Mauve Corp

Fictional but secure products

Common Criteria Certification
BSI-DSZ-CC-xyz BSI-CC-PP-00zz

Functional Specification

MAUVECORP MAUVEVPN CLIENT
Version 2.11

MauveCorp
Fliederweg 98
D-50020 Köln
certification@mauvecorp.com

Document Version 1.0-SNAPSHOT
2022-09-13
[Commit cc54422 / main]

Contents

List of Tables

List of Figures

1. Introduction

This document contains the necessary information for the evaluation of the security assurance component ADV_FSP.4 for the evaluation of MauveVPN Client. The document starts by describing the physical and logical interfaces that concern the TSF, the security functions of the TOE. After that, the security functions are described in detail. The documentation shows the relationship between interfaces and security functions.

Remarks on Notation

Table 1.1 lists the typographical conventions used in this document and their usage. Often the categories are not as clear-cut as they seem. Occasionally, the same term is used on different levels of abstraction. It is not always possible to determine that level at first glance. For example, a term can be both a keyword taken from the specification and the name of a variable in the code. Typographic consistency supports the reader in determining the level of abstraction. The explanations in Table 1.1 shall motivate the differentiation.

Subsystems and modules are separated by a double colon, as in Subsystem::Module. Interfaces of modules are separated by two forward slashes: Subsystem::Module//Interface.

The quoted names of code elements – especially of the Java-based components – tend to be quite long. In such cases, hyphens are used to split the name across lines, . This avoids extraneous amounts of white space and supports readability.

Typographic Markup	Purpose
keywords	*Keywords* are terms that are directly taken from the specification. This can be the names of configuration parametes and their values. Also other highly specific terms can be typeset as *keywords*.
`code elements`	`code elements` are terms that are taken from the TOE's source code. This can be names of classes, methods and other types, but also their parameters or logical structures of the programming language employed by the TOE.
`file names`	*`file names`* relate to names of file system elements, such as files or directories.
security terms	Terms that are directly related to the Common Criteria framework are typeset as security terms. This can be SFRs or the names of SF and TSFI. Furthermore, the names of subsystems and modules that constitute the TOE are typeset in this way. This holds also for the names of certficiates and other key material.

Table 1.1.: Typographic Conventions

2. Physical Interfaces

2.1. PS.LAN

The TOE uses the interface PS.LAN to communicate with other IT products in the LAN. The interface is an ethernet interface in the form of a RJ45 jack. The logical interface LI.LAN is provided via this interface. Despite the TOE being a software TOE, the hardware plays an important role for the TOE's security. For this reason, we examine the properties of the physical interfaces provided by MauveVPN Client. Sinc ethe physical properties are identical, the description holds for PS.LAN as well as for PS.WAN.

2.2. PS.WAN

This interface is analogous to the interface PS.LAN. It is used to communicate with other IT products in the WAN.

2.3. PS.LED

The TOE's case has LEDs on the front side. The LEDs indicate the TOE's operational status. The logical interface LI.LED is provided via this physical interface.

3. Logical Interfaces

This chapter describes the logical interfaces of the TOE. There is a section for each logical interface. These sections provide a graphical representation of the protocols that are provided by the interface. In these depictions, the protocols that are connected to a TSFI are marked orange. Dotted protocols are interfaces to non-security functions of the TOE, thus non-TSFI. They are depicted nevertheless, because they are part of the external interfaces of the TOE.

3.1. LI.LAN

LI.LAN is a logical interface to the IT products in the LAN. It is accessible via the physical interface PS.LAN. The interface comprises the protocols listed in Table 3.1 together with their port numbers. Figure 3.1 depicts the protocols that contribute to the security aspects of the TSFI (see also the introductory remarks in Chapter 3.

Ethernet for the data link layer,

IP for routing,

TCP und UDP for transport,

DHCP for address assignment in the LAN (TOE as client),

TLS for securing the communications with other IT prodducts in the LAN.

HTTP_Mgmt HTTP for access to the management interface.

Table 3.1 lists protocols and their ports in more detail. Figure 3.1 shows the protocols of interface LI.LAN in relation to each other and to the TCP/IP layer model.

