Secusmart

SecuSUITE SIP Server v1.0

Security Target

May 2017

Document prepared by:

Ark Infosec Labs, Inc.
www.arkinfosec.net

Document prepared for:

Electronic Warfare Associates-Canada, Ltd.
https://www.ewa-canada.com/

Document History

Version	Date	Author	Description
1.0	26 June 2016	L Turner	Release for evaluation.
1.1	6 July 2016	L Turner	Update to overwrite details.
1.2	8 July 2016	L Turner	Update physical scope.
1.3	22 Sep 2016	L Turner	Address certification comments. Include TDs 90, 93, 94 & 95.
1.4	6 Feb 2017	L Turner	Final for certification
1.5	7 Apr 2017	L Turner	Address certification comment (ETR Review)
1.6	8 May 2017	L Turner	Address additional TDs.
1.7	9 May 2017	L Turner	TD related updates.

Table of Contents

List of Tables

1 Introduction

1.1 Overview

1 The SecuSUITE Security Solution is Secusmart's solution for customers that want to achieve secure mobile communication across multiple mobile device platforms. It provides end-to-end secure mobile voice communication and instant messaging, using IP-based mobile data connections such as EDGE, UMTS/HSPA, LTE, and Wi-Fi.

2 The SecuSUITE security solution is comprised of a client application (SecuSUITE and Vodafone Secure Call Client) and supporting backend infrastructure, including the SecuSUITE SIP Server v1.0, which enables use of the Session Initiation Protocol (SIP) to establish secure connections between mobile devices.

3 This Security Target (ST) defines the SecuSUITE SIP Server v1.0 Target of Evaluation (TOE) for the purposes of Common Criteria (CC) evaluation.

1.2 Identification

Table 1: Evaluation identifiers

Target of Evaluation	Secusmart SecuSUITE SIP Server v1.0 TOE Build: 1.0.2
Security Target	Secusmart SecuSUITE SIP Server v1.0 Security Target, v1.7, 9 May 2017

1.3 Conformance Claims

4 This ST supports the following conformance claims:

a) CC version 3.1 Revision 4

b) CC Part 2 extended

c) CC Part 3 conformant

d) collaborative Protection Profile for Network Devices, v1.0

e) Network Device collaborative Protection Profile Extended Package SIP Server, v2.0

f) NIAP Technical Decisions:

i) TD0077: Digital Signature Clarification in the SIP EP

ii) TD0090: NIT Technical Decision for FMT_SMF.1.1 Requirement in NDcPP

iii) TD0093: NIT Technical Decision for FIA_X509_EXT.1.1 Requirement in NDcPP

iv) TD0094: NIT Technical Decision for validating a published hash in NDcPP

v) TD0095: NIT Technical Interpretations regarding audit, random bit generation, and entropy in NDcPP

vi) TD0111: NIT Technical Decision for third party libraries and FCS_CKM.1 in NDcPP and FWcPP

vii) TD0112: NIT Technical Decision for TLS testing in the NDcPP v1.0 and FW cPP v1.0

viii) TD0113: NIT Technical Decision for testing and trusted updates in the NDcPP v1.0 and FW cPP v1.0

ix) TD0114: NIT Technical Decision for Re-Use of FIPS test results in NDcPP and FWcPP

x) TD0116: NIT Technical Decision for a Typo in reference to RSASSA-PKCS1v1_5 in NDcPP and FWcPP

xi) TD0117: NIT Technical Decision for FIA_X509_EXT.1.1 Requirement in NDcPP

xii) TD0126: NIT Technical Decision for TLS Mutual Authentication

xiii) TD0130: NIT Technical Decision for Requirements for Destruction of Cryptographic Keys

xiv) TD0143: NIT Technical Decision for Failure testing for TLS session establishment in NDcPP and FWcPP

xv) TD0150: NIT Technical Decision for Removal of SSH re-key audit events in the NDcPP v1.0 and FW cPP v1.0

xvi) TD0151: NIT Technical Decision for FCS_TLSS_EXT Testing - Issue 1 in NDcPP v1.0

xvii) TD0152: NIT Technical Decision for Reference identifiers for TLS in the NDcPP v1.0 and FW cPP v1.0

xviii) TD0153: NIT Technical Decision for Auditing of NTP Time Changes in the NDcPP v1.0 and FW cPP v1.0

xix) TD0154: NIT Technical Decision for Versions of TOE Software in the NDcPP v1.0 and FW cPP v1.0

xx) TD0155: NIT Technical Decision for TLSS tests using ECDHE in the NDcPP v1.0

xxi) TD0156: NIT Technical Decision for SSL/TLS Version Testing in the NDcPP v1.0 and FW cPP v1.0

xxii) TD0164: NIT Technical Decision for Negative testing for additional ciphers for SSH

xxiii) TD0165: NIT Technical Decision for Sending the ServerKeyExchange message when using RSA

xxiv) TD0167: NIT Technical Decision for Testing SSH 2^28 packets

xxv) TD0168: NIT Technical Decision for Mandatory requirement for CSR generation

xxvi) TD0169: NIT Technical Decision for Compliance to RFC5759 and RFC5280 for using CRLs

xxvii) TD0181: NIT Technical Decision for Self-testing of integrity of firmware and software

xxviii) TD0182: NIT Technical Decision for Handling of X.509 certificates related to ssh-rsa and remote comms

xxix) TD0183: NIT Technical Decision for Use of the Supporting Document

xxx) TD0184: NIT Technical Decision for Mandatory use of X.509 certificates

xxxi) TD0185: NIT Technical Decision for Channel for Secure Update

xxxii) TD0187: NIT Technical Decision for Clarifying FIA_X509_EXT.1 test 1

xxxiii) TD0188: NIT Technical Decision for Optional use of X.509 certificates for digital signatures

xxxiv) TD0189: NIT Technical Decision for SSH Server Encryption Algorithms

xxxv) TD0191: NIT Technical Decision for Using secp521r1 for TLS communication

xxxvi) TD0199: NIT Technical Decision for Elliptic Curves for Signatures

xxxvii) TD0200: NIT Technical Decision for Password authentication for SSH clients

xxxviii) TD0201: NIT Technical Decision for Use of intermediate CA certificates and certificate hierarchy depth

1.4 Terminology

Table 2: Terminology

Term	Definition
AIDE	Advanced Intrusion Detection Environment
CC	Common Criteria
EAL	Evaluation Assurance Level
H(A1)	SHA-256 hash of (username-value ":" realm-value ":" SIP password)
OSP	Organizational Security Policy
NDcPP	collaborative Protection Profile for Network Devices, v1.0
PP	Protection Profile
PRF	Pseudorandom Function
RTP	Real-time Transport Protocol – RTP is a network protocol for delivering audio and video over IP networks.
RTCP	Real-time Transport Control Protocol – RTCP provides out-of-band statistics and control information for an RTP session.
SCA	Secure Client Authentication (i.e. SCA server)

Term	Definition
SBC	Session Border Controller – logical component made up of the SIP Server and RTP proxy.
SRTP	Secure Real-time Transport Protocol – SRTP employs AES and HMAC-SHA-1 to provide encryption, message authentication and integrity, and replay protection to RTP data.
SDP	Session Description Protocol – SDP is a format for describing streaming media initialization parameters.
SIP	Session Initiation Protocol - SIP is an application layer communications protocol for signaling and controlling multimedia communication sessions. SIP works in conjunction with several other application layer protocols that identify and carry the session media. Media identification and negotiation is achieved with the Session Description Protocol (SDP). For the transmission of media streams (voice, video) SIP typically employs the Real-time Transport Protocol (RTP) or Secure Real-time Transport Protocol (SRTP). For secure transmissions of SIP messages, the protocol may be encrypted with Transport Layer Security (TLS).
SIP EP	Network Device collaborative Protection Profile Extended Package SIP Server, v2.0
TOE	Target of Evaluation
TSF	TOE Security Functionality
URI	Uniform Resource Identifier

2 TOE Description

2.1 Type

5 The TOE is a SIP server.

2.2 Usage

2.2.1 SecuSUITE Context

6 The SIP Server is an infrastructure component of the SecuSUITE Security Solution shown in Figure 1 below. The SIP Server (TOE) does not work in isolation but relies on other infrastructure components to enable secure VoIP communications.

