapter 5;***
b) ***8192 bit MODP group18 according to RFC 3526 [20], chapter 7.***

#### 6.1.2.3 Generation of random numbers (FCS_RNG)

The selections and assignments in this SFR were taken from AIS 20/31 [31] with the goal to be compliant with class DRT.1. Therefore the SFR text contains nested assignments and selections without extra markup.

### FCS_RNG.1_(DRT.1) Random number generation (Class DRT.1)

**FCS_RNG.1.1_(DRT.1)** The TSF shall provide a *deterministic* random number generator that implements:

- **(DRT.1.1)** ***The TSF shall implement a DRNG tree that consists of $\geq$ 1 DRNGs, together with the relation "is direct seed predecessor". One of these DRNGs, called root DRNG, shall be distinguished and remain distinguished during the lifetime of the DRNG tree. During the lifetime of the DRNG tree, all DRNGs can be instantiated and uninstantiated. The root DRNG shall not be replaced during the lifetime of the DRNG tree.***
- **(DRT.1.2)** ***The RNG that generates seed material for the root DRNG ("initial randomness source") and all DRNGs of the DRNG tree shall be implemented and operated inside the same security boundary.***
- **(DRT.1.3)** ***The seed material for the root DRNG (for the seeding procedure and reseeding procedures) shall be generated by an NPTRNG that generates random bits with a min-entropy of more than 0,997 bits per bit.***
- **(DRT.1.4)** ***For each DRNG in the DRNG tree, the length of a single request shall comprise at most $2^{19}$ bits.***
- **(DRT.1.5)** ***For each DRNG in the DRNG tree, the effective internal state shall comprise at least 246 bits.***
- **(DRT.1.6)** ***The seed material for the root DRNG shall be generated by a TRNG, and the initial effective internal state (after the seeding procedure or the reseeding procedure) shall have min-entropy $\geq$ 240 bits.***
- **(DRT.1.7)** ***For each DRNG in the DRNG tree, except the root DRNG, the seed material shall be generated by a DRNG in the DRNG tree, and the following conditions shall be fulfilled: The seed material shall comprise $\geq$ 256 bits. If the seed material had $\geq$ 250 bits of min-entropy, then the initial effective internal state (after the seeding procedure or the reseeding procedure) would have min-entropy $\geq$ 240 bits.***
- **(DRT.1.8)** ***Each DRNG in the DRNG tree shall use the received seed material only for the seeding procedure or the reseeding procedure.***
- **(DRT.1.9)** ***For each DRNG in the DRNG tree, except the root DRNG, the seed material for reseeding procedures shall be generated by the same DRNG that has generated the seed material for the seeding procedure, i. e., by the direct seed predecessor (parent node in the seed graph), or, alternatively, by a sibling DRNG of the direct seed predecessor.***
- **(DRT.1.10)** ***Each DRNG in the DRNG tree shall provide forward secrecy on the granularity level of internal random numbers.***
- **(DRT.1.11)** ***Each DRNG in the DRNG tree shall provide backward secrecy on the granularity level of internal random numbers.***
- **(DRT.1.12)** ***Each DRNG in the DRNG tree shall either provide enhanced backward secrecy on the granularity level of internal random numbers, or enhanced backward secrecy on the granularity level of requests and the requests shall satisfy the condition of conceptual atomicity.***
- **(DRT.1.13)** ***For each DRNG in the DRNG tree (if applicable), additional input shall not weaken the strength of the DRNG even if an adversary is able to control the additional input.***
- **(DRT.1.14)** ***For each DRNG in the DRNG tree, the state transition function $\phi$ and the output function $\psi$ shall be cryptographic. The state transition function***

***shall be a one-way function.***

**FCS_RNG.1.2_(DRT.1)** The TSF shall provide *octets of bits* that meet

**(DRT.1.15)** ***For each DRNG in the DRNG tree, the internal random numbers shall have statistical inconspicuousness.***

### 6.1.3 Class FDP: User data protection

#### 6.1.3.1 Access control policy (FDP_ACC)

##### FDP_ACC.1 Subset access control

**FDP_ACC.1.1** The TSF shall enforce the ***Administrative Access Control SFP*** on

***a) subjects: all;***

***b) objects:***

- ***users;***
- ***network configuration;***
- ***relay configuration;***
- ***DNS server configuration;***
- ***mail server configuration;***
- ***packet filter rules;***
- ***Squid web proxy configuration;***
- ***virus scanner configuration;***
- ***SNMP server configuration;***
- ***audit configuration;***
- ***audit records;***

***c) operation: all.***

#### 6.1.3.2 Access control functions (FDP_ACF)

##### FDP_ACF.1 Security attribute-based access control

**FDP_ACF.1.1** The TSF shall enforce the ***Administrative Access Control SFP*** to objects based on the following:

***a) whether the subject is authenticated for the role of an Administrator or an Auditor, and the authentication is still valid;***

***b) in case of a command request via SSH: whether the authenticated user's permission flag for access via SSH is set;***

***c) whether the operation request originates from the administration network or from a local console or locally connected keyboard;***

***d) in case of a remote command request: IP and TCP header information of the network packets of the command request.***

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

***a) Only subjects with a valid authentication for the role of an Administrator are allowed to change or delete any of the configuration files covered by this SFP; add, modify or delete a user; delete audit log records;***

***b) Only subjects with a valid authentication for the role of an Administrator or Auditor are allowed to query any of the configuration files covered by this SFP; query the user list and user attributes; query audit log records;***

***c) If the command request for the operation comes from remote:***

- ***The Filter Rules SFP is applied and all checks have passed.***

- ***The request comes from the administration network.***

**FDP_ACF.1.3** The TSF shall explicitly authorize 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:

- ***there is no valid authentication for any of the roles;***
- ***the command request for the operation is neither a local request (from a local console or locally connected keyboard) nor does it originate from the administration network;***
- ***the command request for the operation comes via SSH but the respective user's permission flag for access via SSH is not set;***
- ***every operation request that is not explicitly allowed is denied.***

#### 6.1.3.3 Information flow control policy (FDP_IFC)

##### FDP_IFC.1 Subset information flow control

**FDP_IFC.1.1** The TSF shall enforce the ***Filter Rules SFP*** on

***a) subjects: users that send and receive information through the TOE to one another;***
***b) information: data traffic sent through the TOE from one subject to another;***
***c) operation: pass information.***

#### 6.1.3.4 Information flow control functions (FDP_IFF)

##### FDP_IFF.1 Simple security attributes

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

- ***The header information of network packets, depending on their type:***
  ***a) TCP: IP and TCP header;***
  ***b) UDP: IP and UDP header;***
  ***c) ICMP: IP header and ICMP message;***
  ***d) IGMP: IP header and IGMP message;***
  ***e) IP: IP header;***
- ***The actual date and time.***
- ***The interfaces from which the packets are received and to which they are delivered.***
- ***Additional information depending on the handling relay:***
  ***a) iprelay: none;***
  ***b) pingrelay: none;***
  ***c) udprelay: if the protocol conformance filter is active: protocol and/or application data;***
  ***d) tcprelay: if the protocol conformance filter is active: protocol and/or application data;***
  ***e) smtprelay: protocol and application data;***
  ***f) smtp2smtprelay: protocol and application data;***
  ***g) imaprelay: protocol and application data;***
  ***h) poprelay: protocol and application data;***
  ***i) siprelay: protocol and application data;***
  ***j) sshrelay: protocol data;***

*k) wwwrelay: protocol and application data;*
*l) httprelay: protocol and application data;*
*m) ftprelay: protocol data;*
*n) mcastudprelay: IGMP and multicast UDP packets;*

**Application Note**: The list only considers the relays, but not the meta-policies because they are convenience methods to easily configure services.

**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:

- ***IP spoofing check pass.***
- ***IP option check pass.***
- ***The connection is configured, and, where relevant and if configured, the user can be successfully authenticated for the respective protocol:***
  *a) iprelay: source and destination IP address and protocol are allowed;*
  *b) pingrelay: source and destination IP address are allowed;*
  *c) udprelay: source and destination IP address and port are allowed;*
  *d) tcprelay: source and destination IP address and port are allowed;*
  *e) smtprelay: source and destination IP address and port are allowed. The user can be successfully authenticated by the given authentication data for smtp (if configured).*
  *f) ftprelay: source and destination IP address and port are allowed. The user can be successfully authenticated by the given authentication data for ftp.*
  *g) sshrelay: source and destination IP address and port are allowed. The user can be successfully authenticated by the given authentication data for ssh.*
  *h) mcastudprelay: packets of the respective multicast group are allowed;*
  *i) all other relays: source and destination IP address and port are allowed.*
- ***The packet filter rules pass.***
- ***All ACL checks for the respective relay pass.***

**FDP_IFF.1.3** The TSF shall enforce the ***none***.

**FDP_IFF.1.4** The TSF shall explicitly authorize 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: ***The protocol data is filtered:***

*a) iprelay: none.*
*b) pingrelay: none.*
*c) udprelay: none.*
*d) tcprelay: none.*
*e) smtprelay: depending on the configuration: configured checks for mail sender and recipient, greylisting, mail relay lead to the rejection of mail. The message is also discarded if the user authentication check failed (if authentication check is activated for this protocol).*
*E-mail contents of content type text/html can be filtered for active contents, if configured. A virus scanner can check the application data. MIME-encoded e-mails are (recursively) parsed and their parts are checked like non-encoded e-mails.*
*f) smtp2smtprelay: E-mail contents of content type text/html can be filtered for active contents, if configured. A virus scanner can check the application data. MIME-encoded e-mails are (recursively) parsed and their parts are checked like non-encoded e-mails.*
*g) imaprelay: the ACL and request method checks fail. A virus scanner can check the application data.*

h) ***poprelay: configurable protocol elements from the client are discarded. Application data of content type text/html can be filtered for active contents, if configured. A virus scanner can check the application data. MIME-encoded messages are (recursively) parsed and their parts are checked like non-encoded messages.***
i) ***siprelay: the tests for the configured internal and external domains and RTP port ranges fail. The ACL and request method checks fail.***
j) ***sshrelay: depending on the configuration: a subset of SSH protocol messages can be filtered out of the connection. The message is also discarded if the user authentication check failed (if authentication check is activated for this protocol).***
k) ***wwwrelay: configurable protocol elements from the client or server are discarded; configurable cookies are filtered. The application data is filtered. Server replies of content type text/html can be filtered for active contents, if configured. A virus scanner can check the application data. MIME-encoded replies are (recursively) parsed and their parts are checked like non-encoded contents.***
l) ***httprelay: the ACL and protocol (HTTP/Websockets) checks fail. The XML validation of the application data fails when using the webservice policy.***
m) ***ftprelay: depending on the configuration: configurable protocol elements from the client are discarded. The message is also discarded if the user authentication check failed (if authentication check is activated for this protocol).***
n) ***mcastudprelay: none***
o) ***all relays: An authenticated Administrator can explicitly terminate an existing connection.***
p) ***Every packet that cannot be explicitly assigned to a permitting rule is dropped. This includes the situation if no appropriate relay is available that can control the information flow.***

**Application Note**: The list only considers the relays, but not the meta-policies because they are convenience methods to easily configure services. However, the httprelay cannot be configured itself, so in that case the meta-policies have to be considered.

**Application Note**: The http-, imap-, pop-, sip-, and smtp2smtprelay do not allow authentication at the TOE even if the respective protocols support authentication.

### 6.1.4 Class FIA: Identification and authentication

#### 6.1.4.1 User identification (FIA_UID)

**FIA_UID.1 Timing of identification**

**FIA_UID.1.1** The TSF shall allow

a) ***Self-test during initial start-up according to FPT_SST.1EX;***
b) ***sending and receiving information through the TOE, in accordance with the Filter Rules SFP;***
c) ***usage of administrative commands where no user authentication is needed, including requests for the status of the TOE***

on behalf of the user to be performed before the user is identified.

**FIA_UID.1.2** The TSF shall require each user to be successfully identified before allowing any **other** TSF-mediated actions on behalf of that user.

### 6.1.4.2 User authentication (FIA_UAU)

#### FIA_UAU.1 Timing of authentication

**FIA_UAU.1.1** The TSF shall allow

a) ***Self-test during initial start-up according to FPT_SST.1EX;***
b) ***sending and receiving information through the TOE, in accordance with the Filter Rules SFP;***
c) ***Identification of the user by means of TSF required by FIA_UID.1;***
d) ***usage of administrative commands where no user authentication is needed, including requests for the status of the TOE;***

on behalf of the user to be performed before the user is authenticated.

**FIA_UAU.1.2** The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

#### FIA_UAU.5 Multiple authentication mechanisms

**FIA_UAU.5.1** The TSF shall provide ***password-based (local password, password via TLS, SSH with password, LDAP, RADIUS), SSH with public key, and TLS client certificate authentication mechanisms*** to support user authentication **by internal or external means**.

**FIA_UAU.5.2** The TSF shall authenticate any user's claimed identity according to the ***following list:***

a) ***authentication for remote administration (Administrator, Auditor):***
   - ***password-based mechanisms (password via TLS, SSH with password, password via LDAP, password via RADIUS);***
   - ***SSH with public key;***
   - ***TLS client certificate: This authentication mechanism is not sufficient as an authentication mechanisms in its own but is exclusively used as second factor for Two-Factor-Authentication (2FA): 2FA with a TLS client certificate is possible if one of the password-based authentication mechanisms password via TLS, LDAP or RADIUS is used;***
b) ***authentication for local administration (Administrator, Auditor): local password. Authentication will only be accepted if the request comes from a directly connected serial console or keyboard/monitor.***

#### FIA_UAU.6 Re-authenticating

**FIA_UAU.6.1** The TSF shall re-authenticate the user under the conditions ***timeout after inactivity (time period can be configured by an Administrator)***.