<table>
<tr><td rowspan="2">Application</td><td rowspan="2">—</td><td>HTTP
Mgmt</td></tr>
<tr><td>TLS</td></tr>
<tr><td>Transport</td><td>UDP</td><td>TCP</td></tr>
<tr><td>Internet</td><td colspan="2">IPv4</td></tr>
<tr><td>Link</td><td colspan="2">Ethernet</td></tr>
</table>

Figure 3.1.: Protocols on LS.LAN for the TSFI

Service	In/Out	Protocol	via	Source port	Dest. port	TSFI	Note
Base protocols	–	IEEE802.3	–	–	–	LI.LAN.Ether	
	–	IPv4	IEEE802.3	–	–	LI.LAN.IP	
	–	TCP	IPv4	–	–	LI.LAN.TCP	
	–	UDP	IPv4	–	–	LI.LAN.UDP	
Administration	In	TLS	TCP	any	9443	LI.LAN.TLS	
	In	HTTP	TLS	any	9443	LI.LAN.HTTP_MGMT	

Table 3.1.: Protocols und port numbers for IP/TCP/UDP on LS.LAN

3.1.1. LI.LAN.Ether

3.1.1.1. Purpose and Methods of Use

This interface servers as the *data link layer* to the ethernet network.

Security functions called via this interface
Security functions called via this interface LI.LAN.Ether: (none)

SFR enforced via this interface
SFR enforced via this interface LI.LAN.Ether: (none)

SFR supported via this interface
SFR supported via this interface LI.LAN.Ether: (none)

3.1.1.2. Parameters

This interface implements the ethernet protocol as specified in [IEEE802.3].

3.1.1.3. Actions

3.1.1.4. Error Message Description

3.1.2. LI.LAN.IP

3.1.2.1. Purpose and Methods of Use

This interface implements the *internet layer*. On the internet layer, the TOE supports IPv4. Additionally, the TOE supports ICMP (which is part of IP).

Security functions called via this interface
Security functions called via this interface LI.LAN.IP: (none)

SFR enforced via this interface
SFR enforced via this interface LI.LAN.IP: (none)

SFR supported via this interface
SFR supported via this interface LI.LAN.IP: (none)

3.1.2.2. Parameters

The implementation of IPv4 fulfills the requirements from RFC 791 [RFC 791], RFC 1812 [RFC 1812] and the update in RFC 2644 [RFC 2644]. ICMP is specified in RFC 792 [RFC 792]. The Linux kernel provides the protocol implementation.

3.1.2.3. Actions

3.1.2.4. Error Message Description

3.1.3. LI.LAN.TCP

3.1.3.1. Purpose and Methods of Use

This interface implements the *transport layer*. The TOE implements the Transmission Control Protocol.

SFR enforced via this interface
SFR enforced via this interface LI.LAN.TCP: (none)

SFR supported via this interface
SFR supported via this interface LI.LAN.TCP: (none)

Security functions called via this interface
Security functions called via this interface LI.LAN.TCP: (none)

3.1.3.2. Parameters

The implementation of TCP fulfills the requirements from RFC 793 [RFC 793]. The Linux kernel provides the protocol implementation.

3.1.3.3. Actions

3.1.3.4. Error Message Description

3.1.4. LI.LAN.UDP

3.1.4.1. Purpose and Methods of Use

This interface implements the *transport layer*. The TOE implements the User Datagram Protocol (UDP).

SFR enforced via this interface
SFR enforced via this interface LI.LAN.UDP: (none)

SFR supported via this interface
SFR supported via this interface LI.LAN.UDP: (none)

Security functions called via this interface
Security functions called via this interface LI.LAN.UDP: (none)

3.1.4.2. Parameters

The implementation of UDP fulfills the requirements from RFC 768 [RFC 768]. The Linux kernel provides the protocol implementation.

3.1.4.3. Actions

3.1.4.4. Error Message Description

3.1.5. LI.LAN.TLS

3.1.5.1. Purpose and Methods of Use

LI.LAN.TLS is used to secure the communications to other IP products in the LAN. TLS provides confidentiality and integrity to the connections. An overview of the TLS connections of the TOE in listed in Table B.4.

SFR enforced via this interface
SFR enforced via this interface LI.LAN.TLS: FCS_COP.1/Hash, FCS_COP.1/HMAC, FCS_COP.1/TLS.AES, FCS_COP.1/TLS.Auth, FPT_TDC.1/TLS.Zert, FTP_ITC.1/TLS

SFR supported via this interface
SFR supported via this interface LI.LAN.TLS: FCS_CKM.1, FCS_CKM.2/TLS, FCS_CKM.4, FCS_RNG.1/Hash_DRBG

Security functions called via this interface
Security functions called via this interface LI.LAN.TLS: SF.CryptographicServices, SF.TLS

3.1.5.2. Parameters

The implementation of TLS fulfills the requirements from RFC 5246 [RFC 5246]. The description of the TLS implementation and all TLS connections of the TOE are documented in Section 4.6.

3.1.5.3. Actions

3.1.5.4. Error Message Description

3.1.6. LI.LAN.HTTP_MGMT

3.1.6.1. Purpose and Methods of Use

The TOE implements an HTTP server for communication with the management web application. Confguration data is transported in JSON notation. The protocol is defined as a REST API. Measures are taken to prevent XSS and CSRF attacks. This is described in detail in the Security Architecture [ADV_ARC].