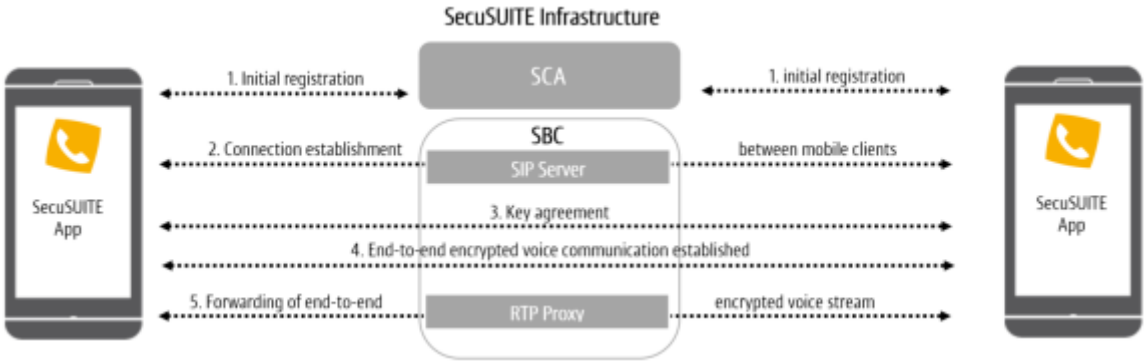


Figure 1: SecuSUITE Security Solution

7 As shown in Figure 1, the SecuSUITE VoIP process flow is as follows:

a) **Step 1 Initial Registration.** Every participating client has to register first to the Secure Client Authentication (SCA) server. The SCA server authenticates users and enrols required client and user certificates as well as client configuration. Only clients that have been enrolled via the SCA service are able to connect to the SIP server and are allowed to establish end-to-end encrypted communication to other SecuSUITE clients. **Note:** Clients must also register to the SIP server using a SIP password. This is in addition to initial client registration with the SCA server.

b) **Step 2 Connection establishment.** The Session Initiation Protocol (SIP) together with TLS is used to establish a secure connection between mobile devices and the SIP server. The use of a TLS connection, providing encryption and mutual authentication, ensures that the devices connect with authorized SIP servers and the dialled call numbers are transmitted encrypted.

c) **Step 3 Key agreement.** When a call is placed and accepted, SecuSUITE clients exchange SIP messages that include digital certificates used to confirm caller identity and perform key agreement for SRTP encryption. The SecuSUITE and Vodafone Secure Call Client Security Target addresses this aspect.

d) **Step 4 End-to-end encrypted voice communication established.** Clients utilize the SRTP protocol to exchange encrypted voice communications. The

voice stream remains encrypted while traversing the SecuSUITE infrastructure and only the clients have access to the session keys. The SecuSUITE and Vodafone Secure Call Client Security Target addresses this aspect.

e) **Step 5 Forwarding of end-to-end encrypted voice stream.** During connection signalling, the SIP server sets up the RTP/RTCP packet bridging in the Real-Time Transport Protocol (RTP) Proxy for this connection. The RTP Proxy relays / bridges the encrypted data stream between clients. The main purpose of the RTP Proxy is to make the communication between SIP user agents behind NAT/NAPT possible.

2.2.2 SIP Server

8 The SIP Server interacts with the SecuSUITE VoIP client and provides registrar and proxy capabilities required for call-session management (e.g. establishing, processing, and terminating VoIP calls). As a SIP registrar, the SIP Server accepts REGISTER requests and places the information received into the location service on the SIP Server. As a SIP proxy server, the SIP Server is a stateful server that manages transactions to route SIP requests and responses. The SIP Server also provides a secure connection between mobile devices running the SecuSUITE app using TLS, providing encryption and mutual authentication.

2.2.3 RTP Proxy

9 The Real-time Transport Protocol (RTP) Proxy bridges media packets sent between clients. The TOE creates and deletes RTP and Real-time Transport Control Protocol (RTCP) bridging sessions in the RTP Proxy.

2.3 Security Functions

10 The TOE provides the following security functions:

a) **Session Initiation Protocol (SIP) Server.** The TOE implements SIP with the Session Description Protocol (SDP) to describe multimedia sessions that are used to carry VoIP traffic. The TOE requires VoIP clients to authenticate with a password (of 8 or more characters) for SIP REGISTER function requests.

b) **Protected Communications.** The TOE protects the integrity and confidentiality of communications with VoIP clients, remote administrators, remote audit servers, update server, database server and peer SIP servers.

c) **Trusted Update.** The TOE ensures the authenticity and integrity of software updates through digital signatures.

d) **System Monitoring.** The TOE keeps local and remote audit records of security relevant events.

e) **Secure Administration.** The TOE enables secure local and remote management of its security functions.

f) **Self Test.** The TOE performs a suite of self-tests to ensure the correct operation and enforcement of its security functions.

g) **Cryptographic Module.** The TOE includes the OpenSSL FIPS Object Module Version 2.0.12.