#### FIA_UAU.7 Protected authentication feedback

**FIA_UAU.7.1** The TSF shall provide only ***no feedback or neutral feedback per character*** to the user while the authentication is in progress.

### 6.1.4.3 User attribute definition (FIA_ATD)

#### FIA_ATD.1 User attribute definition

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

***a) administrative role (Administrator or Auditor or none);***
***b) permission flag for access via SSH***
***c) user password (if configured);***
***d) user SSH public key (if configured; one of either password or SSH public key is mandatory);***
***e) user's TLS client certificate (if configured)***

#### 6.1.4.4 Specification of secrets (FIA_SOS)

##### FIA_SOS.1 Verification of secrets

**FIA_SOS.1.1** The TSF shall provide a mechanism to verify that secrets meet ***the following metric:***

- ***the user password length is between 10 and 128 characters***
- ***the user password contains at least one number***
- ***the user password contains at least one lower case letter***
- ***the user password contains at least one upper case letter***
- ***the user password contains at least one symbol***
- ***the user password does not contain any of the words "root", "admin", "genugate" or "superuser"***

**Application Note**: This SFR does not apply to authentication at an external RADIUS or LDAP server, because in these two cases the passwords are configured at the external servers.

#### 6.1.4.5 Authentication failures (FIA_AFL)

##### FIA_AFL.1 Authentication failure handling

**FIA_AFL.1.1** The TSF shall detect when *an administrator configurable positive integer within* ***1 to infinite (default 5)*** unsuccessful authentication attempts occur related to ***consecutive failed authentication attempts for remote administration with password-based authentication mechanism (password via TLS, SSH with password, password via LDAP or password via RADIUS)***.

**FIA_AFL.1.2** When the defined number of unsuccessful authentication attempts has been *surpassed*, the TSF shall ***prevent the offending user from successfully authenticating until an authorized Administrator unblocks the user in question***.

### 6.1.5 Class FMT: Security management

#### 6.1.5.1 Security management roles (FMT_SMR)

##### FMT_SMR.1 Security roles

**FMT_SMR.1.1** The TSF shall maintain the roles ***Administrator and Auditor***.

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

#### 6.1.5.2 Specification of Management Functions (FMT_SMF)

##### FMT_SMF.1 Specification of Management Functions

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

***a) user configuration;***
***b) timeouts configuration;***
***c) network configuration;***
***d) relay configuration;***
***e) DNS server configuration;***
***f) mail server configuration;***
***g) packet filter rule configuration;***
***h) Squid web proxy configuration;***
***i) virus scanner configuration;***
***j) audit log configuration;***
***k) SNMP server configuration;***
***l) verify the current patch level;***
***m) query audit log;***
***n) audit log management (export to external audit server if configured, delete audit records).***

#### 6.1.5.3 Management of TSF data (FMT_MTD)

##### FMT_MTD.1_(Modify) Management of TSF data

**FMT_MTD.1.1_(Modify)** The TSF shall restrict the ability to *modify, delete,* **create** the

***a) users;***
***b) timeouts configuration;***
***c) network configuration;***
***d) relay configuration;***
***e) DNS server configuration;***
***f) mail server configuration;***
***g) packet filter rules;***
***h) Squid web proxy configuration;***
***i) virus scanner configuration;***
***j) audit log configuration;***
***k) SNMP server configuration;***
***l) (only deletion is possible) audit log;***

to ***the Administrator***.

##### FMT_MTD.1_(Query) Management of TSF data

**FMT_MTD.1.1_(Query)** The TSF shall restrict the ability to *query* the

***a) user list and user configuration;***
***b) timeouts configuration;***
***c) network configuration;***
***d) relay configuration;***
***e) DNS server configuration;***
***f) mail server configuration;***
***g) packet filter rules;***
***h) Squid web proxy configuration;***
***i) virus scanner configuration;***

*j) audit log configuration;*
*k) SNMP server configuration;*
*l) audit log;*

to ***the Administrator and Auditor***.

#### 6.1.5.4 Management of functions in TSF (FMT_MOF)

##### FMT_MOF.1 Management of security functions behaviour

**FMT_MOF.1.1** The TSF shall restrict the ability to **perform**~~[selection: determine the behaviour of, disable, enable, modify the behaviour of]~~ the functions

*a) start-up and shutdown;*
*b) change to maintenance and normal operation mode;*
*c) apply software patches*

to ***the Administrator***.

**Application Note**: This SFR uses a refinement instead of a selection, because the security function behavior is not changed by setting attributes but by performing maintenance actions.

#### 6.1.5.5 Management of security attributes (FMT_MSA)

##### FMT_MSA.1_(Modify) Management of security attributes

**FMT_MSA.1.1_(Modify)** The TSF shall enforce the ***Administrative Access Control SFP*** to restrict the ability to *modify* the security attributes

*a) the user's administrative role,*
*b) the user's permission flag for access via SSH,*
*c) the user's password (check value),*
*d) the user's SSH public key,*
*e) the user's TLS client certificate (check value),*
*f) the CA certificates for TLS client certificate verification*

to ***the Administrator***.

##### FMT_MSA.1_(Query) Management of security attributes

**FMT_MSA.1.1_(Query)** The TSF shall enforce the ***Administrative Access Control SFP*** to restrict the ability to *query* the security attributes

*a) the user's administrative role,*
*b) the user's permission flag for access via SSH,*
*c) the user's SSH public key,*
*d) the user's TLS client certificate (check value),*
*e) the CA certificates for TLS client certificate verification (check value)*

to ***the Administrator and the Auditor***.

##### FMT_MSA.1_(NoQuery) Management of security attributes

**FMT_MSA.1.1_(NoQuery)** The TSF shall enforce the ***Administrative Access Control SFP*** to restrict the ability to *query* the security attributes

*a) the user's password;*

to ***none***.

#### FMT_MSA.3 Static attribute initialization

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

**FMT_MSA3.2** The TSF shall allow the ***Administrator*** to specify alternative initial values to override the default values when an object or information is created.

### 6.1.6 Class FPT: Protection of the TSF

#### 6.1.6.1 Trusted recovery (FPT_RCV)

#### FPT_RCV.2 Automated recovery

**FPT_RCV.2.1** When automated recovery from ***a failure or service discontinuity*** is not possible, the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

**FPT_RCV.2.2** For ***configurable events (default: none)***, the TSF shall ensure the return of the TOE to a secure state using automated procedures.

#### 6.1.6.2 Simple Self Test (FPT_SST)

#### FPT_SST.1EX TOE testing

**FPT_SST.1EX.1** The TSF shall run a suite of self tests *during initial start-up, periodically during normal operation and at the conditions* ***software update*** to perform the following checks:

***a) during initial start-up:***
- ***Software authenticity and integrity test including all configuration files and crypto libraries;***

***b) periodically during normal operation:***
- ***specified processes are running (default: all relays, daemons, web servers);***
- ***the file system usage is below a threshold (default: 90 %);***
- ***the OpenBSD file system permissions and flags;***

***c) at software update:***
- ***software update is intended for the current software version;***
- ***correct patch level;***
- ***Software authenticity and integrity test;***

**FPT_SST.1EX.2** The TSF shall provide authorized users with the capability to query the results of the following checks:

***a) specified processes are running (default: all relays, daemons, web servers);***
***b) the file system usage is below a threshold (default: 90 %);***
***c) the OpenBSD file system permissions and flags.***

#### 6.1.6.3 Fail secure (FPT_FLS)

#### 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:

***a) Self-test according to FPT_SST.1EX.1 fails.***

#### 6.1.6.4 Time stamps (FPT_STM)

##### FPT_STM.1 Reliable time stamps

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

**Application Note**: The reliability is realized by synchronizing the real time clock with one or more NTP time servers using the protocol NTP version 4.

#### 6.1.6.5 Internal TOE TSF data replication consistency (FPT_TRC)

##### FPT_TRC.1 Internal TSF consistency

**FPT_TRC.1.1** The TSF shall ensure that TSF data is consistent when replicated between parts of the TOE.

**FPT_TRC.1.2** When parts of the TOE containing replicated TSF data are disconnected, the TSF shall ensure the consistency of the replicated TSF data upon reconnection before processing any requests for ***services provided by the Filter Rules SFP***.

**Application Note**: The systems use an internal revision number to check the configuration. They only reactivate services when their configuration is up to date. The new configuration is used only for new connections, existing connections are not reconfigured.

#### 6.1.6.6 TOE Update (FPT_UPD)

##### FPT_UPD.1EX Trusted Update

**FPT_UPD.1EX.1** The TOE shall cryptographically verify additional code/patches to itself using a digital signature prior to installation using schemes specified in ***FCS_COP.1_(SWVerify)***.

**FPT_UPD.1EX.2** A modification of the TOE shall only be allowed if the software update

- is intended for the current software version,
- has the correct patch level and
- has been cryptographically verified with regard to integrity and authenticity.

##### FPT_UPD.2EX Update identification data

**FPT_UPD.2EX.1** The TSF shall verify if the activation of the patch and the update of the identification data have been both completed.

**FPT_UPD.2EX.2** The TSF shall update the active identification data when the patch is applied in order to keep the system in a defined state.

**FPT_UPD.2EX.3** The TSF shall use the maintenance mode to activate the final TOE.

### 6.1.7 Class FTP: Trusted path/channels

#### 6.1.7.1 Trusted path (FTP_TRP)

##### FTP_TRP.1_(TLS) Trusted path

**FTP_TRP.1.1_(TLS)** The TSF shall provide a communication path **TLS** 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 and disclosure*.

**FTP_TRP.1.2_(TLS)** The TSF shall permit *remote users* to initiate communication via the trusted path.

**FTP_TRP.1.3_(TLS)** The TSF shall require the use of the trusted path for *initial user authentication and* ***execution of administrative tasks from remote via TLS (by using Web-GUI or REST-API)***.

#### FTP_TRP.1_(SSH) Trusted path

**FTP_TRP.1.1_(SSH)** The TSF shall provide a communication path **SSH** 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 and disclosure*.

**FTP_TRP.1.2_(SSH)** The TSF shall permit *remote users* to initiate communication via the trusted path.

**FTP_TRP.1.3_(SSH)** The TSF shall require the use of the trusted path for *initial user authentication and* ***execution of administrative tasks from remote via SSH***.

#### FTP_TRP.1_(Local) Trusted path

**FTP_TRP.1.1_(Local)** The TSF shall provide a **local** communication path between itself and *local* 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 and disclosure*.

**FTP_TRP.1.2_(Local)** The TSF shall permit *local users* to initiate communication via the trusted path.

**FTP_TRP.1.3_(Local)** The TSF shall require the use of the trusted path for *local initial user authentication and* ***local execution of administrative tasks***.

### 6.1.7.2 Inter-TSF trusted channel (FTP_ITC)

#### FTP_ITC.1_(IPsec) Inter-TSF trusted channel

**FTP_ITC.1.1_(IPsec)** The TSF shall provide a communication channel **IPsec** between itself and another ~~trusted IT product~~ **TOE instance that is part of the configured separate HA network** 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_(IPsec)** The TSF shall permit *the TSF* **or the TSF of another TOE instance that is part of the configured separate HA network** to initiate communication via the trusted channel.

**FTP_ITC.1.3_(IPsec)** The TSF shall initiate communication via the trusted channel for ***communication between HA peers, i. e. communication and synchronization between two instances of the TOE via IPsec for the purpose of high availability***.

## 6.2 Security Assurance Requirements

In order to handle patch management, the Security Target defines one new assurance component for the class ALC: Life-cycle support, defined in chapter 5.2.1.

Table 4 shows the Security Assurance Requirements for the level EAL4. The augmented components ALC_FLR.2, ASE_TSS.2 and AVA_VAN.5 are set in a bold font. For the level EAL4, the SARs ADV_INT and ADV_SPM are not needed.

The table also contains the new assurance component ALC_PAM.1.

Table 4: SAR

| Class | Family | Level | Name |
|---|---|---|---|
| Development | ADV_ARC | ADV_ARC.1 | Security architecture description |
| | ADV_FSP | ADV_FSP.4 | Complete functional specification |
| | ADV_IMP | ADV_IMP.1 | Implementation representation of the TSF |
| | ADV_INT | | TSF internals |
| | ADV_SPM | | Security policy modelling |
| | ADV_TDS | ADV_TDS.3 | Basic modular design |
| Guidance | AGD_OPE | AGD_OPE.1 | Operational user guidance |
| | AGD_PRE | AGD_PRE.1 | Preparative procedures |
| Life-cycle | ALC_CMC | ALC_CMC.4 | Production support, acceptance procedures and automation |
| | ALC_CMS | ALC_CMS.4 | Problem tracking CM coverage |
| | ALC_DEL | ALC_DEL.1 | Delivery procedures |
| | ALC_DVS | ALC_DVS.1 | Identification of security measures |
| | ALC_FLR | **ALC_FLR.2** | Flaw reporting procedures |
| | ALC_LCD | ALC_LCD.1 | Developer defined life-cycle model |
| | ALC_TAT | ALC_TAT.1 | Well-defined development tools |
| | ALC_PAM | **ALC_PAM.1** | Patch Management Processes |
| Security Target | ASE_CCL | ASE_CCL.1 | Conformance claims |
| | ASE_ECD | ASE_ECD.1 | Extended components definition |
| | ASE_INT | ASE_INT.1 | ST introduction |
| | ASE_OBJ | ASE_OBJ.2 | Security objectives |
| | ASE_REQ | ASE_REQ.2 | Derived security requirements |
| | ASE_SPD | ASE_SPD.1 | Security problem definition |
| | ASE_TSS | **ASE_TSS.2** | TOE summary specification with architectural design summary |
| Tests | ATE_COV | ATE_COV.2 | Analysis of coverage |
| | ATE_DPT | ATE_DPT.1 | Testing: basic design |
| | ATE_FUN | ATE_FUN.1 | Functional testing |
| | ATE_IND | ATE_IND.2 | Independent testing - sample |
| Vulnerability | AVA_VAN | **AVA_VAN.5** | Advanced methodical vulnerability analysis |

## 6.3 Security Functional Requirements Rationale

### 6.3.1 SFR Dependencies

Tables 5–11 show that all dependencies between SFRs are met.