SFR enforced via this interface
SFR enforced via this interface LI.LAN.HTTP_MGMT: FPT_TST.1, FTP_TRP.1/Admin

SFR supported via this interface
SFR supported via this interface LI.LAN.HTTP_MGMT: (none)

Security functions called via this interface
Security functions called via this interface LI.LAN.HTTP_MGMT: SF.SelfProtection, SF.Administration

3.1.6.2. Parameters

The HTTP server implements RFC 2616 [RFC 2616]. The parameters of the REST API are documented in the guidance documentation [AGD_ADM]. JSON is specified in RFC 7159 [RFC 7159].

3.1.6.3. Actions

3.1.6.4. Error Message Description

3.2. LI.WAN

LI.WAN is a logical interface to the IT products in the WAN. It is accessible via the physical interface PS.WAN. The interface comprises the protocols listed in Table **??** together with their port numbers. Figure **??** depicts the protocols that contribute to the security aspects of the TSFI (see also the introductory remarks in Chapter 3.

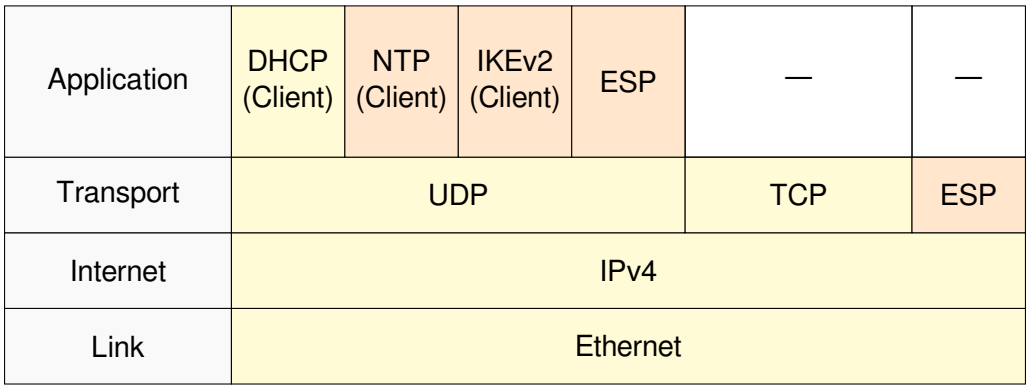

Figure 3.2.: Protocols on LS.WAN for the TSFI

3.2.1. LI.WAN.Ether

The implementation of Ethernet at the WAN interface is identical in all aspects to the implementation at the LAN interface, see section 3.1.1 on page 11.

3.2.2. LI.WAN.IP

The implementation of IPv4 und ICMP at the WAN interface is identical in all aspects to the implementation at the LAN interface, see section 3.1.2 on page 11.

3.2.3. LI.WAN.TCP

The implementation of TCP at the WAN interface is identical in all aspects to the implementation at the LAN interface, see section 3.1.3 on page 12.

3.2.4. LI.WAN.UDP

The implementation of UDP at the WAN interface is identical in all aspects to the implementation at the LAN interface, see section 3.1.4 on page 12.

3.2.5. LI.WAN.DHCP

3.2.5.1. Purpose and Methods of Use

The TOE provides the capability to act as an DHCP client on the WAN interface. The TOE obtain its IP address from a DHCP server in the WAN. This function can be activated/deactivated with the management interface LI.LAN.HTTP_MGMT.

Service	In/Out	Protocol	via	Source port	Dest. port	TSFI	Note
Base protocols	–	IEEE802.3	–	–	–	LI.WAN.Ether	
	–	IPv4	IEEE802.3	–	–	LI.WAN.IP	
	–	TCP	IPv4	–	–	LI.WAN.TCP	
	–	UDP	IPv4	–	–	LI.WAN.UDP	
IPSec	Out	IKEv2	UDP	dyn.	500	LI.WAN.IPSec	
	Out	IKEv2	UDP	dyn.	4500	LI.WAN.IPSec	for UDP-Encapsulation
	–	ESP	IPv4	–	–	LI.WAN.IPSec	
	–	ESP	UDP	dyn.	4500	LI.WAN.IPSec	for UDP-Encapsulation
Zeitdienst	Out	NTP	UDP	–	123	LI.WAN.NTP	
DHCP-Service	Out	DHCP	UDP	68	67	LI.WAN.DHCP	

Table 3.2.: Protocols und port numbers for IP/TCP/UDP on LS.WAN

SFR enforced via this interface
SFR enforced via this interface LI.WAN.DHCP: (none)

SFR supported via this interface
SFR supported via this interface LI.WAN.DHCP: (none)

Security functions called via this interface
Security functions called via this interface LI.WAN.DHCP: (none)[1]: SF.NetworkServices

3.2.5.2. Parameters

DHCP is implemented according to RFC 2131 [RFC 2131].