2.4 Physical Scope

11 The TOE boundary is illustrated in Figure 2.

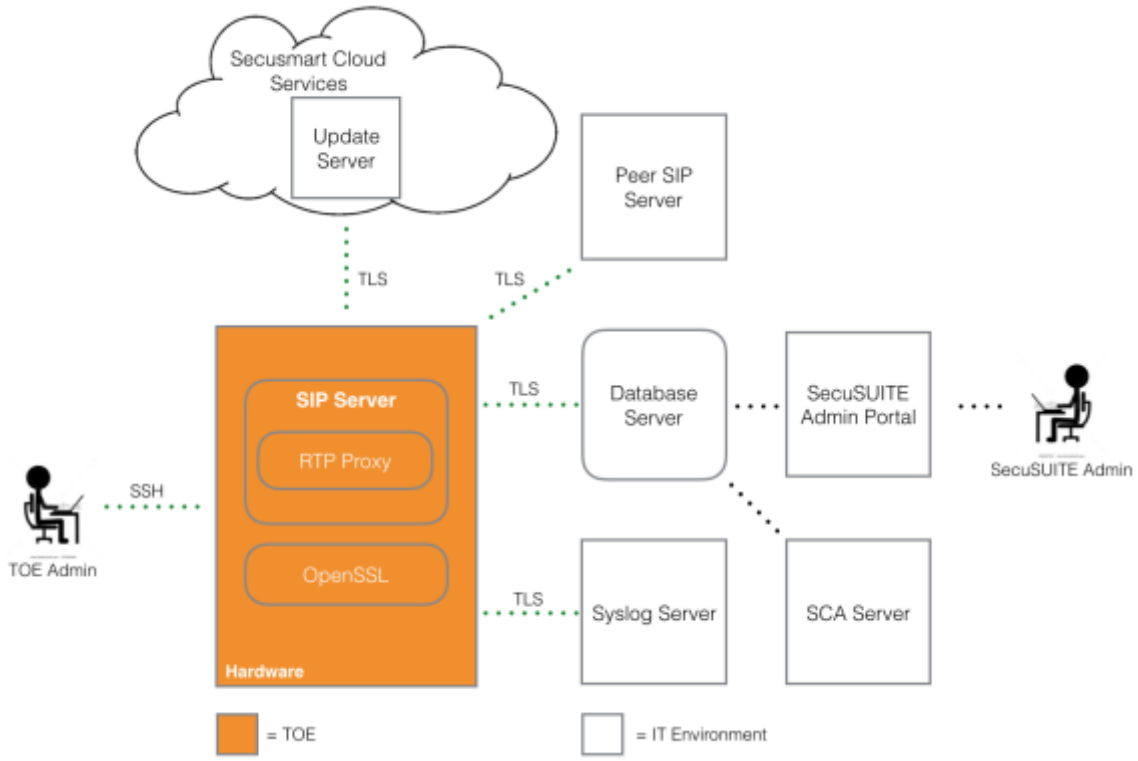


Figure 2: TOE Boundary

12 The TOE is comprised of the following software:

a) SecuSUITE SIP Server including RTP Proxy

b) OpenSSL 1.0.2 with FIPS Object Module v2.0.12

c) Underlying OS - CentOS Linux 7.2.1511 **Note:** SecuSUITE SIP Server also supports Red Hat OS (RHEL 7) however this is not an evaluated configuration.

13 The TOE incorporates the following hardware:

a) Supermicro SYS-1028R-WMR with Intel Xeon E5-2620v3 CPU

2.4.1 Guidance Documents

14 The TOE includes the following guidance documents:

a) SecuSUITE Administration Guide v1.1

b) SecuSUITE SIP Server Administration Guide v1.0

2.4.2 Non-TOE Components

15 The TOE is part of a broader system (SecuSUITE security solution) and requires the following components to be present in the environment:

a) **SecuSUITE Admin Portal v1.0.** Enables VoIP user creation and high-level SecuSUITE administration including statistics and report generation. Resulting settings and configurations are stored in the database server. The TOE does not communicate directly with the Admin Portal.

b) **SecuSUITE Database Server v1.0.** Database that stores configuration data for use by SecuSUITE components including basic settings for the TOE. The TOE communicates with the database server over TLS.

c) **SecuSUITE SCA Server v1.0.** The SCA Server authenticates users and facilitates VoIP client enrollment and pushes client SIP configuration from the database server to the client. Only clients which have been enrolled via the SCA service are able to connect to the SIP server The TOE does not communicate directly with the SCA Server.

d) **Syslog server.** The TOE is able to send audit logs to a remote syslog server.

e) **Peer SIP server.** The TOE can communicate with another SIP server (such as Asterisk SIP or similar) over TLS.

f) **Update Server.** File server that provides updates to the TOE – hosted as a Secusmart cloud service.

2.5 Logical Scope

16 The logical scope of the TOE comprises the security functions defined in section 2.3.

3 Security Problem Definition

17 The security problem definition is reproduced from section 3 of the NDcPP and Annex A.1 and A.2 of the SIP EP. No additional threats, assumptions or OSPs are added in this Security Target.

3.1 Threats

Table 3: SIP EP Threats

Identifier	Description
T.ADMIN_ERROR	An administrator may unintentionally install or configure the TOE incorrectly, resulting in ineffective security mechanisms.
T.TSF_FAILURE	Security mechanisms of the TOE may fail, leading to a compromise of the TSF.
T.UNDETECTED_ACTIONS	Malicious remote users or external IT entities may take actions that adversely affect the security of the TOE. These actions may remain undetected and thus their effects cannot be effectively mitigated.
T.UNAUTHORIZED_ACCESS	A user may gain unauthorized access to the TOE data and TOE executable code. A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources. A malicious user, process, or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data.
T.UNAUTHORIZED_UPDATE	A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE.
T.USER_DATA_REUSE	User data may be inadvertently sent to a destination not intended by the original sender.

Table 4: NDcPP Threats

Identifier	Description
T.UNAUTHORIZED_ADMINISTRATOR_ACCESS	Threat agents may attempt to gain administrator access to the network device by nefarious means such as masquerading as an administrator to the device, masquerading as the device to an administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between network devices. Successfully gaining administrator access allows malicious actions that

Identifier	Description
	compromise the security functionality of the device and the network on which it resides.
T.WEAK_CRYPTOGRAPHY	Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.
T.UNTRUSTED_COMMUNICATION_CHANNELS	Threat agents may attempt to target network devices that do not use standardized secure tunnelling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the network device itself.
T.WEAK_AUTHENTICATION_ENDPOINTS	Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g., shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the network device itself could be compromised.
T.UPDATE_COMPROMISE	Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.
T.UNDETECTED_ACTIVITY	Threat agents may attempt to access, change, and/or modify the security functionality of the network device without administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the administrator would have no knowledge that the device has been compromised.
T.SECURITY_FUNCTIONALITY_COMPROMISE	Threat agents may compromise credentials and device data enabling continued access to the network device and its critical data. The compromise of credentials include replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the administrator or device credentials for use by the attacker.

Identifier	Description
T.PASSWORD_CRACKING	Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic, and may allow them to take advantage of any trust relationships with other network devices.
T.SECURITY_FUNCTIONALITY_FAILURE	A component of the network device may fail during start-up or during operations causing a compromise or failure in the security functionality of the network device, leaving the device susceptible to attackers.

3.2 Organizational Security Policies

Table 5: NDcPP OSPs

Identifier	Description
P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.

3.3 Assumptions

Table 6: SIP EP Assumptions

Identifier	Description
A.NO_GENERAL_PURPOSE	It is assumed that there are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
A.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.
A.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

Table 7: NDcPP Assumptions

Identifier	Description
A.PHYSICAL_PROTECTION	The network device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security and/or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP will not include any requirements on physical tamper protection or other physical attack mitigations. The cPP will not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device.
A.LIMITED_FUNCTIONALITY	The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example the device should not provide computing platform for general purpose applications (unrelated to networking functionality).
A.NO_THRU_TRAFFIC_PROTECTION	A standard/generic network device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the network device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the network device, destined for another network entity, is not covered by the ND cPP. It is assumed that this protection will be covered by cPPs for particular types of network devices (e.g, firewall).
A.TRUSTED_ADMINISTRATOR	The Security Administrator(s) for the network device are assumed to be trusted and to act in the best interest of security for the organization. This includes being appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The network device is not expected to be capable of defending against a malicious administrator that actively works to bypass or compromise the security of the device.
A.REGULAR_UPDATES	The network device firmware and software is assumed to be updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.ADMIN_CREDENTIALS_SECURE	The administrator's credentials (private key) used to access the network device are protected by the platform on which they reside.