Table 5: SFR Dependencies for class FAU

| SFR | Dependencies | Satisfied by |
|---|---|---|
| FAU_GEN.1 | FPT_STM.1 | FPT_STM.1 |
| FAU_SAA.1 | FAU_GEN.1 | FAU_GEN.1 |
| FAU_ARP.1 | FAU_SAA.1 | FAU_SAA.1 |
| FAU_SAR.1 | FAU_GEN.1 | FAU_GEN.1 |
| FAU_SAR.2 | FAU_SAR.1 | FAU_SAR.1 |
| FAU_SAR.3 | FAU_SAR.1 | FAU_SAR.1 |
| FAU_STG.1 | FAU_GEN.1 | FAU_GEN.1 |
| FAU_STG.2 | FAU_GEN.1 | FAU_GEN.1 |
| FAU_STG.4 | FAU_STG.2 | FAU_STG.2 |
| FAU_STG.5 | FAU_STG.2<br>FAU_GEN.1 | FAU_STG.2<br>FAU_GEN.1 |

All dependencies for the class FAU are fulfilled automatically, because there is only one possible resolution, which is in all instances the appropriate SFR.

Table 6: SFR Dependencies for class FCS

| SFR | Dependencies | Satisfied by |
|---|---|---|
| FCS_CKM.1_(RSA) | [FCS_CKM.2, or FCS_CKM.5, or FCS_COP.1]<br>[FCS_RBG.1, or FCS_RNG.1]<br>FCS_CKM.6 | FCS_COP.1_(RSA)<br><br>FCS_RNG.1_(DRT.1)<br>FCS_CKM.6 |
| FCS_CKM.1_(ECDSA) | [FCS_CKM.2, or FCS_CKM.5, or FCS_COP.1]<br>[FCS_RBG.1, or FCS_RNG.1]<br>FCS_CKM.6 | FCS_COP.1_(ECDSA)<br><br>FCS_RNG.1_(DRT.1)<br>FCS_CKM.6 |
| FCS_CKM.1_(PSK) | [FCS_CKM.2, or FCS_CKM.5, or FCS_COP.1]<br>[FCS_RBG.1, or FCS_RNG.1]<br>FCS_CKM.6 | FCS_CKM.2_(PSK) and FCS_CKM.5_(IPsecAuth)<br>FCS_RNG.1_(DRT.1)<br>FCS_CKM.6 |
| FCS_CKM.1_(DH) | [FCS_CKM.2, or FCS_CKM.5, or FCS_COP.1]<br>[FCS_RBG.1, or FCS_RNG.1]<br>FCS_CKM.6 | FCS_COP.1_(DH)<br><br>FCS_RNG.1_(DRT.1)<br>FCS_CKM.6 |
| FCS_CKM.1_(ECDH) | [FCS_CKM.2, or FCS_CKM.5, or FCS_COP.1]<br>[FCS_RBG.1, or FCS_RNG.1]<br>FCS_CKM.6 | FCS_COP.1_(ECDH)<br><br>FCS_RNG.1_(DRT.1)<br>FCS_CKM.6 |
| FCS_CKM.2_(PSK) | [FDP_ITC.1, or FDP_ITC.2, or FCS_CKM.1, or FCS_CKM.5] | FCS_CKM.1_(PSK) |
| FCS_CKM.5_(SSH) | [FCS_CKM.2, or FCS_COP.1]<br><br>FCS_CKM.6 | FCS_COP.1_(AES) and FCS_COP.1_(HMAC)<br>FCS_CKM.6 |
| FCS_CKM.5_(TLS) | [FCS_CKM.2, or FCS_COP.1]<br><br>FCS_CKM.6 | FCS_COP.1_(AES) and FCS_COP.1_(HMAC)<br>FCS_CKM.6 |
| FCS_CKM.5_(IPsec) | [FCS_CKM.2, or FCS_COP.1]<br><br>FCS_CKM.6 | FCS_COP.1_(AES) and FCS_COP.1_(HMAC)<br>FCS_CKM.6 |
| FCS_CKM.5_(IPsecAuth) | [FCS_CKM.2, or FCS_COP.1]<br>FCS_CKM.6 | FCS_COP.1_(HMAC)<br>FCS_CKM.6 |
| FCS_CKM.6 | [FDP_ITC.1, or FDP_ITC.2, or FCS_CKM.1, or FCS_CKM.5] | FCS_CKM.1 (all iterations) and FCS_CKM.5 (all iterations) |
| FCS_COP.1_(SWVerify) | [FDP_ITC.1, or FDP_ITC.2, or FCS_CKM.1, or FCS_CKM.5]<br><br>FCS_CKM.6 | N/A (the (public) key to verify the signature of the patch is distributed on the TOE installation medium that is secured by the delivery process in ALC_DEL, therefore no explicit import function is necessary)<br>FCS_CKM.6 |
| FCS_COP.1_(Hash) | [FDP_ITC.1, or FDP_ITC.2, or FCS_CKM.1, or FCS_CKM.5]<br>FCS_CKM.6 | N/A (no cryptographic key used)<br><br>FCS_CKM.6 |

Table 6: SFR Dependencies for class FCS (continued)

| SFR | Dependencies | Satisfied by |
|---|---|---|
| FCS_COP.1_(HMAC) | [FDP_ITC.1, or FDP_ITC.2, or FCS_CKM.1, or FCS_CKM.5]<br><br>FCS_CKM.6 | FCS_CKM.5_(SSH), FCS_CKM.5_(TLS), FCS_CKM.5_(IPsec) and FCS_CKM.5_(IPsecAuth)<br>FCS_CKM.6 |
| FCS_COP.1_(AES) | [FDP_ITC.1, or FDP_ITC.2, or FCS_CKM.1, or FCS_CKM.5]<br><br>FCS_CKM.6 | FCS_CKM.5_(SSH), FCS_CKM.5_(TLS) and FCS_CKM.5_(IPsec)<br>FCS_CKM.6 |
| FCS_COP.1_(RSA) | [FDP_ITC.1, or FDP_ITC.2, or FCS_CKM.1, or FCS_CKM.5]<br>FCS_CKM.6 | FCS_CKM.1_(RSA)<br><br>FCS_CKM.6 |
| FCS_COP.1_(ECDSA) | [FDP_ITC.1, or FDP_ITC.2, or FCS_CKM.1, or FCS_CKM.5]<br>FCS_CKM.6 | FCS_CKM.1_(ECDSA)<br><br>FCS_CKM.6 |
| FCS_COP.1_(ECDH) | [FDP_ITC.1, or FDP_ITC.2, or FCS_CKM.1, or FCS_CKM.5]<br>FCS_CKM.6 | FCS_CKM.1_(ECDH)<br><br>FCS_CKM.6 |
| FCS_COP.1_(DH) | [FDP_ITC.1, or FDP_ITC.2, or FCS_CKM.1, or FCS_CKM.5]<br>FCS_CKM.6 | FCS_CKM.1_(DH)<br><br>FCS_CKM.6 |
| FCS_RNG.1_(DRT.1) | (no dependencies) | |

In table 6 above, the errata in Common Criteria Part 2 [11] regarding SFR dependencies for class FCS as identified in the Common Criteria Errata and Interpretation for CC:2022 [15] are corrected in accordance with [15]. This concerns the SFR FCS_CKM.6 (where the optional dependency on FCS_CKM.5 is added in accordance with the Errata document) and all iterations of SFRs FCS_CKM.1, FCS_CKM.2 and FCS_COP.1 (where the dependency on FCS_CKM.3 is removed).

Given this, the dependencies for the class FCS are fulfilled as follows:

- All components that depend on either FCS_RBG.1 or FCS_RNG.1 are fulfilled by FCS_RNG.1_(DRT.1) which is the only appropriate choice.
- Most dependencies of FCS_CKM.1 on cryptographic operations are resolved by the respective FCS_COP.1 component. Only for FCS_CKM.1_(PSK) the dependency is fulfilled by FCS_CKM.2_(PSK) and FCS_CKM.5_(IPsecAuth).
- All dependencies on FCS_CKM.6 are automatically fulfilled by the only existing instance of that component.
- The dependency for FCS_CKM.2_(PSK) on cryptographic keys is fulfilled by FCS_CKM.1_(PSK).
- For all instances of FCS_CKM.5, the dependency on cryptographic operations is fulfilled by FCS_COP.1_(AES) and FCS_COP.1_(HMAC), with the exception of FCS_CKM.5_(IPsecAuth), where only FCS_COP.1_(HMAC) is applicable.
- For FCS_CKM.6 the dependencies are fulfilled by FCS_CKM.1 (all iterations) and FCS_CKM.5 (all iterations).
- For FCS_COP.1_(HMAC), the dependency on cryptographic keys is resolved by all applicable iterations of FCS_CKM.5, i. e. FCS_CKM.5_(SSH), FCS_CKM.5_(TLS), FCS_CKM.5_(IPsec) and FCS_CKM.5_(IPsecAuth).

- For FCS_COP.1_(AES), the dependency on cryptographic keys is resolved by some of the iterations of FCS_CKM.5, i. e. FCS_CKM.5_(SSH), FCS_CKM.5_(TLS) and FCS_CKM.5_(IPsec).
- For all further iterations of FCS_COP.1, the dependency on cryptographic keys is resolved by the respective FCS_CKM.1 key generation component.
- The component FCS_RNG.1_(DRT.1) has no dependencies so no rationale is needed.

Table 7: SFR Dependencies for class FDP

| SFR | Dependencies | Satisfied by |
|---|---|---|
| FDP_ACC.1 | FDP_ACF.1 | FDP_ACF.1 |
| FDP_ACF.1 | FDP_ACC.1<br>FMT_MSA.3 | FDP_ACC.1<br>FMT_MSA.3 |
| FDP_IFC.1 | FDP_IFF.1 | FDP_IFF.1 |
| FDP_IFF.1 | FDP_IFC.1<br>FMT_MSA.3 | FDP_IFC.1<br>N/A (the Filter Rules SFP does not depend on security attributes but on TSF data like the relay configuration (which is handled in FMT_MTD.1_(Modify) and FMT_MTD.1_(Query)), interfaces and header information, protocol and application data of the network packets, hence the dependency on FMT_MSA.3 is not applicable) |

Most dependencies for the class FDP are fulfilled automatically, because there is only one possible resolution, which is in all instances the appropriate SFR.

The dependency of FDP_IFF.1 on FMT_MSA.3 is not applicable, because the default depends on the actual TOE configuration, which is handled by FMT_MTD.1_(Modify) and FMT_MTD.1_(Query).

Table 8: SFR Dependencies for class FIA

| SFR | Dependencies | Satisfied by |
|---|---|---|
| FIA_UID.1 | (no dependencies) | |
| FIA_UAU.1 | FIA_UID.1 | FIA_UID.1 |
| FIA_UAU.5 | (no dependencies) | |
| FIA_UAU.6 | (no dependencies) | |
| FIA_UAU.7 | FIA_UAU.1 | FIA_UAU.1 |
| FIA_ATD.1 | (no dependencies) | |
| FIA_SOS.1 | (no dependencies) | |
| FIA_AFL.1 | FIA_UAU.1 | FIA_UAU.1 |

Most dependencies for the class FIA are fulfilled automatically, because there is only one possible resolution, which is in all instances the appropriate SFR.

Some components have no dependencies so no rationale is needed.

Table 9: SFR Dependencies for class FMT

| SFR | Dependencies | Satisfied by |
|---|---|---|
| FMT_SMR.1 | FIA_UID.1 | FIA_UID.1 |
| FMT_SMF.1 | (no dependencies) | |
| FMT_MTD.1_(Modify) | FMT_SMR.1<br>FMT_SMF.1 | FMT_SMR.1<br>FMT_SMF.1 |
| FMT_MTD.1_(Query) | FMT_SMR.1<br>FMT_SMF.1 | FMT_SMR.1<br>FMT_SMF.1 |
| FMT_MOF.1 | FMT_SMR.1<br>FMT_SMF.1 | FMT_SMR.1<br>FMT_SMF.1 |
| FMT_MSA.1_(Modify) | [FDP_ACC.1, or FDP_IFC.1]<br>FMT_SMR.1<br>FMT_SMF.1 | FDP_ACC.1<br>FMT_SMR.1<br>FMT_SMF.1 |
| FMT_MSA.1_(Query) | [FDP_ACC.1, or FDP_IFC.1]<br>FMT_SMR.1<br>FMT_SMF.1 | FDP_ACC.1<br>FMT_SMR.1<br>FMT_SMF.1 |
| FMT_MSA.1_(NoQuery) | [FDP_ACC.1, or FDP_IFC.1]<br>FMT_SMR.1<br>FMT_SMF.1 | FDP_ACC.1<br>FMT_SMR.1<br>FMT_SMF.1 |
| FMT_MSA.3 | FMT_MSA.1<br><br><br>FMT_SMR.1 | FMT_MSA.1_(Modify),<br>FMT_MSA.1_(Query) and<br>FMT_MSA.1_(NoQuery)<br>FMT_SMR.1 |

Most dependencies for the class FMT are fulfilled automatically, because there is only one possible resolution, which is in all instances the appropriate SFR.