3.2.5.3. Actions

3.2.5.4. Error Message Description

3.2.6. LI.WAN.NTP

3.2.6.1. Purpose and Methods of Use

This interface is used to synchronize the system time with time servers in the WAN.

SFR enforced via this interface
SFR enforced via this interface LI.WAN.NTP: FPT_STM.1

SFR supported via this interface
SFR supported via this interface LI.WAN.NTP: (none)

Security functions called via this interface
Security functions called via this interface LI.WAN.NTP: SF.NetworkServices

3.2.6.2. Parameters

The protocol is specified in RFC 5905 [RFC 5905] dokumentiert. The destination port for the client is UDP port 123.

3.2.6.3. Actions

3.2.6.4. Error Message Description

3.2.7. LI.WAN.IPSec

3.2.7.1. Purpose and Methods of Use

The TOE uses the WAN interface to establish the VPN channel with IPsec.

SFR enforced via this interface
SFR enforced via this interface LI.WAN.IPSec: FCS_COP.1/Hash, FCS_COP.1/HMAC, FPT_TDC.1/Zert, FTP_ITC.1/VPN

[1]

SFR supported via this interface

SFR supported via this interface LI.WAN.IPSec: FCS_CKM.1, FCS_CKM.2/IKE, FCS_CKM.4, FCS_RNG.1/Hash_DRBG

Security functions called via this interface

Security functions called via this interface LI.WAN.IPSec: SF.CryptographicServices, SF.VPN

3.2.7.2. Parameters

The TOE fulfills the requirements from RFC 4301 [RFC 4301]. It uses two protocols for IPsec. Key exchange is implemented by IKE version 2 according to RFC 7296 [RFC 7296]. User data is tranmitted via the Encapsulation Security Payload (ESP) according toRFC 4303 [RFC 4303].

3.2.7.3. Actions

3.2.7.4. Error Message Description

3.2.7.5. Internet Key Exchange Version 2 (IKEv2)

IKE is used for the distribution of key material and the authentication of the peers. The TOE is initiator of VPN connections and thus submits the suitable algorithms.

3.2.7.6. Encrypted Security Payload (ESP)

ESP provides authenticity, integrity and confidentiality for the transmitted data. Separate algorithms for encryption and integrity protection are used. The negotiated SA including the required keys and algorithm are used to secure the and IP packets sent over the tunnel. ESP headers have the structure shown in Figure 3.3 (see also [RFC 4303, Figure 2]).