4 Security Objectives

18 The security objectives are reproduced from section 4 of the NDcPP and Annex A.3 of the SIP EP. No additional objectives are added in this Security Target.

4.1 Objectives for the Operational Environment

Table 8: SIP EP Operational environment objectives

Identifier	Description
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

Table 9: NDcPP Operational environment objectives

Identifier	Description
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
OE.NO_THRU_TRAFFIC_PROTECTION	The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.
OE.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.
OE.UPDATES	The TOE firmware and software is updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
OE.ADMIN_CREDENTIALS_SECURE	The administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.

4.2 Objectives for the TOE

Table 10: SIP EP Security objectives

Identifier	Description
O.PROTECTED_COMMUNICATIONS	The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities.
O.VERIFIABLE_UPDATES	The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered and (optionally) from a trusted source.
O.SYSTEM_MONITORING	The TOE will provide the capability to generate audit data and send those data to an external IT entity.
O.DISPLAY_BANNER	The TOE will display an advisory warning regarding use of the TOE.
O.TOE_ADMINISTRATION	The TOE will provide mechanisms to ensure that only administrators are able to log in and configure the TOE, and provide protections for logged-in administrators.
O.RESIDUAL_INFORMATION_CLEARING	The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.
O.SESSION_LOCK	The TOE shall provide mechanisms that mitigate the risk of unattended sessions being hijacked.
O.TSF_SELF_TEST	The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly.

Table 11: NDcPP Security objectives

Identifier	Description
None defined	

5 Security Requirements

5.1 Conventions

19 This document uses the following font conventions to identify the operations defined by the CC:

a) **Assignment.** Indicated with italicized text.

b) **Refinement.** Indicated with bold text and strikethroughs.

c) **Selection.** Indicated with underlined text.

d) **Assignment within a Selection:** Indicated with italicized and underlined text.

e) **Iteration.** Indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3).

20 **Note:** This ST reproduces the SFRs, including applied conventions and identified operations, from the NDcPP and SIP EP.

5.2 Extended Components Definition

21 Table 12 identifies the extended components that are incorporated into this ST.

Table 12: Extended Components

Component	Title	Rationale
FAU_STG_EXT.1	Protected Audit Event Storage	Drawn from NDcPP
FCS_SSHS_EXT.1	SSH Server Protocol	
FCS_TLSC_EXT.2	TLS Client Protocol with authentication	
FCS_TLSS_EXT.2	TLS Server Protocol with mutual authentication	
FCS_RBG_EXT.1	Random Bit Generation	
FIA_PMG_EXT.1	Password Management	
FIA_UIA_EXT.1	User Identification and Authentication	
FIA_UAU_EXT.2	Password-based Authentication Mechanism	
FIA_X509_EXT.1	X.509 Certificate Validation	
FIA_X509_EXT.2	X.509 Certificate Authentication	
FIA_X509_EXT.3	X.509 Certificate Requests	
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all symmetric keys)	
FPT_APW_EXT.1	Protection of Administrator Passwords	

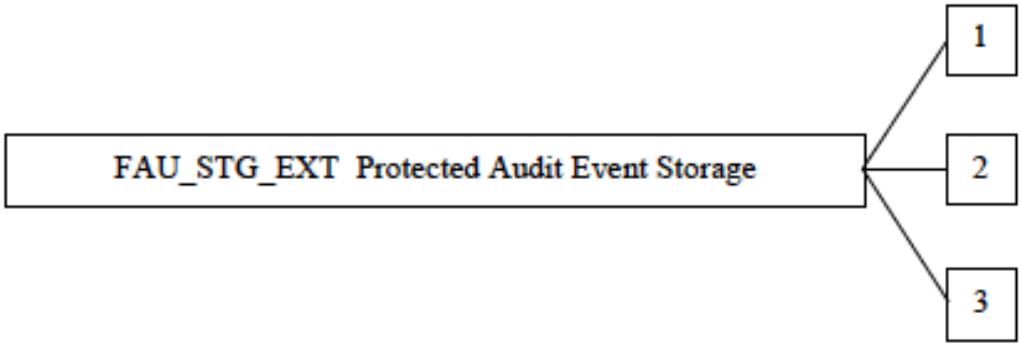
Component	Title	Rationale
FPT_TST_EXT.1	TSF testing	
FPT_TUD_EXT.1	Extended: Trusted update	
FTA_SSL_EXT.1	TSF-initiated Session Locking	
FIA_SIPS_EXT.1	Session Initiation Protocol (SIP) Server	Drawn from SIP EP

5.2.1 Protected audit event storage (FAU_STG_EXT)

5.2.1.1 Family Behavior

22 This component defines the requirements for the TSF to be able to securely transmit audit data between the TOE and an external IT entity.

5.2.1.2 Component Leveling



23 FAU_STG_EXT.1 Protected audit event storage requires the TSF to use a trusted channel implementing a secure protocol.

24 FAU_STG_EXT.2 Counting lost audit data requires the TSF to provide information about audit records affected when the audit log becomes full.

25 FAU_STG_EXT.3 Display warning for local storage space requires the TSF to generate a warning before the audit log becomes full.

5.2.1.3 Management: FAU_STG_EXT.1, FAU_STG_EXT.2, FAU_STG_EXT.3

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

a) The TSF shall have the ability to configure the cryptographic functionality.

5.2.1.4 Audit: FAU_STG_EXT.1, FAU_STG_EXT.2, FAU_STG_EXT.3

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

a) No audit necessary.

FAU_STG_EXT.1 Protected Audit Event Storage

Hierarchical to: No other components.

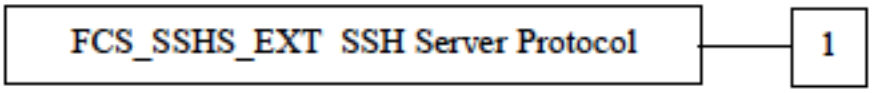
Dependencies:	FAU_GEN.1 Audit data generation FTP_ITC.1 Inter-TSF Trusted Channel
FAU_STG_EXT.1.1	The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC.
FAU_STG_EXT.1.2	The TSF shall be able to store generated audit data on the TOE itself.
FAU_STG_EXT.1.3	The TSF shall [selection: drop new audit data, overwrite previous audit records according to the following rule: [assignment: rule for overwriting previous audit records], [assignment: other action]] when the local storage space for audit data is full.

5.2.2 SSH Server Protocol (FCS_SSHS_EXT.1)

5.2.2.1 Family Behavior

28 The component in this family addresses the ability for a server to offer SSH to protect data between a client and the server using the SSH protocol.

5.2.2.2 Component Leveling



29 FCS_SSHS_EXT.1 SSH Server requires that the server side of SSH be implemented as specified.

5.2.2.3 Management: FCS_CKM_EXT.4

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

a) There are no management activities foreseen.