The dependencies of FMT_MSA.1 (three instances) are in all cases fulfilled by FDP_ACC.1. The security attributes of FMT_MSA.1 in all instances are administrative attributes and therefore ruled by access control.

The dependency of FMT_MSA.3 is fulfilled by all three instances of FMT_MSA.1.

Table 10: SFR Dependencies for class FPT

| SFR | Dependencies | Satisfied by |
|---|---|---|
| FPT_RCV.2 | AGD_OPE.1 | AGD_OPE.1 |
| FPT_SST.1EX | (no dependencies) | |
| FPT_FLS.1 | (no dependencies) | |
| FPT_STM.1 | (no dependencies) | |
| FPT_TRC.1 | FPT_ITT.1 | The SFR depends on FPT_ITT.1 which requires the protection of the TSF transfer against modification and/or disclosure. This requirement is satisfied by FTP_ITC.1_(IPsec) which provides a secure channel that protects the TSF data in the communication and synchronization between HA peers. |
| FPT_UPD.1EX | FCS_COP.1 | FCS_COP.1_(SWVerify) |
| FPT_UPD.2EX | (no dependencies) | |

Most dependencies for the class FPT have no dependencies so no rationale is needed.
For FPT_RCV.2 the dependency is fulfilled by the single component AGD_OPE.1.
The dependency for FPT_UPD.1EX is fulfilled by FCS_COP.1_(SWVerify) because it addresses software updates.

Table 11: SFR Dependencies for class FTP

| SFR | Dependencies | Satisfied by |
|---|---|---|
| FTP_TRP.1_(TLS) | (no dependencies) | |
| FTP_TRP.1_(SSH) | (no dependencies) | |
| FTP_TRP.1_(Local) | (no dependencies) | |
| FTP_ITC.1_(IPsec) | (no dependencies) | |

All components for the class FTP have no dependencies so no rationale is needed.

### 6.3.2 Security Objectives

In this section a rationale is given regarding the coverage of all Security Objectives for the TOE by the defined Security Functional Requirements.

Table 12 gives an overview of the coverage of all security objectives for the TOE by the SFRs and also show that every SFR contributes to at least one objective.

**O.Rules**

This objective is primarily addressed by FDP_IFC.1 and FDP_IFF.1 which define and enforce the Filter Rules SFP, including the Default-Deny-Policy. This is supported by FMT_SMF.1 which provides the management functions for the detailed configuration of the filter rules, and FMT_MTD.1_(Modify), which restricts these capabilities for configuration to the Administrator.

FPT_FLS.1 requires that a secure state is preserved in case of any self-test failure as required by FPT_SST.1EX, ensuring that the TOE cannot be bypassed in this case. FPT_RCV.2 describes the recovery after failures.

FPT_TRC.1 requires that replicated data is consistent between parts of the TOE and that data flow requests according to the Filter Rules SFP are only accepted if consistency is assured. This is supported by FTP_ITC.1_(IPsec) which provides a trusted channel to protect the TSF data on replication from modification and disclosure.

These SFRs are mutually supportive and together meet the objective.

**O.AuditTrail**

FAU_SAA.1 describes the required security violation analysis, and FAU_ARP.1 ensures configurable actions including the generation of respective audit log entries and issuance of warning messages upon detection of potential security violations.

FAU_GEN.1 describes details for the required audit log generation, supported by FPT_STM.1 that provides the reliable time stamps that are required for the audit log entries.

FAU_SAR.1, FAU_SAR.2 and FAU_SAR.3 ensure that the audit information is readable for Administrators and Auditors, and otherwise access-restricted, and that the audit data is provided suitable for interpretation and appropriate searches.

FAU_STG.1 ensures the storage of the generated audit data on the TOE itself and the ability to transmit the audit data to an external audit server.

In combination these SFRs meet the objective.

**O.AccessControl**

FMT_SMR.1 defines the basis for a user and rights management by introducing the user roles Administrator and Auditor.

The SFRs FDP_ACC.1 and FDP_ACF.1, FMT_MTD.1_(Modify), FMT_MTD.1_(Query), FMT_MOF.1, FMT_MSA.1_(Modify), FMT_MSA.1_(Query) and FMT_MSA.3 allow the execution of the various management functions, but restrict the execution to appropriate roles. In particular FDP_ACF.1.2 a) and FMT_MTD.1_(Modify) restrict the ability for configuration changes to the Administrator, FMT_MOF.1 restricts the ability for software update to the Administrator, and FDP_ACF.1.2 b) and FMT_MTD.1_(Query) restricts the ability to query audit log data to Administrator or Auditor.

Also FAU_SAR.2 requires the restriction of read access to the audit log data to specified users.

This shows that the objective is covered by combination of these SFRs.

**O.Authentication**

FMT_SMR.1 introduces the user roles Administrator and Auditor.

The SFRs FIA_UID.1 and FIA_UAU.1 directly require each user to be successfully authenticated before allowing execution of any management function according to their role. FIA_UAU.5 provides various authentication mechanisms to support this, and FIA_UAU.6 introduces a

Table 12: Coverage of Security Objectives by SFRs

| SFR | O.Rules | O.AuditTrail | O.AccessControl | O.Authentication | O.AuthenticationData | O.Management | O.Logging | O.Update | O.Crypto |
|---|---|---|---|---|---|---|---|---|---|
| FAU_GEN.1 | | X | | | | | X | | |
| FAU_SAA.1 | | X | | | | | | | |
| FAU_ARP.1 | | X | | | | | | | |
| FAU_SAR.1 | | X | | | | | X | | |
| FAU_SAR.2 | | X | X | | | | X | | |
| FAU_SAR.3 | | X | | | | | X | | |
| FAU_STG.1 | | X | | | | | X | | |
| FAU_STG.2 | | | | | | | X | | |
| FAU_STG.4 | | | | | | | X | | |
| FAU_STG.5 | | | | | | | X | | |
| FCS_CKM.1_(RSA) | | | | | | | | | X |
| FCS_CKM.1_(ECDSA) | | | | | | | | | X |
| FCS_CKM.1_(PSK) | | | | | | | | | X |
| FCS_CKM.1_(DH) | | | | | | | | | X |
| FCS_CKM.1_(ECDH) | | | | | | | | | X |
| FCS_CKM.2_(PSK) | | | | | | | | | X |
| FCS_CKM.5_(SSH) | | | | | | | | | X |
| FCS_CKM.5_(TLS) | | | | | | | | | X |
| FCS_CKM.5_(IPsec) | | | | | | | | | X |
| FCS_CKM.5_(IPsecAuth) | | | | | | | | | X |
| FCS_CKM.6 | | | | | | | | | X |
| FCS_COP.1_(SWVerify) | | | | | | | | X | X |
| FCS_COP.1_(Hash) | | | | | | | | | X |
| FCS_COP.1_(HMAC) | | | | | | | | | X |
| FCS_COP.1_(AES) | | | | | | | | | X |
| FCS_COP.1_(RSA) | | | | | | | | | X |
| FCS_COP.1_(ECDSA) | | | | | | | | | X |
| FCS_COP.1_(ECDH) | | | | | | | | | X |
| FCS_COP.1_(DH) | | | | | | | | | X |
| FCS_RNG.1_(DRT.1) | | | | | | | | | X |
| FDP_ACC.1 | | | X | X | | | | | |
| FDP_ACF.1 | | | X | X | | | | | |
| FDP_IFC.1 | X | | | | | | | | |
| FDP_IFF.1 | X | | | | | | | | |

timeout for authentication by requiring re-authentication upon a configurable timeout period of inactivity.

FIA_ATD.1 defines the user security attributes that support the authentication concept. FIA_AFL.1 requires blocking of a user after a configurable number of consecutive failed authentication

Table 12: Coverage of Security Objectives by SFRs (continued)

| SFR | O.Rules | O.AuditTrail | O.AccessControl | O.Authentication | O.AuthenticationData | O.Management | O.Logging | O.Update | O.Crypto |
|---|---|---|---|---|---|---|---|---|---|
| FIA_UID.1 | | | | X | | | | | |
| FIA_UAU.1 | | | | X | | | | | |
| FIA_UAU.5 | | | | X | | | | | |
| FIA_UAU.6 | | | | X | | | | | |
| FIA_UAU.7 | | | | | X | | | | |
| FIA_ATD.1 | | | | X | | | | | |
| FIA_SOS.1 | | | | | X | | | | |
| FIA_AFL.1 | | | | X | X | | | | |
| FMT_SMR.1 | | | X | X | | | | | |
| FMT_SMF.1 | X | | | | | X | | X | |
| FMT_MTD.1_(Modify) | X | | X | | | | | | |
| FMT_MTD.1_(Query) | | | X | | | | | | |
| FMT_MOF.1 | | | X | | | | | X | |
| FMT_MSA.1_(Modify) | | | X | | X | | | | |
| FMT_MSA.1_(Query) | | | X | | | | | | |
| FMT_MSA.1_(NoQuery) | | | | | X | | | | |
| FMT_MSA.3 | | | X | | | | | | |
| FPT_RCV.2 | X | | | | | | | | |
| FPT_SST.1EX | X | | | | | | | X | |
| FPT_FLS.1 | X | | | | | | | X | |
| FPT_STM.1 | | X | | | | | X | | |
| FPT_TRC.1 | X | | | | | | | | |
| FPT_UPD.1EX | | | | | | | | X | |
| FPT_UPD.2EX | | | | | | | | X | |
| FTP_TRP.1_(TLS) | | | | | X | | | | |
| FTP_TRP.1_(SSH) | | | | | X | | | | |
| FTP_TRP.1_(Local) | | | | | X | | | | |
| FTP_ITC.1_(IPsec) | X | | | | | | | | |

attempts with password, hence protecting from brute force attacks on user passwords.

FDP_ACC.1 and FDP_ACF.1 introduce and enforce the Administrative Access Control SFP, where FDP_ACF.1.2 a) and b) require the users to be authenticated for their role before being allowed to perform operations covered by this SFP.

These SFRs are mutually supportive and together meet the objective.

**O.AuthenticationData**

The objective is met by the combination of the following SFRs:

FIA_UAU.7 ensures that there is no feedback or neutral feedback per character while the authentication is in progress, hence protecting the authentication data like password on entry.

FIA_SOS.1 protects the user passwords from being easily guessed by enforcing a specific quality

metric on passwords. Furthermore FIA_AFL.1 protects the user passwords from brute force attacks by requiring a user to be blocked after a specific (configurable) number of consecutive failed authentication attempts.

FMT_MSA.1_(Modify) and FMT_MSA.1_(NoQuery) ensure that stored authentication data can only be modified by an Administrator, and that secret or private authentication data (passwords) cannot be queried by anybody.

The SFRs FTP_TRP.1_(TLS), FTP_TRP.1_(SSH) and FTP_TRP.1_(Local) require trusted channels for initial user authentication and execution of administrative tasks, this protects the authentication data from modification or disclosure when being transmitted to the TOE.

**O.Management**

This objective is directly met by FMT_SMF.1 which requires the capability of the TOE of performing the requested management functions.

**O.Logging**

FAU_GEN.1 enforces the TOE to generate audit log data and describes minimum requirements for the events that must be covered by audit log entries and for the information to be contained. It is supported by FPT_STM.1 that provides the reliable time stamps that are required for the audit log entries.

FAU_SAR.1, FAU_SAR.2 and FAU_SAR.3 ensure that the audit information is readable for Administrators and Auditors, and otherwise access-restricted, and that the audit data is provided suitable for interpretation and appropriate searches.

FAU_STG.1 ensures the storage of the generated audit data on the TOE itself and the ability to transmit the audit data to an external audit server. FAU_STG.2 requires the TOE to protect the audit data from unauthorized deletion and modification. The measures to be taken by the TOE to prevent audit data loss are given by FAU_STG.4 and FAU_STG.5.

These SFRs are mutually supportive and together meet the objective.

**O.Update**

FMT_SMF.1 requires the TOE to provide the functionality to apply software patches. FMT_MOF.1 ensures that software updates require authentication by an Administrator.

FPT_UPD.1EX requires various checks before a software patch is accepted for update, including cryptographic verification of the integrity and authenticity of the software patch before update the checks for authentic patches. The cryptographic operations required for this are provided by FCS_COP.1_(SWVerify).

The SFR FPT_UPD.2EX requires that also the identification data is kept in a consistent state.

Also FPT_SST.1EX requires tests upon software update that ensure that the software patch is intended for the current software version, that the patch level is correct and that the software authenticity and integrity could be successfully verified. If any of these checks fails, FPT_FLS.1 ensures that the TOE remains in a secure state.

In combination these SFRs meet the objective.

**O.Crypto**

This objective is directly met by the use of approved standards in the SFRs FCS_CKM.1_(RSA), FCS_CKM.1_(ECDSA), FCS_CKM.1_(PSK), FCS_CKM.1_(DH) and FCS_CKM.1_(ECDH) (for cryptographic key generation), FCS_CKM.2_(PSK) (for cryptographic key distribution), FCS_CKM.5_(SSH), FCS_CKM.5_(TLS), FCS_CKM.5_(IPsec) and FCS_CKM.5_(IPsecAuth) (for cryptographic key derivation), FCS_CKM.6 (for cryptographic key destruction), and FCS_COP.1_(SWVerify), FCS_COP.1_(Hash), FCS_COP.1_(HMAC), FCS_COP.1_(AES), FCS_COP.1_(RSA), FCS_COP.1_(ECDSA), FCS_COP.1_(ECDH) and FCS_COP.1_(DH) (for cryptographic operation), and by the use of an approved random number generator as required by FCS_RNG.1_(DRT.1). These SFRs describe all cryptographic

operations and key management that are used in the context of secure software update, Administrator or Auditor authentication or to provide a secure channel for remote administration, or for synchronization between HA peers in the HA network.