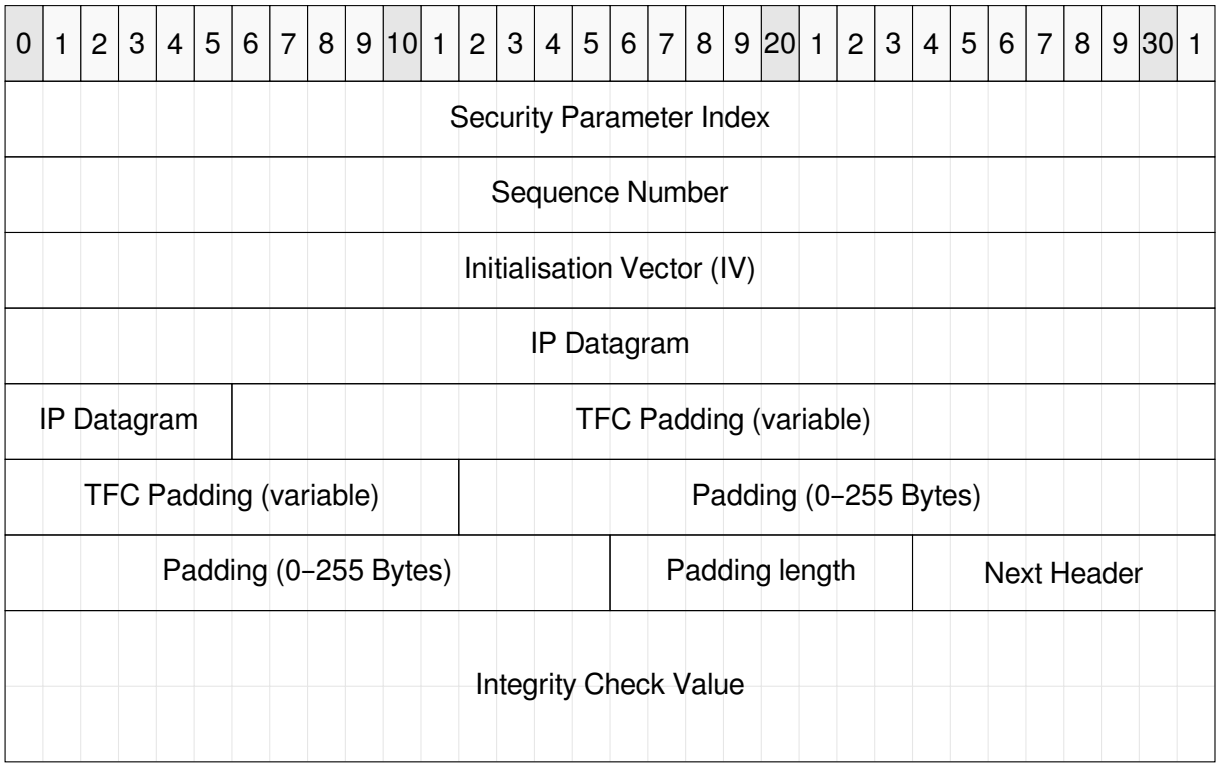

Figure 3.3.: ESP Header

3.3. LI.LED

The TOE can display its operational status via this interface. LI.LED is used by the operating system. The meaning of the different color codes of the LEDs are described in the administrator's guide [AGD_ADM, Section 6].

4. Security Functions of the TOE

4.1. VPN Client (SF.VPN)

Interfaces of the Security Function

The security function SF.VPN is called via LI.WAN.IPSec.

Configuration Parameters of the Security Function

Configuration of the security function is documented in the administrator guide [AGD_ADM, Section 7.4.3.3 VPN (Virtual Private Network)].

Description of the Security Function

The TOE provides a VPN client that enables the TOE to establish secure channels between itself and other IT products of type VPN concentrator. These channels are separated logically from other channels and provide identification and authentication of their end points. Integrity and confidentiality of transmitted data is guaranteed. VPN conections are established on the interface LI.WAN.

4.1.1. Authentication and Key Negotiation with IKE

Key neogtiation is implemented by the IKEv2 protocol according to RFC 4306 [RFC 4306]. The TOE requires the peer to accept Diffie-Hellman-Group 14 according to RFC 3526 [RFC 3526]. The TOE uses a DH exponent of 384 bit. Initiator (TOE) and responder (VPN concentrator) are authenticated mutually with certificate authentication. Signature validation is performed by the security function SF.CryptographicServices (see section 4.5 on page 23).

Implemented SFR	
FTP_ITC.1/VPN	FCS_CKM.2/IKE

4.1.2. Certificate Verification

Certificates of the VPN concentrators are checked for their mathematical correctness, their expiry date and the revocation information.

Implemented SFR
FPT_TDC.1/Zert

.

4.2. Network Services (SF.NetworkServices)

Interfaces of the Security Function

The security function SF.NetworkServices is called via LI.WAN.NTP.

Configuration Parameters of the Security Function

Configuration of the security function is documented in the administrator guide [AGD_ADM, Section 7].

Description of the Security Function

This security function implements several protocols and services that support secure TOE operation by providing access to network services offered by other IT products in the WAN.

4.2.1. NTP Client

The TOE provides a client for the NTP protocol. The TOE uses this service to synchronize the system time with NTP servers in the WAN. This is required by the audit log which requires reliable time stamps.

Implemented SFR
FPT_STM.1

4.2.2. DHCP Client

The TOE provides a DHCP client on its WAN interface LI.WAN.DHCP to obtain IP addresses and routing information according to RFC 2131 [RFC 2131] und RFC 2132 [RFC 2132].

4.3. Self Protection (SF.SelfProtection)

Interfaces of the Security Function

The security function SF.SelfProtection is called via Keine TSFI, LI.LAN.HTTP_MGMT.

Configuration Parameters of the Security Function

This security function has no configuration parameters.

Description of the Security Function

The security function is responsible for the self protection of the TOE and for the protection of data transmitted via the TOE between IT products in the LAN and WAN.

4.3.1. Secure Memory Deallocation

Sensitive data and cryptographic keys are erased securely from the TOE's memory as soon as they are not used anymore. Deletion is implemented by overwriting the memory areas with null bytes. This security function has no TSFI and is only called from within the TOE.

Implemented SFR
FDP_RIP.1

4.3.2. Self-Tests

The TOE implements self-tess that can be used to check integrity and correct functionality of the TSF. The administrator can run the self-tests by calling them via the management interface LI.LAN.HTTP_MGMT.

Implemented SFR
FPT_TST.1

4.4. Administration (SF.Administration)

Interfaces of the Security Function

The security function SF.Administration is called via LI.LAN.HTTP_MGMT.

Configuration Parameters of the Security Function

Configuration of the security function is documented in the administrator guide [AGD_ADM, Section 7].