5.2.2.4 Audit: FCS_SSHS_EXT.1

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

a) Failure of SSH session establishment.

b) SSH session establishment

c) SSH session termination

FCS_SSHS_EXT.1 SSH Server Protocol

Hierarchical to:	No other components.
Dependencies:	FCS_COP.1(1) Cryptographic operation (AES Data encryption/decryption) FCS_COP.1(2) Cryptographic operation (Signature Verification) FCS_COP.1(3) Cryptographic Operation (Hash Algorithm)

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254, and [selection: 5647, 5656, 6187, 6668, no other RFCs].

FCS_SSHS_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, password based.

FCS_SSHS_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [assignment: number of bytes] bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [selection: aes128-cbc, aes256-cbc, AEAD_AES_128_GCM, AEAD_AES_256_GCM].

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH transport implementation uses [assignment: List of public key algorithms] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [assignment: List of MAC algorithms] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7 The TSF shall ensure that [assignment: List of key exchange methods] are the only allowed key exchange methods used for the SSH protocol.

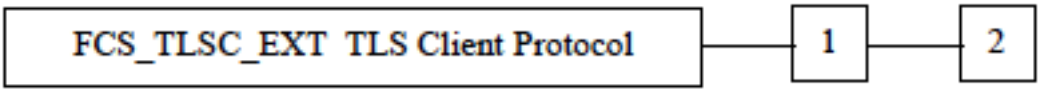
FCS_SSHS_EXT.1.8 The TSF shall ensure that within SSH connections the same session keys are used for a threshold of no longer than one hour, and no more than one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

5.2.3 TLS Client Protocol (FCS_TLSC_EXT)

5.2.3.1 Family Behavior

32 The component in this family addresses the ability for a client to use TLS to protect data between the client and a server using the TLS protocol.

5.2.3.2 Component Leveling



33 FCS_TLSC_EXT.1 TLS Client requires that the client side of TLS be implemented as specified.

34 FCS_TLSC_EXT.2 TLS Client requires that the client side of the TLS implementation include mutual authentication.

5.2.3.3 Management: FCS_TLSC_EXT.1, FCS_TLSC_EXT.2

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

a) There are no management activities foreseen.

5.2.3.4 Audit: FCS_TLSC_EXT.1, FCS_TLSC_EXT.2

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

a) Failure of TLS session establishment.

b) TLS session establishment

c) TLS session termination

FCS_TLSC_EXT.2 TLS Client Protocol with Authentication

Hierarchical to: FCS_TLSC_EXT.1

Dependencies: FCS_COP.1(1) Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1(2) Cryptographic operation (Signature Verification)
FCS_COP.1(3) Cryptographic Operation (Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation

FCS_TLSC_EXT.2.1 The TSF shall implement "[selection: TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)] supporting the following ciphersuites:

Mandatory Ciphersuites:

- [assignment: List of mandatory ciphersuites and reference to RFC in which each is defined]

[selection: Optional Ciphersuites:

- [assignment: List of optional ciphersuites and reference to RFC in which each is defined]

no other ciphersuite].

FCS_TLSC_EXT.2.2 The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.2.3 The TSF shall only establish a trusted channel if the peer certificate is valid.

FCS_TLSC_EXT.2.4 The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [assignment: List of supported curves including an option for ‘none’].

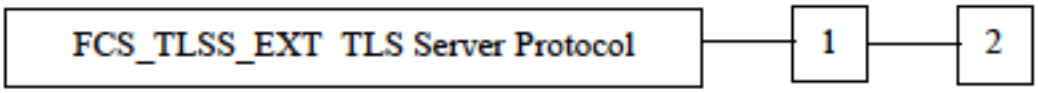
FCS_TLSC_EXT.2.5 The TSF shall support mutual authentication using X.509v3 certificates.

5.2.4 TLS Server Protocol (FCS_TLSS_EXT)

5.2.4.1 Family Behavior

37 The component in this family addresses the ability for a server to use TLS to protect data between a client and the server using the TLS protocol.

5.2.4.2 Component Leveling



38 FCS_TLSS_EXT.1 TLS Server requires that the server side of TLS be implemented as specified.

39 FCS_TLSS_EXT.2: TLS Server requires the mutual authentication be included in the TLS implementation.

5.2.4.3 Management: FCS_TLSS_EXT.1, FCS_TLSS_EXT.2

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

a) There are no management activities foreseen.

5.2.4.4 Audit: FCS_TLSS_EXT.1, FCS_TLSS_EXT.2

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

a) Failure of TLS session establishment.

b) TLS session establishment

c) TLS session termination

FCS_TLSS_EXT.2 TLS Server Protocol with mutual authentication

Hierarchical to: No other components.

Dependencies: FCS_CKM.1 Cryptographic Key Generation
FCS_COP.1(1) Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1(2) Cryptographic operation (Signature Verification)
FCS_COP.1(3) Cryptographic Operation (Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation

FCS_TLSS_EXT.2.1 The TSF shall implement [selection: TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)] supporting the following ciphersuites:

Mandatory Ciphersuites:

- [assignment: List of mandatory ciphersuites and reference to RFC in which each is defined]

[selection: Optional Ciphersuites:

- [assignment: List of optional ciphersuites and reference to RFC in which each is defined]

no other ciphersuite].

FCS_TLSS_EXT.2.2 The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, and [selection: TLS 1.1, TLS 1.2, none].

FCS_TLSS_EXT.2.3 The TSF shall generate key establishment parameters using RSA with key size 2048 bits and [selection: 3072 bits, 4096 bits, no other size] and [selection: over NIST curves [selection: secp256r1, secp384r1, secp521r1] and no other curves; Diffie-Hellman parameters of size 2048 bits and [selection: 3072 bits, no other size]; no other].

FCS_TLSS_EXT.2.4 The TSF shall support mutual authentication of TLS clients using X.509v3 certificates.

FCS_TLSS_EXT.2.5 The TSF shall not establish a trusted channel if the peer certificate is invalid.

FCS_TLSS_EXT.2.6 The TSF shall not establish a trusted channel if the distinguished name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match the expected identifier for the peer.

5.2.5 Random Bit Generation (FCS_RBG_EXT)

5.2.5.1 Family Behavior

42 Components in this family address the requirements for random bit/number generation. This is a new family defined for the FCS class.

5.2.5.2 Component Leveling

43 FCS_RBG_EXT.1 Random Bit Generation requires random bit generation to be performed in accordance with selected standards and seeded by an entropy source.

5.2.5.3 Management: FCS_RBG_EXT.1

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

a) There are no management activities foreseen

5.2.5.4 Audit: FCS_RBG_EXT.1

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

a) Minimal: failure of the randomization process

FCS_RBG_EXT.1 Extended: Cryptographic operation (Random Bit Generation)

Hierarchical to: No other components.

Dependencies: None

FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [selection: Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES)].

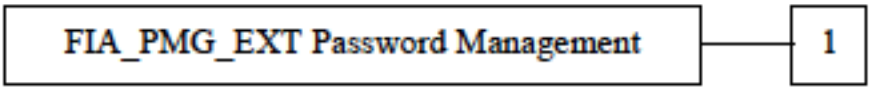
FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [selection: [assignment: number of software-based sources] software-based noise source, [assignment: number of hardware-based sources] hardware based noise source] with minimum of [selection; 128 bits, 192 bits, 256 bits] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 “Security Strength Table for Hash Functions”, of the keys and hashes that it will generate.