### 6.3.3 New or tailored SFR

The following rationale justifies the introduction of new SFR components and families.

**FPT_SST.1EX:** The single component of this new family FPT_SST is modelled after component FPT_TST.1. The existing self-tests do not just check the TSFs, but perform tests that can also check any other targets. Therefore, a new family has been introduced.

**FPT_UPD.1EX** This is a new component in the new family FPT_UPD that handles software updates. This functionality is currently not covered in Common Criteria and therefore must be a new family and component.

**FPT_UPD.2EX** This is a new component in the new family FPT_UPD that handles software updates. This functionality is currently not covered in Common Criteria and therefore must be a new family and component.

## 6.4 Security Assurance Requirements Rationale

The overall security claim of this Security Target is aimed at EAL4.

The attack potential of the anonymous users is high. The firewall components are exposed to unrestricted attackers, simply because they are exposed to the Internet. Therefore the vulnerability analysis has been augmented to AVA_VAN.5 in order to match the resistance to attackers with a high attack potential.

For the same reason the TOE summary specification has been augmented to ASE_TSS.2. This augmentation explains the security architecture of the product.

The life cycle support has been augmented by ALC_FLR.2 to demonstrate genua's flaw handling procedures.

The component ALC_PAM.1 has been included in order to have a well defined, secure and correct patch generation process.

The new components are necessary, because application of patches has not been addressed by Common Criteria.

The following table 13 shows that all dependencies are met.

Table 13: SAR Dependencies

| Requirement | Dependency | Solution |
|---|---|---|
| ADV_ARC.1 | ADV_FSP.1 | ADV_FSP.4 |
| | ADV_TDS.1 | ADV_TDS.3 |
| ADV_FSP.4 | ADV_TDS.1 | ADV_TDS.3 |
| ADV_IMP.1 | ADV_TDS.3 | ADV_TDS.3 |
| | ADV_TAT.1 | ALC_TAT.1 |
| ADV_TDS.3 | ADV_FSP.4 | ADV_FSP.4 |
| AGD_OPE.1 | ADV_FSP.1 | ADV_FSP.4 |
| AGD_PRE.1 | - | - |
| ALC_CMC.4 | ALC_CMS.1 | ALC_CMS.4 |
| | ALC_DVS.1 | ALC_DVS.1 |
| | ALC_LCD.1 | ALC_LCD.1 |
| ALC_CMS.4 | - | - |
| ALC_DEL.1 | - | - |
| ALC_DVS.1 | - | - |
| **ALC_PAM.1** | ALC_FLR.2 | ALC_FLR.2 |
| **ALC_FLR.2** | - | - |
| ALC_LCD.1 | - | - |
| ALC_TAT.1 | ADV_IMP.1 | ADV_IMP.1 |
| ASE_CCL.1 | ASE_INT.1 | ASE_INT.1 |
| | ASE_ECD.1 | ASE_ECD.1 |
| | ASE_REQ.1 | ASE_REQ.2 |
| ASE_ECD.1 | - | - |
| ASE_INT.1 | - | - |
| ASE_OBJ.2 | ASE_SPD.1 | ASE_SPD.1 |
| ASE_REQ.2 | ASE_OBJ.2 | ASE_OBJ.2 |
| | ASE_ECD.1 | ASE_ECD.1 |
| ASE_SPD.1 | - | - |
| **ASE_TSS.2** | ASE_INT.1 | ASE_INT.1 |
| | ASE_REQ.1 | ASE_REQ.2 |
| | ADV_ARC.1 | ADV_ARC.1 |
| ATE_COV.2 | ADV_FSP.2 | ADV_FSP.4 |
| | ATE_FUN.1 | ATE_FUN.1 |
| ATE_DPT.1 | ADV_ARC.1 | ADV_ARC.1 |
| | ADV_TDS.2 | ADV_TDS.3 |
| | ATE_FUN.1 | ATE_FUN.1 |
| ATE_FUN.1 | ATE_COV.1 | ATE_COV.2 |
| ATE_IND.2 | ADV_FSP.2 | ADV_FSP.4 |
| | AGD_OPE.1 | AGD_OPE.1 |
| | AGD_PRE.1 | AGD_PRE.1 |
| | ATE_COV.1 | ATE_COV.2 |
| | ATE_FUN.1 | ATE_FUN.1 |
| **AVA_VAN.5** | ADV_ARC.1 | ADV_ARC.1 |
| | ADV_FSP.2 | ADV_FSP.4 |
| | ADV_TDS.3 | ADV_TDS.3 |
| | ADV_IMP.1 | ADV_IMP.1 |
| | AGD_OPE.1 | AGD_OPE.1 |
| | AGD_PRE.1 | AGD_PRE.1 |

# 7 TOE Summary Specification

## 7.1 Implementation of SFRs by the TOE's Security Functions

### 7.1.1 SF.SecurityAudit

**SF.SA.1:** The TOE generates audit log data whenever important events occur. This includes starting and stopping of the system, and changing from normal to the maintenance mode and back. Starting and stopping or reconfiguration of the relays generate log data. Loading of packet filter rules generate log data. The failure of critical components generates audit log entries (self-test failure, in particular corruption of TOE software or TOE configuration).

**SF.SA.2:** All relays generate audit log data when the connection state changes. Log data includes the IP address of source and destination, ports for TCP- and UDP-based protocols, the time stamps for connection and disconnection and the amount of data transferred in both directions for the source and the destination side. The protocol-specific relays log part of the protocol data (e. g. URLs, SMTP-Envelope-lines, ...). The ftprelay, smtprelay, wwwrelay and sshrelay (if configured) log information about authentication. All unsuccessful connection attempts are logged.

**SF.SA.3:** All administration activity through the Web-GUI, REST-API, via SSH or local console that changes any configuration generates log data. The administrative activity is logged together with the user name of the respective Administrator. All successful and unsuccessful authentication attempts are logged. Every log entry contains a time stamp.

**SF.SA.4:** The TOE implements a real time clock that is synchronized with one or more NTP time servers using the protocol NTP version 4. This time from the real time clock is also used for the time stamps in the audit log.

**SF.SA.5:** The log data is continuously analyzed by the tool logwatch that looks for pattern in the log data. The pattern include packet filter violations, selected daemon messages, selected relay messages, selected kernel messages, ARP spoofing messages, failure of time synchronization, usage of duplicate IP addresses, and messages from other processes, e. g. the processes that implement the self-tests. If a pattern matches, this is seen as potential security violation, and a security event is generated. The actions taken upon a security event are configurable, they include logging of the event (per default), use of 'wall' to show the event on the consoles, mailing the event to the Administrators (per default), shutting down network interfaces, and halting the system. It can furthermore be configured that an Administrator-defined shell script with further activities shall be executed.

The extracted log data is written to specific audit log records (var/log/alert, var/log/critical) that are marked with "append-only" OpenBSD file system flags. This mechanism protects these specific audit log records in normal operation mode from any deletion, rotation or manipulation. The file system flags can only be changed in maintenance mode (single-user mode).

**SF.SA.6:** The log data can be transformed into a human readable form and can be searched by all Administrators and Auditors. Other users are not allowed to read the audit log. The possible search criteria are: time, date, process ID and additional log data. For relays the additional log data contains the relay type, connection state, IP addresses and ports, status of logged event, and bytes transferred.

**SF.SA.7:** The audit log data are stored on the TOE, but can also be transmitted for storage to an external audit server if this is configured by an Administrator.

**SF.SA.8:** The system monitors the audit log. If it fills beyond a threshold (default: 80 % of the storage capacity), a configurable action is executed (per default the Administrators are informed).

**SF.SA.9:** When the audit log is full, most parts of the audit log is automatically rotated, i. e. the log data will be deleted after multiple rounds of rotation. In particular the oldest stored audit log records are deleted first. Excepted from this are the flagged audit logs, see SF.SA.5.

**SF.SA.10:** The flagged audit logs (see SF.SA.5) can only be rotated in maintenance mode, and only by a successfully authenticated Administrator. The time span between the rotation passes is large enough so that the security audit can extract relevant log entries and write them to the flagged audit log.

This Security Function addresses the following SFRs: FAU_GEN.1 (Audit data generation); FAU_SAA.1 (Potential violation analysis); FAU_ARP.1 (Security alarms); FAU_SAR.1 (Audit review), FAU_SAR.2 (Restricted audit review) and FAU_SAR.3 (Selectable audit review); FAU_STG.1 (Audit data storage location), FAU_STG.2 (Protected audit data storage), FAU_STG.4 (Action in case of possible audit data loss), and FAU_STG.5 (Prevention of audit data loss); and FPT_STM.1 (Reliable time stamps).

### 7.1.2 SF.DataFlowControl

**SF.DF.1:** The packet filter at the ALG implement the flow control at the network layer (IP) and transport layer (TCP/UDP). The filter rules take the information from the IP and TCP/UDP header (where applicable) in order to apply the filter rules.

All IP packets are identified at the network layer by their source and destination IP addresses (and ports if applicable). The source and destination address of the IP packet are checked against the IP address (and netmask) of the receiving interface. Packets with spoofed source or destination IP addresses are dropped. Packets with source routing are dropped. Packets are not forwarded at the ALG, so that packets that cannot be transmitted to the socket layer are dropped.

**SF.DF.2:** In general, SF.DataFlowControl enforces a Default-deny policy, i. e. every packet that cannot be explicitly assigned to a permitting rule is dropped. In particular any packet is dropped for which there is no appropriate protocol relay available that can control the information flow.

**SF.DF.3:** The relays check and apply filter rules based on the following attributes:

- The header information of network packets, depending on their type:

  **TCP:** IP and TCP header;
  **UDP:** IP and UDP header;
  **ICMP:** IP header and ICMP message;
  **IGMP:** IP header and IGMP message;
  **IP:** IP header;
- The incoming and outgoing interfaces.
- The actual date and time.
- Additional information depending on the handling relay:
  - iprelay: none;
  - pingrelay: none;
  - udprelay: protocol conformance by applying regular expressions at the start of the communication if the filter is activated.
  - tcprelay: protocol conformance by applying regular expressions at the start of the communication if the filter is activated.
  - smtprelay: protocol and application data;
  - smtp2smtprelay: protocol and application data;
  - imaprelay: protocol and application data;
  - poprelay: protocol and application data;

- siprelay: protocol and application data;
- sshrelay: protocol data;
- wwwrelay: protocol and application data;
- httprelay: protocol and application data.
- ftprelay: protocol data;
- mcastudprelay: IGMP and multicast UDP packets;

A virus scanner can be used to scan the application data of the relays smtprelay, poprelay, ftprelay, wwwrelay, smtp2smtprelay, and imaprelay. (The virus scanner itself is not part of the TOE.)

**SF.DF.4:** The TCP-based relays are already connection oriented. The UDP- and IP-related relays introduce a UDP association or IP association respectively. Packages with the same destination IP, (destination port,) source IP, (source port,) and packets where source and destination are reversed are treated as belonging to one connection if they appear within a short timespan one after the other. The connections time out after an idle time with no traffic. As with TCP connections, the connection establishment can be configured to be initiated only by one side. For the iprelay, the IP protocol takes the role of the port.

**SF.DF.5:** The smtprelay can block mails depending on the mail data (virus, blocked extension type of a MIME part). The mail stays on the TOE and must be handled by an Administrator.

**SF.DF.6:** wwwrelay: For data of the content type text/html, a filter can remove the following tags that imply active content: `<applet>`, `<embed>`, `<object>`, `<script>`, and comments. Typical JavaScript fragments, like event handler (on-tags) can also be removed.

**SF.DF.7:** MIME-encoded messages are (recursively) parsed. Their parts are checked like non-encoded messages.

**SF.DF.8:** The sshrelay can intercept SSH connections and block the following SSH protocol messages: shell spawning, command execution and file transfer with scp, local port forwarding, remote port forwarding, X11 forwarding, authentication agent forwarding, and subsystem execution.

**SF.DF.9:** The siprelay can block connections that do not use the configured internal and external domains or use RTP ports outside of the configured port range. The protocol methods can be filtered.

**SF.DF.10:** For the protocols SMTP, SSH, HTTPS and FTP the authentication check can be activated. If this is done, the respective relays smtprelay, sshrelay, wwwrelay and ftprelay will discard all messages for which the user authentication check failed.

**SF.DF.11:** The webservice policy of the httprelay can validate the application data against configurable XML schemas and use only configurable transport protocols.

**SF.DF.12:** An authenticated Administrator can actively terminate connections in the traffic monitor section at the genugate administration interface.

**SF.DF.13:** The filter rules are also applied to any data packets that are directed to the TOE itself and that belong e. g. to administrative commands.

This Security Function addresses the SFRs FDP_IFC.1 (Subset information flow control) and FDP_IFF.1 (Simple security attributes) which cover the security policy ***Filter Rules SFP***, and the part FDP_ACF.1.2 c) (first bullet point) of FDP_ACF.1 (Security attribute-based access control).