Description of the Security Function

This security function provides an interface and processes for configuration of the TOE. All operational parameters can be set with this function.

4.5. Cryptographic Services (SF.CryptographicServices)

Interfaces of the Security Function

The security function SF.CryptographicServices is called via LI.LAN.TLS, LI.WAN.IPSec.

Configuration Parameters of the Security Function

This security function has no configuration parameters.

Description of the Security Function

The security function provides cryptographic services that can be used by other TSF.

4.5.1. Random Number Generation

The TOE contains a DRNG to generate random numbers of high quality [NIST SP 800-90A]. This RNG is used to generate the random numbers and nonces used for establishing TLS and IPSec connections.

Implemented SFR
FCS_RNG.1/Hash_DRBG

Supported SFR	
FCS_CKM.1	FCS_COP.1/TLS.AES

4.5.2. HMAC Algorithms

The security function provides implementations of algorithms for HMAC generation. The TOE supports HMAC-SHA-256(-128).

Implemented SFR
FCS_COP.1/HMAC

4.5.3. Signature Verification

The TOE supports signature verification. This is used during certificate validation.

4.5.3.1. Key Exchange for IKE (Diffie-Hellman)

The TOE implements the Diffie-Hellman algorithm for key exchange used by the IKE protocol. The TOE does not reuse DH-keys and thus provides perfect forward secrecy.

Implemented SFR
FCS_CKM.2/IKE

4.5.3.2. Key destruction

The TOE destroys symmetric keys for IKE, ESP and TLS by overwriting the memory areas with null bytes.

Implemented SFR
FCS_CKM.4

4.6. TLS Service (SF.TLS)

The TOE implements a TLS server in protocol version 1.2 according to RFC 5246 [RFC 5246]. All TLS connections and their parameters are listed in Appendix B.

A. Mapping of SFR to TSFI

SFR	TSFI	Relation	Use
FCS_CKM.1	LI.LAN.TLS	Sup.	Key negotiation for TLS
	LI.WAN.IPSec	Sup.	Key negotiation for VPN
FCS_CKM.2/IKE	LI.WAN.IPSec	Sup.	Key distribution for VPN
FCS_CKM.2/TLS	LI.LAN.TLS	Sup.	Key distribution for TLS
FCS_CKM.4	LI.LAN.TLS	Sup.	Terminate TLS connections to LAN
	LI.WAN.IPSec	Sup.	Terminate IPSEC connections to WAN
FCS_COP.1/Hash	LI.WAN.IPSec	Enf.	IPSec hash operations
	LI.LAN.TLS	Enf.	TLS hash operations
FCS_COP.1/HMAC	LI.WAN.IPSec	Enf.	IPSec HMAC operations
	LI.LAN.TLS	Enf.	TLS HMAC operations
FCS_COP.1/TLS.AES	LI.LAN.TLS	Enf.	TLS connections
FCS_COP.1/TLS.Auth	LI.LAN.TLS	Enf.	TLS connections
FCS_RNG.1/Hash_DRBG	LI.LAN.TLS	Sup.	TLS connections
	LI.WAN.IPSec	Sup.	Key negotiation for VPN
FDP_RIP.1	Keine TSFI	Enf.	Not accessible via TSFI
FPT_TDC.1/TLS.Zert	LI.LAN.TLS	Enf.	Validate TLS certificate
FPT_TDC.1/Zert	LI.WAN.IPSec	Enf.	Validate VPN certificate
FPT_STM.1	LI.WAN.NTP	Enf.	Access to time service
FPT_TST.1	LI.LAN.HTTP_MGMT	Enf.	Call self-test
FTP_ITC.1/TLS	LI.LAN.TLS	Enf.	Secure connection to management interface
FTP_ITC.1/VPN	LI.WAN.IPSec	Enf.	Secure IPSec tunnel
FTP_TRP.1/Admin	LI.LAN.HTTP_MGMT	Enf.	Connection to management interface

Table A.1.: Mapping of SFR to TSFI

B. TLS Connections

The Protection Profiles defines the cipher suites required for TLS connections. The TOE uses provides exactly those cipher suites and no other. Table B.1 lists these cipher suites. Table B.2 lists the elliptic curves used for ECDHE.