5.2.6 Password Management (FIA_PMG_EXT)

5.2.6.1 Family Behavior

46 The TOE defines the attributes of passwords used by administrative users to ensure that strong passwords and passphrases can be chosen and maintained.

5.2.6.2 Component Leveling



47 FIA_PMG_EXT.1 Password management requires the TSF to support passwords with varying composition requirements, minimum lengths, maximum lifetime, and similarity constraints.

5.2.6.3 Management: FIA_PMG_EXT.1

48 No management functions.

5.2.6.4 Audit: FIA_PMG_EXT.1

49 No specific audit requirements.

FIA_PMG_EXT.1 Password Management

Hierarchical to: No other components

Dependencies: No other components

FIA_PMG_EXT.1.1 The TSF shall provide the following password management capabilities for administrative passwords:

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [selection: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, [assignment: other characters]];

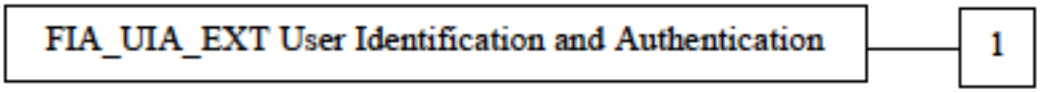
b) Minimum password length shall be settable by the Security Administrator, and support passwords of 15 characters or greater.

5.2.7 User Identification and Authentication (FIA_UIA_EXT)

5.2.7.1 Family Behavior

50 The TSF allows certain specified actions before the non-TOE entity goes through the identification and authentication process.

5.2.7.2 Component Leveling



51 FIA_UIA_EXT.1 User Identification and Authentication requires administrators (including remote administrators) to be identified and authenticated by the TOE, providing assurance for that end of the communication path. It also ensures that every user is identified and authenticated before the TOE performs any mediated functions.

5.2.7.3 Management: FIA_UIA_EXT.1

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

a) Ability to configure the list of TOE services available before an entity is identified and authenticated

5.2.7.4 Audit: FIA_UIA_EXT.1

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

a) All use of the identification and authentication mechanism

b) Provided user identity, origin of the attempt (e.g. IP address)

FIA_UIA_EXT.1 User Identification and Authentication

Hierarchical to: No other components.

Dependencies: FTA_TAB.1 Default TOE Access Banners

FIA_UIA_EXT.1.1 The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- [selection: no other actions, [assignment: list of services, actions performed by the TSF in response to non-TOE requests.]]

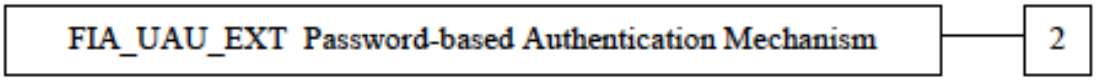
FIA_UIA_EXT.1.2 The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

5.2.8 User authentication (FIA_UAU_EXT)

5.2.8.1 Family Behavior

54 Provides for a locally based administrative user authentication mechanism.

5.2.8.2 Component Leveling



55 FIA_UAU_EXT.2 The password-based authentication mechanism provides administrative users a locally based authentication mechanism.

5.2.8.3 Management: FIA_UAU_EXT.2

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

a) None

5.2.8.4 Audit: FIA_UAU_EXT.2

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

a) Minimal: All use of the authentication mechanism

FIA_UAU_EXT.2 Password-based Authentication Mechanism

Hierarchical to: No other components.

Dependencies: None

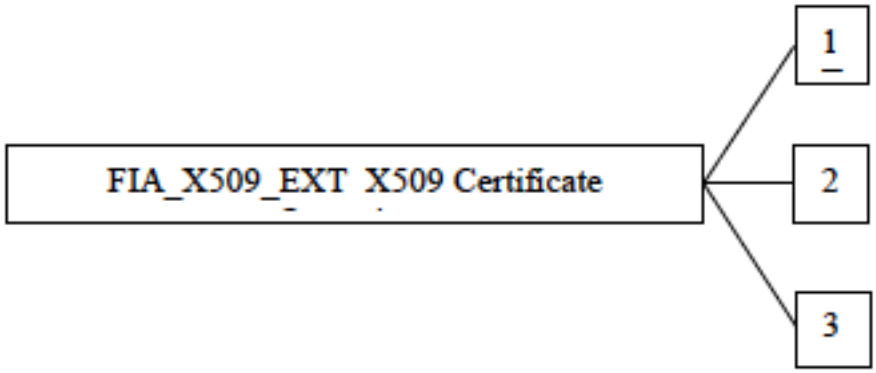
FIA_UAU_EXT.2.1 The TSF shall provide a local password-based authentication mechanism, [selection: [assignment: other authentication mechanism(s)], none] to perform administrative user authentication.

5.2.9 Authentication using X.509 certificates (Extended – FIA_X509_EXT)

5.2.9.1 Family Behavior

58 This family defines the behavior, management, and use of X.509 certificates for functions to be performed by the TSF. Components in this family require validation of certificates according to a specified set of rules, use of certificates for authentication for protocols and integrity verification, and the generation of certificate requests.

5.2.9.2 Component Leveling



59 FIA_X509_EXT.1 X509 Certificate Validation, requires the TSF to check and validate certificates in accordance with the RFCs and rules specified in the component.

60 FIA_X509_EXT.2 X509 Certificate Authentication, requires the TSF to use certificates to authenticate peers in protocols that support certificates, as well as for integrity verification and potentially other functions that require certificates.

61 FIA_X509_EXT.3 X509 Certificate Requests, requires the TSF to be able to generate Certificate Request Messages and validate responses.

5.2.9.3 Management: FIA_X509_EXT.1, FIA_X509_EXT.2, FIA_X509_EXT.3

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

a) Remove imported X.509v3 certificates

b) Approve import and removal of X.509v3 certificates

c) Initiate certificate requests

5.2.9.4 Audit: FIA_X509_EXT.1, FIA_X509_EXT.2, FIA_X509_EXT.3

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

a) Minimal: No specific audit requirements are specified.

FIA_X509_EXT.1 X.509 Certificate Validation

Hierarchical to: No other components

Dependencies: No other components

FIA_X509_EXT.1.1 The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [selection: the Online Certificate Status Protocol (OCSP) as specified in RFC 2560, a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3, Certificate Revocation List (CRL) as specified in RFC 5759 Section 5].
- The TSF shall validate the extendedKeyUsage field according to the following rules: [assignment: rules that govern contents of the extendedKeyUsage field that need to be verified].

FIA_X509_EXT.1.2 The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

FIA_X509_EXT.2 X.509 Certificate Authentication

Hierarchical to: No other components

Dependencies: No other components

FIA_X509_EXT.2.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [selection: IPsec, TLS, HTTPS, SSH, [assignment: other protocols], no protocols], and [selection: code signing for system software updates, code signing for integrity verification, [assignment: other uses], no additional uses].

FIA_X509_EXT.2.2 When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [selection: allow the administrator to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate].

FIA_X509_EXT.3 X.509 Certificate Requests

Hierarchical to: No other components

Dependencies: No other components

FIA_X509_EXT.3.1 The TSF shall generate a Certificate Request Message as specified by RFC 2986 and be able to provide the following information in the request: public key and [selection: device-specific information, Common Name, Organization, Organizational Unit, Country, [assignment: other information]].