### 7.1.3 SF.IdentificationAuthentication

**SF.IA.1:** Supported by SF.SecurityManagement, the TOE implements two different roles for administrative activity: Administrator and Auditor. Administrators can view and also change the configuration (including user and audit log configuration), create, change and delete users, and read and delete audit log records. Auditors (Administrators with read-only rights) can only view the configuration, the users and their attributes and the audit log records. Any other users cannot execute administrative tasks.

**SF.IA.2:** The TOE requires successful identification and authentication of a user before allowing any administrative activity. Only the TOE self-test during initial start-up are done without authentication, and also the usage of the TOE for its original purpose, i. e. sending data traffic through the TOE and apply the Filter Rules SFP, is possible without Administrator or Auditor authentication. Also status information like the software version and patch level can be seen without authentication

**SF.IA.3:** The TOE implements several authentication mechanisms:

1. Password-based authentication:
   For this mechanism the user's password must be configured. The password can be stored on the TOE (only a check value is stored), or on an external authentication server (LDAP or RADIUS).
   The password can be used from remote (via Web-GUI, REST-API or SSH, if configured) or for local administrative access, i. e. via a directly connected serial console or keyboard/monitor.
   If password authentication at the Web-GUI or REST-API is used, the HTTPS protocol and a TLS trusted channel must be used.
   If password authentication via SSH is used, the Secure Shell protocol (SSH version 2) and an SSH trusted channel must be used. SSH is only available if the permission flag for access via SSH is set for the respective user.
   The password is transmitted in the trusted channel and the authentication attempt is accepted if the hash of the password equals the stored hashed password for the respective user.
2. Authentication with a private/public key pair:
   This mechanism can only be used with SSH. Also here, this authentication mechanism is only available if the permission flag for access via SSH is set for the respective user, and if the user's SSH public key is configured.
   The authentication attempt is accepted if the transmitted signature, calculated with the user's SSH private key over some protocol data as specified by the SSH protocol, can be successfully verified with the user's public key as stored on the TOE. As signature algorithms either ECDSA with a key based on one of the NIST curves P-256, P-384 or P-521 and hash algorithm SHA-256 in accordance with FCS_COP.1_(ECDSA) (Cryptographic operation) can be used, or the RSA signature scheme RSASSA-PKCS-v1_5 with hash algorithm SHA-256 or SHA-512, in accordance with FCS_COP.1_(RSA) (Cryptographic operation).

Other interfaces for initial user authentication and administrative access are not implemented. Connections for authentication from remote (i. e. to the Web-GUI, REST-API or via SSH) are only accepted if the request originates from the administration network. The authentication mechanisms are supported by SF.TrustedCommunication which provides the respective trusted channels (TLS, SSH) and the local trusted path.

**SF.IA.4:** Moreover, the security function SF.IdentificationAuthentication implements the option for an additional authentication with a TLS client certificate. This authentication mechanism is not sufficient as an authentication mechanisms in its own but is exclusively used as second factor for Two-Factor-Authentication (2FA): 2FA with a TLS client certificate is possible if used in addition to one of the password-based authentication mechanisms password via TLS. For using 2FA, at least one CA certificate must be configured for the TOE. 2FA is mandatory for a

user if for the respective user a TLS client certificate is configured, in this case a check value for this certificate is stored as security attribute for the user.

An authentication attempt is accepted if the transmitted signature, calculated with the user's private key belonging to the TLS client certificate, can be successfully verified with the public key of the given TLS user's client certificate, if furthermore the given client certificate matches with the stored check value, and if the TLS client certificate can be successfully verified with the public key from one of the CA certificates that are stored on the TOE.

As signature algorithms either ECDSA with a key based on one of the NIST curves P-256 (with hash algorithm SHA-256), P-384 (with SHA-384) or P-521 (with SHA-512) in accordance with FCS_COP.1_(ECDSA) (Cryptographic operation) can be used, or one the RSA signature schemes RSASSA-PKCS-v1_5 or RSASSA-PSS with hash algorithm SHA-256, SHA-384 or SHA-512, in accordance with FCS_COP.1_(RSA) (Cryptographic operation).

**SF.IA.5:** For the realization of these authentication mechanisms, the TOE maintains for each registered user the following security attributes:

- administrative role (Administrator or Auditor or none);
- permission flag for access via SSH;
- user password (if configured);
- user SSH public key (if configured; one of either password or SSH public key is mandatory);
- TLS client certificate (if configured).

The user password can be maintained by the TOE or by an external authentication server (LDAP or RADIUS). If maintained by the TOE, it is not stored in clear but only a check value of the password is stored. Also for a user's TLS client certificate only a check value will be stored.

Furthermore the TOE maintains CA certificates for the TLS client certificate verification (if configured).

**SF.IA.6:** While any authentication attempt is in progress, the TOE does not provide any usable feedback, but only neutral feedback per character (e. g. during password entry at the Web-GUI) or no feedback (e. g. during SSH authentication or at local console).

**SF.IA.7:** When a user password for authentication is set or changed, and the password is not managed on an LDAP or RADIUS server but stored on the TOE, a minimum quality check is performed by the TOE:

- the password length is between 10 and 128 characters
- the password contains at least one number
- the password contains at least one lower case letter
- the password contains at least one upper case letter
- the password contains at least one symbol
- the password does not contain any of the words "root", "admin", "genugate" or "superuser".

If an external authentication server is used for the password management (LDAP, RADIUS), then the Administrator managing the passwords at the authentication server is responsible for the enforcement of an appropriate minimum password quality.

**SF.IA.8:** For all password-based authentication mechanisms, a user account is disabled after a configurable number of consecutive failed authentication attempts from remote. The default value is 5. An Administrator can change this value, and can also reactivate the user account.

**SF.IA.9:** For all administrative activities requiring authentication, the security function implements a timeout for inactivity, after which the Administrator/Auditor has to re-authenticate. The timespan can be configured by an Administrator: the default timeout is 10 minutes for authentication with password via TLS and Web-GUI, 15 minutes for authentication with password via TLS and REST-API, and 60 minutes for authentication via SSH (password or public key) or local password (i. e. password entered via a directly connected serial console or keyboard/monitor).

This Security Function addresses the SFRs FMT_SMR.1 (Security roles); FIA_UID.1 (Timing of identification) and FIA_UAU.1 (Timing of authentication); FIA_UAU.5 (Multiple authentication mechanisms); parts of FDP_ACC.1 (Subset access control) and FDP_ACF.1 (Security attribute-based access control) (SF.IA.1 and SF.IA.3 - with support of SF.SecurityManagement, see in particular SF.SM.3, SF.SM.4 and SF.SM.5 - cover all of FDP_ACC.1 and FDP_ACF.1, except for the first bullet point of FDP_ACF.1.2 c), which is covered by SF.DF.13 of SF.DataFlowControl); FIA_UAU.6 (Re-authenticating) and FIA_UAU.7 (Protected authentication feedback); FIA_ATD.1 (User attribute definition); FIA_AFL.1 (Authentication failure handling); and FIA_SOS.1 (Verification of secrets). Together with SF.SecurityManagement and SF.DF.13 of SF.DataFlowControl, this security function covers the security policy ***Administrative Access Control SFP***.

### 7.1.4 SF.SecurityManagement

**SF.SM.1:** The security function SF.SecurityManagement is based on the two roles Administrator and Auditor, as provided by SF.IdentificationAuthentication.

**SF.SM.2:** Generally, the TOE provides interfaces for various management functions including

1. user configuration;
2. configuration of timeouts for re-authenticating after inactivity;
3. network configuration;
4. relay configuration;
5. DNS server configuration;
6. mail server configuration;
7. packet filter rule configuration;
8. Squid web proxy configuration;
9. virus scanner configuration;
10. audit log configuration;
11. SNMP server configuration;
12. verification of the current patch level;
13. audit log management (like export of audit data to an external audit server, or deletion of audit records);
14. query the audit log

which directly implements FMT_SMF.1 (Specification of Management Functions).

The configuration is divided into the following fields:

- System,
- Connections,
- Users,
- Packet Filter,
- HA,
- Statistics,
- Logging.

The TOE in its non-virtualized form provides moreover interfaces for the configuration of the PFL.

**SF.SM.3:** All configurations can only be created, modified or deleted by an Administrator. They can be queried only by Administrators and Auditors. This addresses most parts of FMT_MTD.1_(Modify) and FMT_MTD.1_(Query) (Management of TSF data).

**SF.SM.4:** Only Administrators can change the audit configuration details, can export audit data to an external audit server and/or configure the automated transmission of audit data, and can manually rotate the audit log (for flagged audit data this is only possible in maintenance mode). Only Administrators and Auditors can view the settings, and can query the audit log. Hence the

remaining parts of FMT_MTD.1_(Modify) and FMT_MTD.1_(Query) (Management of TSF data) are covered.

**SF.SM.5:** Only Administrators can create, change and delete users. Thanks to the security function SF.IdentificationAuthentication, the TOE implements the user attributes password, SSH public key, permission flag for access via SSH, TLS client certificate, and the administrative role of the user, and, as further non-user-specific security attributes, it maintains CA certificates for the TLS client certificate verification. Only Administrators can add, delete or change one of these user attributes, and only Administrators can add, delete or change a CA certificate for TLS client certificate verification. Administrators as well as Auditors can query the settings (however for the certificates only check values can be queried), except for the user password, which can be queried by nobody. (Only a check value of the user password is stored, but also the check value cannot be queried.) This addresses FMT_MSA.1_(Modify), FMT_MSA.1_(Query) and FMT_MSA.1_(NoQuery) (Management of security attributes).

Only Administrators can specify initial values for the user attributes. Per default, upon user creation the user attributes administrative role, user password, SSH public key and the TLS client certificate are not set, and the SSH permission flag is set to "not allowed". Moreover, there is no CA certificate for TLS client certificate verification pre-set per default. This addresses FMT_MSA.3 (Static attribute initialisation).

**SF.SM.6:** Only the Administrator can moreover start-up or shutdown the entire TOE, switch between maintenance and normal operation mode, and apply software patches (that are signed by the developer), hence implementing FMT_MOF.1 (Management of security functions behaviour).

**SF.SM.7:** At its Web-GUI and protected by a TLS session, the TOE provides commands for the import or export of the pre-shared key (PSK) for IPsec for the purpose of secure communication between HA peers in the HA network. These commands are restricted to the Administrator. With the support of SF.TrustedCommunication and SF.CryptographicSupport which provide the TLS channel and the required implementation of the cryptographic algorithms, this implements FCS_CKM.2_(PSK) (Cryptographic key distribution).

The Security Function SF.SecurityManagement therefore addresses the SFRs FMT_SMF.1 (Specification of Management Functions); FMT_MTD.1_(Modify) and FMT_MTD.1_(Query) (Management of TSF data); FMT_MSA.1_(Modify), FMT_MSA.1_(Query) and FMT_MSA.1_(NoQuery) (Management of security attributes); FMT_MSA.3 (Static attribute initialisation); FMT_MOF.1 (Management of security functions behaviour) and FCS_CKM.2_(PSK) (Cryptographic key distribution) (the latter with the support of SF.TrustedCommunication and SF.CryptographicSupport). Together with SF.IdentificationAuthentication, this security function covers the security policy ***Administrative Access Control SFP***.

### 7.1.5 SF.SelfProtection

**SF.SP.1:** The TOE executes self-tests on initial start-up, regularly during normal operation, and on specific conditions:

1. During initial start-up, software authenticity and integrity test are performed. This includes also all configuration files and crypto libraries.
2. The self-tests that are periodically executed during normal operation consist of checking that (a configurable number of) processes are running (default: all relays, daemons, web servers; if one of the expected processes is not running, it is first restarted), that the file system usage is below a configurable threshold, and tests for the file system consistency (file system permissions and flag settings).
3. At the occasion of software update it is verified if the software patch is intended for the current software version, if the patch level is correct, and a software authenticity and integrity test is performed, with support of SF.PatchInstallation.

Administrators and Auditors can view the results of the self tests.

**SF.SP.2:** If any of the self-test fails, the TOE preserves a secure state. Depending on the failure, this can be by disabling or not using the respective process, algorithm, file etc., or even shutdown the entire system. An alert log is written. If the software authenticity and integrity self-test during initial start-up fails, the boot process will not be completed but single-user mode (maintenance mode) will be entered.

**SF.SP.3:** If the HA option is installed and activated, an HA peer will take over the services of the failed system.

The attributes synchronized between HA peers include

- user configuration (but not their blocked status);
- network configuration;
- relay configuration;
- DNS server configuration;
- mail server configuration;
- packet filter rule configuration;
- Squid web proxy configuration;
- virus scanner configuration;
- audit configuration;
- SNMP server configuration.

The TOE in its non-virtualized form synchronizes moreover the data of the PFL configuration between the HA peers.

The systems use an internal revision number to check the configuration. They only reactivate the services when their configuration is up to date, i. e. in particular the TOE will otherwise not be used for its original purpose for filtering data as a firewall but will block in case of inconsistency. The new configuration is used only for new connections, existing connections are not reconfigured.

**SF.SP.4:** In maintenance mode, OpenBSD file system flags can be modified and therefore protected files can be manipulated. To allow an interactive session at the TOE only for the Administrator at the local console, in maintenance mode all network packets (and Ethernet frames) are dropped silently.

This Security Function addresses the SFRs FPT_SST.1EX (TOE testing); FPT_FLS.1 (Failure with preservation of secure state); FPT_RCV.2 (Automated recovery); FPT_TRC.1 (Internal TSF consistency), and parts of FPT_UPD.1EX (Trusted Update) and FPT_UPD.2EX (Update identification data).