Algorithmen / Cipher Suite	IANA ID	TLS 1.2 [RFC 5246]
TLS_DHE_RSA_WITH_AES_128_CBC_SHA	0x00,0x33	✓
TLS_DHE_RSA_WITH_AES_256_CBC_SHA	0x00,0x39	✓
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	0xc0,0x13	✓
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	0xc0,0x14	✓
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	0xc0,0x27	✓
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	0xc0,0x28	✓
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	0xc0,0x2f	✓
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	0xc0,0x30	✓

Table B.1.: Cipher suites for TLS connections

Elliptic curve	IANA ID	Standard
secp256r1 (P-256)	23	[RFC 8422; ANSI X9.62]
secp384r1 (P-384)	24	[RFC 8422; ANSI X9.62]
brainpoolP256r1	26	[RFC 7027]
brainpoolP384r1	27	[RFC 7027]

Table B.2.: Elliptic curves for TLS connections

The TOE communicates with other trusted IT products over secure connections. Integrity and confidentiality of these connections is ensured by TLS v1.2. Table B.4 lists all connections the TOE can participate in. Table B.3 describes the columns of this table.

Column	Descriptions
ID	Smybolic name of this connection
Interface	Logical interface whose communication is secured.
Role	Describes client/serer role of the TOE.
Peer	Describes the peer in this TLS connection.
Protocol	Application protocol used in this connection.
Subsystem::Module	Name of the subsystem and the module from which the connection originates or that accepts the connections.
Port	IP-Port that the TOE opens for the connection. If the TOE is client, “dyn.” stands for ephemeral port assignment. “config” stands for a confgurable port number.
Identity of TOE	Certificate that the TOE uses to authenticate itself to the peer.
Identity of Peer	Certificate that the peer uses to authenticate itself to the TOE.
Authentication by	Process, data source or subsystem/module used by the TOE for authentication/verification.

Table B.3.: Legend for TLS connections

ID	Interface (protocol)	Role	Peer	Subsystem::Module	Port	Identity of TOE	Identity of Peer	Authentication by
TLS.1	LI.LAN.HTTP_MGMT	Server	Browser	Administration::HTTP-Server	443	Certificate from Mauve CA	Username/password	User administration in TOE

Table B.4.: TLS connections of MauveVPN Client

Bibliography

Developer Documentation

[ADV_ARC] Mauve Corp. *MauveCorp MauveVPN Client. Security Architecture*. Common Criteria Component ADV_ARC.

[AGD_ADM] Mauve Corp. *Administrator manual MauveCorp MauveVPN Client*. Common Criteria Component AGD_ADM.

Standards

[ANSI X9.62] Accredited Standards Committee X9. *ANSI X9.62, Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*. Standard. ANSI, Nov. 16, 2005.

[IEEE802.3] „IEEE Standard for Ethernet“. In: *IEEE Std 802.3-2015 (Revision of IEEE Std 802.3-2012)* (Mar. 2016), pp. 1–4017. DOI: 10.1109/IEEESTD.2016.7428776.

[NIST SP 800-90A] Elaine Barker and John Kelsey. *Recommendation for Random Number Generation Using Deterministic Random Bit Generators. National Industrial Security Program Operating Manual*. NIST Special Publication. Version Revision 1. National Institute of Standards and Technology, June 2015. URL: http://dx.doi.org/10.6028/NIST.SP.800-90Ar1.

RFC

[RFC 1812] F. Baker. *Requirements for IP Version 4 Routers*. RFC 1812 (Proposed Standard). RFC. Updated by RFCs 2644, 6633. Fremont, CA, USA: RFC Editor, June 1995. DOI: 10.17487/RFC1812. URL: https://www.rfc-editor.org/rfc/rfc1812.txt.

[RFC 2131] R. Droms. *Dynamic Host Configuration Protocol*. RFC 2131 (Draft Standard). RFC. Updated by RFCs 3396, 4361, 5494, 6842. Fremont, CA, USA: RFC Editor, Mar. 1997. DOI: 10.17487/RFC2131. URL: https://www.rfc-editor.org/rfc/rfc2131.txt.

[RFC 2132] S. Alexander and R. Droms. *DHCP Options and BOOTP Vendor Extensions*. RFC 2132 (Draft Standard). RFC. Updated by RFCs 3442, 3942, 4361, 4833, 5494. Fremont, CA, USA: RFC Editor, Mar. 1997. DOI: 10.17487/RFC2132. URL: https://www.rfc-editor.org/rfc/rfc2132.txt.

[RFC 2616] R. Fielding et al. *Hypertext Transfer Protocol – HTTP/1.1*. RFC 2616 (Draft Standard). RFC. Obsoleted by RFCs 7230, 7231, 7232, 7233, 7234, 7235, updated by RFCs 2817, 5785, 6266, 6585. Fremont, CA, USA: RFC Editor, June 1999. DOI: 10.17487/RFC2616. URL: https://www.rfc-editor.org/rfc/rfc2616.txt.

[RFC 2644] D. Senie. *Changing the Default for Directed Broadcasts in Routers*. RFC 2644 (Best Current Practice). RFC. Fremont, CA, USA: RFC Editor, Aug. 1999. DOI: 10.17487/RFC2644. URL: https://www.rfc-editor.org/rfc/rfc2644.txt.