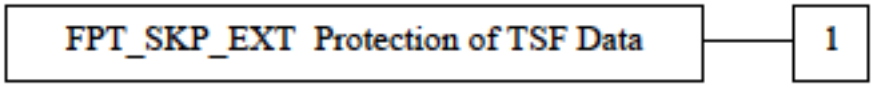
FIA_X509_EXT.3.2 The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

5.2.10 Protection of TSF Data (FPT_SKP_EXT)

5.2.10.1 Family Behavior

64 Components in this family address the requirements for managing and protecting TSF data such as cryptographic keys. This is a new family modelled after the FPT_PTD Class.

5.2.10.2 Component Leveling



65 FPT_SKP_EXT.1 Protection of TSF Data (for reading all symmetric keys), requires preventing symmetric keys from being read by any user or subject. It is the only component of this family.

5.2.10.3 Management: FPT_SKP_EXT.1

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

a) There are no management activities foreseen.

5.2.10.4 Audit: FPT_SKP_EXT.1

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

a) There are no auditable events foreseen.

FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys)

Hierarchical to: No other components.

Dependencies: No other components.

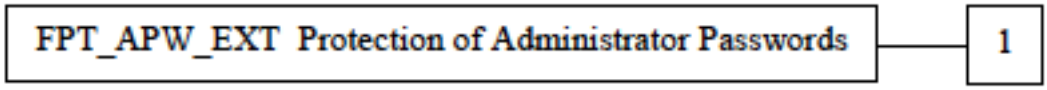
FPT_SKP_EXT.1.1 The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

5.2.11 Protection of Administrator Passwords (FPT_APW_EXT)

5.2.11.1 Family Behavior

68 Components in this family ensure that the TSF will protect plaintext credential data such as passwords from unauthorized disclosure.

5.2.11.2 Component Leveling



69 FPT_APW_EXT.1 Protection of administrator passwords requires that the TSF prevent plaintext credential data from being read by any user or subject.

5.2.11.3 Management: FPT_APW_EXT.1

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

a) No management functions.

5.2.11.4 Audit: FPT_APW_EXT.1

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

a) No audit necessary.

FPT_APW_EXT.1 Protection of Administrator Passwords

Hierarchical to: No other components.

Dependencies: No other components.

FPT_APW_EXT.1.1 The TSF shall store passwords in non-plaintext form.

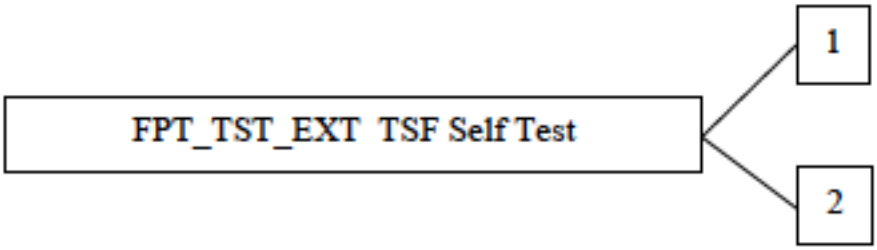
FPT_APW_EXT.1.2 The TSF shall prevent the reading of plaintext passwords.

5.2.12 TSF Testing (FPT_TST_EXT)

5.2.12.1 Family Behavior

72 Components in this family address the requirements for self-testing the TSF for selected correct operation.

5.2.12.2 Component Leveling



73 FPT_TST_EXT.1 TSF Self Test requires a suite of self tests to be run during initial start-up in order to demonstrate correct operation of the TSF.

74 FPT_TST_EXT.2 Self tests based on certificates applies when using certificates as part of self test, and requires that the self test fails if a certificate is invalid.

5.2.12.3 Management: FPT_TST_EXT.1, FPT_TST_EXT.2

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

a) No management functions.

5.2.12.4 Audit: FPT_TST_EXT.1, FPT_TST_EXT.2

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

a) Indication that TSF self test was completed

FPT_TST_EXT.1 TSF testing

Hierarchical to: No other components.

Dependencies: No other components.

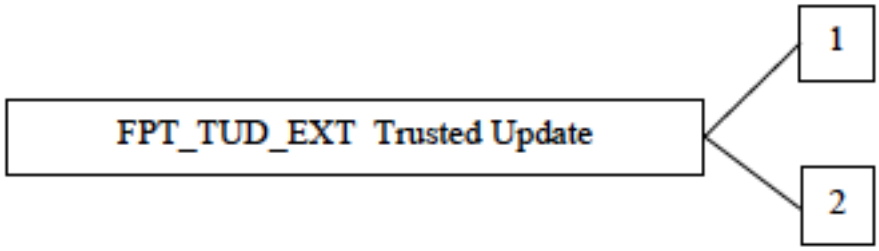
FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests [selection: during initial start-up (on power on), periodically during normal operation, at the request of the authorised user, at the conditions [assignment: conditions under which self-tests should occur]] to demonstrate the correct operation of the TSF: [assignment: list of self-tests run by the TSF].

5.2.13 Trusted Update (FPT_TUD_EXT)

5.2.13.1 Family Behavior

77 Components in this family address the requirements for updating the TOE firmware and/or software.

5.2.13.2 Component Leveling



78 FPT_TUD_EXT.1 Trusted Update requires management tools be provided to update the TOE firmware and software, including the ability to verify the updates prior to installation.

79 FPT_TUD_EXT.2 Trusted update based on certificates applies when using certificates as part of trusted update, and requires that the update does not install if a certificate is invalid. **Note:** Definition included but SFR not used in this ST.

5.2.13.3 Management: FPT_TUD_EXT.1, FPT_TUD_EXT.2

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

a) Ability to update the TOE and to verify the updates

b) Ability to update the TOE and to verify the updates using the digital signature capability (FCS_COP.1(2)) and [selection: no other functions, [assignment: other cryptographic functions (or other functions) used to support the update capability]]

c) Ability to update the TOE, and to verify the updates using [selection: digital signature, published hash, no other mechanism] capability prior to installing those updates

5.2.13.4 Audit: FPT_TUD_EXT.1, FPT_TUD_EXT.2

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

a) Initiation of the update process.

b) Any failure to verify the integrity of the update.

FPT_TUD_EXT.1 Trusted update

Hierarchical to:	No other components.
Dependencies:	[FCS_COP.1(1) Cryptographic operation (for cryptographic signature), or FCS_COP.1(3) Cryptographic operation (for cryptographic hashing)]
FPT_TUD_EXT.1.1	The TSF shall provide [assignment: authorised users] the ability to query the currently executing version of the TOE firmware/software as well as the most recently installed version of the TOE firmware/software.
FPT_TUD_EXT.1.2	The TSF shall provide [assignment: authorised users] the ability to manually initiate updates to TOE firmware/software and [selection: support automatic checking for updates, support automatic updates, no other update mechanism].

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

FPT_TUD_EXT.2 Trusted update based on certificates

Hierarchical to: No other components

Dependencies: FPT_TUD_EXT.1

FPT_TUD_EXT.2.1 The TSF shall not install an update if the code signing certificate is deemed invalid.