### 7.1.6 SF.PatchInstallation

**SF.PI.1:** Whenever a software patch shall be installed, the TOE verifies the integrity and authenticity of the patch using either RSA signatures with a key size of 4096 bit according to PKCS #1, v2.2 using RSASSA-PKCS1-v1_5 and SHA-512 (default), SHA-384 or SHA-256, or ECDSA signatures according to FIPS 186-5 [29] chapter 6, with signature keys based on ECC domain parameters NIST curve P-256 and with hash algorithm SHA-256 in accordance with FCS_COP.1_(SWVerify) (Cryptographic operation).

The patches are signed at the developer's site during the patch generation process and the signature is checked during patch installation.

**SF.PI.2:** During installation of the patch the current patch level is stored on the system in a defined way, and the active identification data is updated. It is verified if the activation of the patch and the update of the identification data have been both completed, in order to keep the system in a defined state.

Activation of the final (updated) TOE will only happen in maintenance mode.

This Security Function addresses the SFR FCS_COP.1.1_(SWVerify) (Cryptographic operation). Together with SF.SelfProtection it also adresses the SFRs FPT_UPD.1EX (Trusted Update) and FPT_UPD.2EX (Update identification data).

### 7.1.7 SF.TrustedCommunication

**SF.TC.1:** The TOE provides four interfaces for initial user authentication (of Administrator or Auditor) and for the execution of administrative commands:

- the Web-GUI,
- the REST-API,
- the SSH console,
- the local console.

Web-GUI, REST-API and SSH are aimed for remote communication (from the administration network). The local console can be used for local administrative access, thus implementing the SFR FTP_TRP.1_(Local) (Trusted path).

**SF.TC.2:** For all communication of a user to the TOE in order to execute administrative tasks from remote via the Web-GUI or the REST-API, the TOE implements an HTTPS server and the TLS protocol (version TLS 1.3) and enforces the usage of a TLS trusted channel. This implements the SFR FTP_TRP.1_(TLS) (Trusted path), and supports SF.SecurityManagement in the implementation of FCS_CKM.2_(PSK).

**SF.TC.3:** For all communication of a user to the TOE in order to execute administrative tasks from remote via SSH, the TOE implements the Secure Shell protocol (SSH version 2, i. e. SSH-2) and enforces the usage of an SSH trusted channel. This implements the SFR FTP_TRP.1_(SSH) (Trusted path).

**SF.TC.4:** For the communication between HA peers, i. e. communication and synchronization between two instances of the TOE for the purpose of high availability, the TOE implements the IPsec protocol (using IKEv2) and provides an IPsec trusted channel and enforces its usage. This implements the SFR FTP_ITC.1_(IPsec) (Inter-TSF trusted channel).

SF.CryptographicSupport supports the realization of the above-mentioned trusted channels TLS, SSH and IPsec with RSA and ECDSA signature generation and verification, hash value calculation, HMAC calculation and verification, Diffie-Hellman and EC Diffie-Hellman shared secret agreement, AES encryption and decryption, cryptographic key generation and cryptographic key derivation, and random number generation by a Deterministic RNG. In particular AES encryption and decryption protects the data transmitted in the respective trusted channel from disclosure, and HMAC or the authentication tag of the AES GCM authenticated encryption protects the data from modification. For details see chapter 8.

### 7.1.8 SF.CryptographicSupport

The security function SF.CryptographicSupport provides various cryptographic algorithms and mechanisms and supports with that other security functions like SF.IdentificationAuthentication and SF.TrustedCommunication.

**SF.CS.1:** The TOE implements the generation of cryptographic keys for various algorithms:

- RSA keys in accordance with FCS_CKM.1_(RSA) (Cryptographic key generation),
- ECDSA keys in accordance with FCS_CKM.1_(ECDSA) (Cryptographic key generation),
- pre-shared keys (PSK) in accordance with FCS_CKM.1_(PSK) (Cryptographic key generation),
- ephemeral Diffie-Hellman (DH) key pairs in accordance with FCS_CKM.1_(DH) (Cryptographic key generation),

- ephemeral EC Diffie-Hellman (ECDH) key pairs in accordance with FCS_CKM.1_(ECDH) (Cryptographic key generation).

**SF.CS.2:** The TOE implements the derivation of cryptographic keys for various algorithms:

- AES and HMAC keys in line with the SSH key derivation algorithm for deriving encryption keys (AES) and integrity protection keys (HMAC), in accordance with FCS_CKM.5_(SSH) (Cryptographic key derivation),
- AES and HMAC keys in line with the TLS key derivation algorithm for deriving AEAD encryption keys (for AES GCM) and MAC keys (HMAC), in accordance with FCS_CKM.5_(TLS) (Cryptographic key derivation),
- AES and HMAC keys in line with the IPsec key derivation algorithm for encryption keys (AES) and integrity protection keys (HMAC), in accordance with FCS_CKM.5_(IPsec) (Cryptographic key derivation),
- HMAC keys in line with the IPsec key derivation algorithm for deriving authentication keys from a pre-shared key (PSK), in accordance with FCS_CKM.5_(IPsecAuth) (Cryptographic key derivation).

**SF.CS.3:** The TOE implements the following cryptographic algorithms:

- RSA or ECDSA signature verification for the integrity and authenticity check of software update packages with an RSA signature key of 4096 bit size or with an ECDSA signature key on NIST curve P-256, in accordance with FCS_COP.1_(SWVerify) (Cryptographic operation),
- hash value calculation with algorithms SHA-256, SHA-384, and SHA-512, in accordance with FCS_COP.1_(Hash) (Cryptographic operation),
- HMAC calculation and verification (with algorithms SHA-256, SHA-384, and SHA-512), in accordance with FCS_COP.1_(HMAC) (Cryptographic operation), and also supporting the security functions SF.TrustedCommunication and SF.SecurityManagement in the implementation of FCS_CKM.2_(PSK),
- AES encryption and decryption in CBC mode, CTR mode or GCM mode, in accordance with FCS_COP.1_(AES) (Cryptographic operation), and also supporting SF.TrustedCommunication and SF.SecurityManagement in the implementation of FCS_CKM.2_(PSK),
- RSA signature generation and verification, in accordance with FCS_COP.1_(RSA) (Cryptographic operation),
- ECDSA signature generation and verification, in accordance with FCS_COP.1_(ECDSA) (Cryptographic operation),
- Elliptic Curve Diffie-Hellman (ECDH) shared secret agreement, in accordance with FCS_COP.1_(ECDH) (Cryptographic operation),
- Diffie-Hellman (DH) shared secret agreement, in accordance with FCS_COP.1_(DH) (Cryptographic operation).

**SF.CS.4:** The TOE provides the following key deletion mechanisms:

- Ephemeral plaintext secret or private keys or intermediate secret key values, including all DRNG entropy input, seed input, and internal state of the DRNG, are zeroized (i. e. overwritten with zeroes) when no longer used.
- Public keys or permanently stored secret or private keys are logically deleted when no longer used.

This implements the SFR FCS_CKM.6 (Timing and event of cryptographic key destruction).

**SF.CS.5:** For the generation of cryptographic keys, nonces, initialization vectors, ECDSA signatures etc. as needed by the other cryptographic functions, the TOE provides random numbers by implementing a deterministic random number generator that is seeded from a non-physical true random number generator, and compliant with class DRT.1 as defined in AIS 20/31 [31], in accordance with the SFR FCS_RNG.1_(DRT.1) (Random number generation).

## 7.2 Self-Protection against Interference and Logical Tampering

The product takes the following additional self-protection measures, supplied by the TOE:

- On the ALG all connections are accepted by relays which are located in a reduced runtime environment (cages). An attacker has only limited capabilities.
- The ALG has a hardened kernel, some system calls are modified and deviate from their POSIX-conformant behaviour. This prevents attackers from escape out of the cages. The system calls are `chroot`, `mknod`, `ktrace`, and `strace`.
- All central processes of the ALG are regularly controlled by the tool sanity-check, which is run as a cron job. In case that a process that is expected to run is not running, sanity-check restarts the process.
- The ALG uses the OpenBSD file system flags and runs at `securelevel=2`. The flags are used to mark most files as read-only and some critical log files as append-only. The securelevel prevents changing the OpenBSD file system flags without going through single user mode.

For the physical genugate, as an additional layer of security hardening, the system is a two-tiered firewall that filters the data traffic on different levels of the network stack, and with both tiers, the ALG and the PFL from the operating environment, running on separated hardware and requiring separated configuration. Both systems have to be overcome to gain unauthorized access from the external network to the internal network. The PFL runs at `securelevel=3`. This means that the packet filter rules as configured on the PFL are immutable.

The following self-protection measures are supplied by the operational environment:

- The OpenBSD kernel uses a randomized stack top, a stack canary to detect stack overflow, and exclusive write or executable memory segments (`W^X`) to mitigate exploits.
- The OpenBSD applications use a randomized stack top, a stack canary to detect stack overflow, and exclusive write or executable memory segments (`W^X`) to mitigate exploits. Further, they use random library memory locations, random `mmap` and `malloc` function results, a read-only data segment `.rodata` for constant data to mitigate exploits.
- The OpenBSD daemons use either privilege revocation or privilege separation if they temporary need enhanced privileges.
- Both the OpenBSD kernel and the core OpenBSD applications use the functions `strlcat` and `strlcpy` to replace `strncat` and `strncpy` that guarantee to null-terminate the result.
- The OpenBSD daemons use `pledge` and `unveil` to prohibit system calls that the daemons do not need to call.
- The OpenBSD applies mitigations for the Spectre/Meltdown class of vulnerabilities. Especially, hyperthreading is disabled.

The measures together build up a multi-layered security barrier that results in a sufficient level of self-protection:

- The low level `strlcat` and `strlcpy` functions prohibit overwriting the allocated memory.
- The stack and memory protection mechanisms make it difficult to insert shell code.
- The privilege reduction functions inhibit a successful attacker to gain further privileges.

The TOE supplies a configuration GUI that check the parameters entered in the HTML forms. This helps to mitigate unintended misconfiguration by Administrators. It also gives a clear user interface for the Administrators and Auditors.

## 7.3 Self-Protection against Bypass

As the TOE is a firewall system, there can be no bypassing if it is installed properly. The security objective for the environment **OE.DataTraffic**, which requires that all traffic to and from the sensitive network passes through the firewall, reflects this.

# 8 Use of Cryptographic Functions

The use of cryptographic functions is summarized in the following tables 14–17, grouped by the respective purposes: software patch verification, SSH, TLS and IPsec.

Table 14: Cryptography used for software patch verification

| No | Purpose | Cryptographic Mechanism | Standard of Implementation | Key Size in Bits | Security Level above 128 Bits |
|---|---|---|---|---|---|
| 1 | Integrity | SHA-512 (default), SHA-384 or SHA-256 | FIPS 180-4 [27] | N/A | yes |
| 2 | Authentication | RSA using RSASSA-PKCS1-v1_5 | PKCS #1, v2.2 [24] | \|k\| = 4096 | yes |
| 3 | Authentication | ECDSA P-256 | FIPS 186-5 [29], NIST SP 800-186 [30] | \|k\| = 256 | yes |

Table 15: Cryptography used for SSH

| No | Purpose | Cryptographic Mechanism | Standard of Implementation | Key Size in Bits | Security Level above 128 Bits |
|---|---|---|---|---|---|
| 1 | Trusted channel | SSHv2 | RFC 4251 [37], RFC 4252 [35], RFC 4253 [38], RFC 4254 [36] | | |
| 2 | SSH server authentication | ECDSA P-256 with SHA-256 | FIPS 186-5 [29], NIST SP 800-186 [30] | \|k\| = 256 | yes |
| 3 | SSH user authentication | ECDSA P-256, P-384 or P-521 with SHA-256 | FIPS 186-5 [29], NIST SP 800-186 [30] | \|k\| = 256, 384 or 521 | yes |
| 4 | Key Agreement | ECDH brainpoolP256r1 with SHA-256, brainpoolP384r1 with SHA-384, or brainpoolP512r1 with SHA-512 | RFC 5656 [34], RFC 5639 [23] | \|k\| = 256, 384 or 512 | yes |
| 5 | Key Agreement | ECDH P-256 with SHA-256, P-384 with SHA-384, or P-521 with SHA-512 | RFC 5656 [34], SEC 1 [33], NIST SP 800-186 [30] | \|k\| = 256, 384 or 521 | yes |
| 6 | Key Agreement | DH MODP group16 or MODP group18 with SHA-512 | RFC 4253 [38], RFC 8268 [1] | \|k\| = 4096 or 8192 | yes |
| 7 | Confidentiality | AES CTR | FIPS 197 [28], NIST-SP800-38A [25] | \|k\| = 128, 192 or 256 | yes |
| 8 | Integrity | HMAC with SHA-256 or SHA-512 | RFC 2104 [21], FIPS 180-4 [27] | \|k\| = 256 or 512 | yes |