[RFC 3526] T. Kivinen and M. Kojo. *More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)*. RFC 3526 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, May 2003. DOI: 10.17487/RFC3526. URL: https://www.rfc-editor.org/rfc/rfc3526.txt.

[RFC 4301] S. Kent and K. Seo. *Security Architecture for the Internet Protocol*. RFC 4301 (Proposed Standard). RFC. Updated by RFCs 6040, 7619. Fremont, CA, USA: RFC Editor, Dec. 2005. DOI: 10.17487/RFC4301. URL: https://www.rfc-editor.org/rfc/rfc4301.txt.

[RFC 4303] S. Kent. *IP Encapsulating Security Payload (ESP)*. RFC 4303 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, Dec. 2005. DOI: 10.17487/RFC4303. URL: https://www.rfc-editor.org/rfc/rfc4303.txt.

[RFC 4306] C. Kaufman. *Internet Key Exchange (IKEv2) Protocol*. RFC 4306 (Proposed Standard). RFC. Obsoleted by RFC 5996, updated by RFC 5282. Fremont, CA, USA: RFC Editor, Dec. 2005. DOI: 10.17487/RFC4306. URL: https://www.rfc-editor.org/rfc/rfc4306.txt.

[RFC 5246] T. Dierks and E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.2*. RFC 5246 (Proposed Standard). RFC. Updated by RFCs 5746, 5878, 6176, 7465, 7507, 7568, 7627, 7685, 7905, 7919. Fremont, CA, USA: RFC Editor, Aug. 2008. DOI: 10.17487/RFC5246. URL: https://www.rfc-editor.org/rfc/rfc5246.txt.

[RFC 5905] D. Mills et al. *Network Time Protocol Version 4: Protocol and Algorithms Specification*. RFC 5905 (Proposed Standard). RFC. Updated by RFC 7822. Fremont, CA, USA: RFC Editor, June 2010. DOI: 10.17487/RFC5905. URL: https://www.rfc-editor.org/rfc/rfc5905.txt.

[RFC 7027] J. Merkle and M. Lochter. *Elliptic Curve Cryptography (ECC) Brainpool Curves for Transport Layer Security (TLS)*. RFC 7027 (Informational). RFC. Fremont, CA, USA: RFC Editor, Oct. 2013. DOI: 10.17487/RFC7027. URL: https://www.rfc-editor.org/rfc/rfc7027.txt.

[RFC 7159] T. Bray. *The JavaScript Object Notation (JSON) Data Interchange Format*. RFC 7159 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, Mar. 2014. DOI: 10.17487/RFC7159. URL: https://www.rfc-editor.org/rfc/rfc7159.txt.

[RFC 7296] C. Kaufman et al. *Internet Key Exchange Protocol Version 2 (IKEv2)*. RFC 7296 (Internet Standard). RFC. Updated by RFCs 7427, 7670. Fremont, CA, USA: RFC Editor, Oct. 2014. DOI: 10.17487/RFC7296. URL: https://www.rfc-editor.org/rfc/rfc7296.txt.

[RFC 768] J. Postel. *User Datagram Protocol*. RFC 768 (Internet Standard). RFC. Fremont, CA, USA: RFC Editor, Aug. 1980. DOI: 10.17487/RFC0768. URL: https://www.rfc-editor.org/rfc/rfc768.txt.

[RFC 791] J. Postel. *Internet Protocol*. RFC 791 (Internet Standard). RFC. Updated by RFCs 1349, 2474, 6864. Fremont, CA, USA: RFC Editor, Sept. 1981. DOI: 10.17487/RFC0791. URL: https://www.rfc-editor.org/rfc/rfc791.txt.

[RFC 792] J. Postel. *Internet Control Message Protocol*. RFC 792 (Internet Standard). RFC. Updated by RFCs 950, 4884, 6633, 6918. Fremont, CA, USA: RFC Editor, Sept. 1981. DOI: 10.17487/RFC0792. URL: https://www.rfc-editor.org/rfc/rfc792.txt.

[RFC 793] J. Postel. *Transmission Control Protocol*. RFC 793 (Internet Standard). RFC. Updated by RFCs 1122, 3168, 6093, 6528. Fremont, CA, USA: RFC Editor, Sept. 1981. DOI: 10.17487/RFC0793. URL: https://www.rfc-editor.org/rfc/rfc793.txt.

[RFC 8422] Y. Nir, S. Josefsson, and M. Pegourie-Gonnard. *Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS) Versions 1.2 and Earlier*. RFC 8422 (Proposed Standard). RFC. Fremont, CA, USA: RFC Editor, Aug. 2018. DOI: 10.17487/RFC8422. URL: https://www.rfc-editor.org/rfc/rfc8422.txt.