Table 16: Cryptography used for TLS

| No | Purpose | Cryptographic Mechanism | Standard of Implementation | Key Size in Bits | Security Level above 128 Bits |
|---|---|---|---|---|---|
| 1 | Trusted channel | TLS 1.3 | [32] | | |
| 2 | TLS server authentication | RSA using RSASSA-PSS or RSASSA-PKCS1-v1_5 with SHA-256, SHA-384 or SHA-512 (with RSASSA-PKCS1-v1_5 only in certificate) | PKCS #1, v2.2 [24] | \|k\| = 3072 (default) or bigger | yes |
| 3 | TLS server authentication | ECDSA P-256 with SHA-256, P-384 with SHA-384, or P-521 with SHA-512 | FIPS 186-5 [29], NIST SP 800-186 [30] | \|k\| = 256, 384 or 512 | yes |
| 4 | TLS client authentication (for 2FA) | RSA using RSASSA-PSS or RSASSA-PKCS1-v1_5 with SHA-256, SHA-384 or SHA-512 (with RSASSA-PKCS1-v1_5 only for certificate verification) | PKCS #1, v2.2 [24] | 3072 <= \|k\| | yes |
| 5 | TLS client authentication (for 2FA) | ECDSA P-256 with SHA-256, P-384 with SHA-384, or P-521 with SHA-512 | FIPS 186-5 [29], NIST SP 800-186 [30] | \|k\| = 256, 384 or 512 | yes |
| 6 | Key Agreement | ECDH P-256 or P-384 | RFC 5656 [34], SEC 1 [33], NIST SP 800-186 [30] | \|k\| = 256 or 384 | yes |
| 7 | Key Agreement | ECDH brainpoolP384r1 or brainpoolP512r1 | RFC 5656 [34], SEC 1 [33], RFC 5639 [23] | \|k\| = 384 or 512 | yes |
| 8 | Confidentiality and Integrity (AEAD) | AES GCM | FIPS 197 [28], NIST SP 800-38D [26] | \|k\| = 128 or 256 | yes |
| 9 | Integrity | HMAC with SHA-256 or SHA-384 | RFC 2104 [21], FIPS 180-4 [27] | \|k\| = 256 or 384 | yes |

Table 17: Cryptography used for IPsec

| No | Purpose | Cryptographic Mechanism | Standard of Implementation | Key Size in Bits | Security Level above 128 Bits |
|---|---|---|---|---|---|
| 1 | Trusted channel | IPsec and IKEv2 | RFC 4301 [19], RFC 7296 [18] | | |
| 2 | Authentication | HMAC with SHA-512, with pre-shared key | RFC 2104 [21], FIPS 180-4 [27] | \|k\| = 512 | yes |
| 3 | Key Agreement | ECDH brainpoolP512r1 | RFC 5656 [34], RFC 5639 [23] | \|k\| = 512 | yes |
| 4 | Pseudo-random function (PRF) | HMAC with SHA-512 | RFC 2104 [21], FIPS 180-4 [27] | N/A | yes |
| 5 | Confidentiality | AES CBC | FIPS 197 [28], NIST SP 800-38A [25] | \|k\| = 256 | yes |
| 6 | Integrity | HMAC with SHA-512 | RFC 2104 [21], FIPS 180-4 [27] | \|k\| = 512 | yes |

# A References


[1] M. Baushke. More Modular Exponentiation (MODP) Diffie-Hellman (DH) Key Exchange (KEX) Groups for Secure Shell (SSH). RFC 8268, Internet Engineering Task Force, December 2017. https://www.rfc-editor.org/info/rfc8268.

[2] Identifikationskennung BSI-VS-AP-0018-2023. VS-Anforderungsprofil Firewall zum Schutz von „VS-NUR FÜR DEN DIENSTGEBRAUCH“ eingestuften Daten, Version 2.0. Technical report, Bundesamt für Sicherheit in der Informationstechnik, 05. Dezember 2023.

[3] Bundesamt für Sicherheit in der Informationstechnik. IT-Grundschutz-Kompendium. https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/IT-Grundschutz/IT-Grundschutz-Kompendium/it-grundschutz-kompendium_node.html.

[4] Bundesamt für Sicherheit in der Informationstechnik. Technische Richtlinie TR-02102-1 – Kryptographische Verfahren: Empfehlungen und Schlüssellängen. Version 2025-01.

[5] Bundesamt für Sicherheit in der Informationstechnik. Technische Richtlinie TR-02102-2 – Kryptographische Verfahren: Verwendung von Transport Layer Security (TLS). Version 2025-01.

[6] Bundesamt für Sicherheit in der Informationstechnik. Technische Richtlinie TR-02102-3 – Kryptographische Verfahren: Verwendung von Internet Protocol Security (IPsec) und Internet Key Exchange (IKEv2). Version 2025-01.

[7] Bundesamt für Sicherheit in der Informationstechnik. Technische Richtlinie TR-02102-4 – Kryptographische Verfahren: Verwendung von Secure Shell (SSH). Version 2025-01.

[8] Bundesamt für Sicherheit in der Informationstechnik. CI Requirements Profile Firewall (BSI-VS-AP-0018_en), Version 2.0. Technical report, Bundesamt für Sicherheit in der Informationstechnik, 05. december 2023.

[9] Bundesamt für Sicherheit in der Informationstechnik. Sicherheitsanforderungen: Nachweise für eine Evaluierung für eine Zulassung bis VS-NfD, Version 1.5, 15. September 2023.

[10] Common Criteria. Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, Version CC:2022 Revision 1, November 2022. CCMB-2022-11-001.

[11] Common Criteria. Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, Version CC:2022 Revision 1, November 2022. CCMB-2022-11-002.

[12] Common Criteria. Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, Version CC:2022 Revision 1, November 2022. CCMB-2022-11-003.

[13] Common Criteria. Common Criteria for Information Technology Security Evaluation, Part 4: Framework for the specification of evaluation methods and activities, Version CC:2022 Revision 1, November 2022. CCMB-2022-11-004.

[14] Common Criteria. Common Criteria for Information Technology Security Evaluation, Part 5: Pre-defined packages of security requirements, CC:2022 Revision 1, November 2022. CCMB-2022-11-005.

[15] Common Criteria. Errata and Interpretation for CC:2022 (Release 1) and CEM:2022 (Release 1), Version 1.1, July 2024. CCMB-2024-07-002.

[16] International Organization for Standardization. ISO/IEC TS 9569:2023(E): Information security, cybersecurity and privacy protection — Evaluation criteria for IT security — Patch Management Extension for the ISO/IEC 15408 series and ISO/IEC 18045, 2023. https://www.iso.org/obp/ui#iso:std:iso-iec:ts:9569:ed-1:v1:en.

[17] C. Kaufman. Internet Key Exchange (IKEv2) Protocol. RFC 4306, Internet Engineering Task Force, December 2005. Obsoleted by RFC 5996, updated by RFC 5282. http://www.ietf.org/rfc/rfc4306.txt.

[18] C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, and T. Kivinen. Internet Key Exchange Protocol Version 2 (IKEv2). RFC 7296, Internet Engineering Task Force, October 2014. Updated by RFC 7427. http://www.ietf.org/rfc/rfc7296.txt.

[19] S. Kent and K. Seo. Security Architecture for the Internet Protocol. RFC 4301, Internet Engineering Task Force, December 2005. Updated by RFC 6040. http://www.ietf.org/rfc/rfc4301.txt.

[20] T. Kivinen and M. Kojo. More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE). RFC 3526, Internet Engineering Task Force, May 2003. http://www.ietf.org/rfc/rfc3526.txt.

[21] H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-Hashing for Message Authentication. RFC 2104, Internet Engineering Task Force, February 1997. Updated by RFC 6151. http://www.ietf.org/rfc/rfc2104.txt.

[22] H. Krawczyk and P. Eronen. HMAC-based Extract-and-Expand Key Derivation Function (HKDF). RFC 5869, Internet Engineering Task Force, May 2010. http://www.ietf.org/rfc/rfc5869.txt.

[23] M. Lochter and J. Merkle. Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation. RFC 5639, Internet Engineering Task Force, March 2010. http://www.ietf.org/rfc/rfc5639.txt.

[24] Kathleen M. Moriarty, Burt Kaliski, Jakob Jonsson, and Andreas Rusch. PKCS #1: RSA Cryptography Specifications Version 2.2. *RFC*, 8017:1–78, 2016. https://doi.org/10.17487/RFC8017, doi:10.17487/RFC8017.

[25] NIST. Recommendation for Block Cipher Modes of Operation – Modes and Techniques. Special Publication 800-38A, U.S. Department of Commerce / National Institute of Standards and Technology, 2001. http://dx.doi.org/10.6028/NIST.SP.800-38A.

[26] NIST. Recommendation for Block Cipher Modes of Operation – Galois/Counter Mode (GCM) and GMAC. Special Publication 800-38D, U.S. Department of Commerce / National Institute of Standards and Technology, November 2007. http://dx.doi.org/10.6028/NIST.SP.800-38D.

[27] NIST. Secure Hash Standard (SHS). Federal Information Processing Standards 180-4, U.S. Department of Commerce / National Institute of Standards and Technology, August 2015. doi:10.6028/NIST.FIPS.180-4.

[28] NIST. Advanced Encryption Standard (AES). Federal Information Processing Standards 197, U.S. Department of Commerce / National Institute of Standards and Technology, 9. May 2023. https://doi.org/10.6028/NIST.FIPS.197-upd1.

[29] NIST. Digital Signature Standard (DSS). Federal Information Processing Standards 186-5, U.S. Department of Commerce / National Institute of Standards and Technology, February 2023. doi:10.6028/NIST.FIPS.186-5.

[30] NIST. Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters. Special Publication 800-186, U.S. Department of Commerce / National Institute of Standards and Technology, February 2023. https://doi.org/10.6028/NIST.SP.800-186.

[31] Matthias Peter and Werner Schindler. A Proposal for Functionality Classes for Random Number Generators. Technical report, Bundesamt für Sicherheit in der Informationstechnik, Bonn, Germany, 10. September 2024. Version 3.0.

[32] E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446, Internet Engineering Task Force, August 2018. https://www.rfc-editor.org/info/rfc8446.

[33] Certicom Research. SEC 1: Elliptic Curve Cryptography. https://www.secg.org/sec1-v2.pdf, May 21 2009. Version 2.0.

[34] D. Stebila and J. Green. Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer. RFC 5656, Internet Engineering Task Force, December 2009. http://www.ietf.org/rfc/rfc5656.txt.

[35] T. Ylonen and C. Lonvick. The Secure Shell (SSH) Authentication Protocol. RFC 4252, Internet Engineering Task Force, January 2006. http://www.ietf.org/rfc/rfc4252.txt.

[36] T. Ylonen and C. Lonvick. The Secure Shell (SSH) Connection Protocol. RFC 4254, Internet Engineering Task Force, January 2006. http://www.ietf.org/rfc/rfc4254.txt.

[37] T. Ylonen and C. Lonvick. The Secure Shell (SSH) Protocol Architecture. RFC 4251, Internet Engineering Task Force, January 2006. http://www.ietf.org/rfc/rfc4251.txt.

[38] T. Ylonen and C. Lonvick. The Secure Shell (SSH) Transport Layer Protocol. RFC 4253, Internet Engineering Task Force, January 2006. Updated by RFC 6668. http://www.ietf.org/rfc/rfc4253.txt.

# B Acronyms

**2FA** Two-Factor-Authentication

**AEAD** Authenticated Encryption with Associated Data

**ALG** Application Level Gateway

**AWC** Advanced Web Categories

**BGP** Border Gateway Protocol

**BSI** Bundesamt für Sicherheit in der Informationstechnik

**CA** Certification Authority

**CARP** Common Address Redundancy Protocol

**CC** Common Criteria for Information Technology Security Evaluation, international standard for IT security evaluation

**CI** Classified Information

**DMZ** Demilitarized Zone

**DNS** Domain Name System

**DNSSEC** Domain Name System Security Extensions

**DRNG** Deterministic RNG

**FTP** File Transfer Protocol

**HA** High Availability

**HTML** Hypertext Markup Language

**HTTPS** Hypertext Transfer Protocol Secure

**ICMP** Internet Control Message Protocol

**IGMP** Internet Group Management Protocol

**IMAP** Internet Message Access Protocol

**IMAPS** IMAP over SSL

**IP** Internet Protocol

**LDAP** Lightweight Directory Access Protocol

**MSSQL** Microsoft SQL Server relational database management system

**MTA** Mail Transfer Agent

**MySQL** A Relational Database Management System

**NTP** Network Time Protocol

**OPC UA** Open Platform Communications Unified Architecture

**OSPF** Open Shortest Path First

**OWASP** Open Web Application Security Project

**PAP** Packet Filter - Application Level Gateway - Packet Filter

**PCF** Protocol Conformance Filter

**PFL** Packet Filter

**POP** Post Office Protocol

**POP3** Post Office Protocol, version 3

**PostgreSQL** PostgreSQL Object-relational Database Management System

**PP** Protection Profile

**PPTP** Point-to-Point Tunneling Protocol

**RADIUS** Remote Authentication Dial-In User Service

**RDP** Remote Desktop Protocol

**REST** Representational State Transfer

**RTP** Real-time Transport Protocol

**SIP** Session Initiation Protocol

**SMB** Server Message Block

**SMTP** Simple Mail Transfer Protocol

**SNMP** Simple Network Management Protocol

**SOAP** Simple Object Access Protocol

**SSH** Secure Shell, see [37], http://www.openssh.org

**SSL** Secure Sockets Layer

**TCP** Transmission Control Protocol

**TLS** Transport Layer Security

**TOE** Target Of Evaluation

**UDP** User Datagram Protocol

**VNC** Virtual Network Computing

**VS** Verschlusssache (Classified Information)

**VSA** VS-Anweisung (Classified Information Instructions)

**VS-AP** VS-Anforderungsprofil (CI Requirements Profile)

**VS-NfD** VS-NUR FÜR DEN DIENSTGEBRAUCH (German CI sensitivity level)

**WAF** Web Application Firewall

**WSDL** Web Service Description Language

# C Glossary

**IPsec** Internet Protocol Security. Protocol suite for encryption and authentication of IP communication

**OpenBSD** Free Unix-like operating system with focus on security (http://www.openbsd.org)