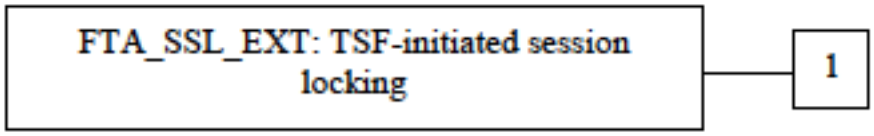
FPT_TUD_EXT.2.2 When the certificate is deemed invalid because the certificate has expired, the TSF shall [selection: allow the administrator to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate].

5.2.14 TSF-initiated Session Locking (FTA_SSL_EXT)

5.2.14.1 Family Behavior

82 Components in this family address the requirements for TSF-initiated and user-initiated locking, unlocking, and termination of interactive sessions. The extended FTA_SSL_EXT family is based on the FTA_SSL family.

5.2.14.2 Component Leveling



83 FTA_SSL_EXT.1 TSF-initiated session locking, requires system initiated locking of an interactive session after a specified period of inactivity. It is the only component of this family.

5.2.14.3 Management: FTA_SSL_EXT.1

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

a) Specification of the time of user inactivity after which lock-out occurs for an individual user.

5.2.14.4 Audit: FTA_SSL_EXT.1

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

a) Any attempts at unlocking an interactive session.

FTA_SSL_EXT.1 TSF-initiated Session Locking

Hierarchical to: No other components

Dependencies: FIA_UAU.1 Timing of authentication

FTA_SSL_EXT.1.1 The TSF shall, for local interactive sessions, [selection:

- lock the session - disable any activity of the user's data access/display devices other than unlocking the session, and requiring that the administrator re-authenticate to the TSF prior to unlocking the session;
- terminate the session]

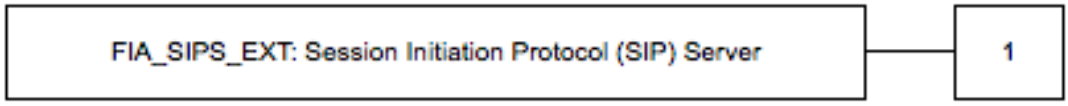
after a Security Administrator-specified time period of inactivity.

5.2.15 Session Initiation Protocol (SIP) Server (FIA_SIPS_EXT)

5.2.15.1 Family Behavior

86 This family provides requirements that address the Session Initiation Protocol (SIP) sever requirements.

5.2.15.2 Component Leveling



87 FIA_SIPS_EXT.1addresses SIP server requirements.

5.2.15.3 Management: FIA_SIPS_EXT.1

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

a) None

5.2.15.4 Audit: FIA_SIPS_EXT.1

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

a) None

FIA_SIPS_EXT.1 Session Initiation Protocol (SIP) Server

FIA_SIPS_EXT.1.1 The TSF shall implement the Session Initiation Protocol (SIP) that complies with RFC 3261 using the Session Description Protocol (SDP) complying with RFC 4566 to describe the multimedia session that will be used to carry the VOIP traffic.

FIA_SIPS_EXT.1.2 The TSF shall require password authentication for SIP REGISTER function requests as specified in section 22 of RFC 3261.

FIA_SIPS_EXT.1.3 The TSF shall support SIP authentication passwords that contain at least [assignment: positive integer of 8 or more] characters in the set of {upper case characters, lower case characters, numbers, and the following special characters: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")", and [assignment: other supported special characters]}.

5.3 Functional Requirements

Table 13: Summary of SFRs

Requirement	Title
FAU_GEN.1	Audit Data Generation
FAU_GEN.2	User Identity Association
FAU_STG_EXT.1	Protected Audit Event Storage
FCS_CKM.1	Cryptographic Key Generation
FCS_CKM.2	Cryptographic Key Establishment
FCS_CKM.4	Cryptographic Key Destruction
FCS_COP.1(1)	Cryptographic Operation (AES Data Encryption/Decryption)
FCS_COP.1(2)	Cryptographic Operation (Signature Generation and Verification)
FCS_COP.1(3)	Cryptographic Operation (Hash Algorithm)
FCS_COP.1(4)	Cryptographic Operation (Keyed Hash Algorithm)
FCS_SSHS_EXT.1	SSH Server Protocol
FCS_TLSC_EXT.2	TLS Client Protocol with authentication
FCS_TLSS_EXT.2	TLS Server Protocol with mutual authentication
FCS_RBG_EXT.1	Random Bit Generation
FIA_PMG_EXT.1	Password Management
FIA_SIPS_EXT.1	Session Initiation Protocol (SIP) Server
FIA_UIA_EXT.1	User Identification and Authentication
FIA_UAU_EXT.2	Password-based Authentication Mechanism
FIA_UAU.7	Protected Authentication Feedback
FIA_X509_EXT.1	X.509 Certificate Validation
FIA_X509_EXT.2	X.509 Certificate Authentication
FIA_X509_EXT.3	X.509 Certificate Requests
FMT_MOF.1(1)/Trusted Update	Management of security functions behaviour

Requirement	Title
FMT_MOF.1(1)/AdminAct	Management of security functions behaviour
FMT_MTD.1	Management of TSF Data
FMT_MTD.1/AdminAct	Management of TSF Data
FMT_SMF.1	Specification of Management Functions
FMT_SMR.2	Restrictions on Security Roles
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all symmetric keys)
FPT_APW_EXT.1	Protection of Administrator Passwords
FPT_TST_EXT.1	TSF testing
FPT_TUD_EXT.1	Extended: Trusted update
FPT_STM.1	Reliable Time Stamps
FTA_SSL_EXT.1	TSF-initiated Session Locking
FTA_SSL.3	TSF-initiated Termination
FTA_SSL.4	User-initiated Termination
FTA_TAB.1	Default TOE Access Banners
FTP_ITC.1(1)	Inter-TSF trusted channel
FTP_ITC.1(2)	Inter-TSF Trusted Channel (TLS/SIP)
FTP_ITC.1(3)	Inter-TSF Trusted Channel (Protection from Modification or Disclosure – SIP Server)
FTP_TRP.1	Trusted Path

5.3.1 Security Audit (FAU)

FAU_GEN.1 Audit Data Generation

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

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

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

c) All administrative actions comprising:

- *Administrative login and logout (name of user account shall be logged if individual user accounts are required for administrators).*
- *Security related configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).*
- *Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).*
- *Resetting passwords (name of related user account shall be logged).*
- *Starting and stopping services (if applicable)*
- no other actions;

d) *Specifically defined auditable events listed in ~~Table 1~~* ***the table below****.*

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	None.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1(1)	None.	None.
FCS_COP.1(2)	None.	None.
FCS_COP.1(3)	None.	None.
FCS_COP.1(4)	None.	None.
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
FCS_TLSC_EXT.2	Failure to establish a TLS Session	Reason for failure
FCS_TLSS_EXT.2	Failure to establish a TLS Session	Reason for failure
FCS_RBG_EXT.1	None.	None.
FIA_PMG_EXT.1	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FIA_SIPS_EXT.1	Session establishment with peer	Source and destination addresses Source and destination ports TOE Interface
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.
FIA_X509_EXT.1	Unsuccessful attempt to validate a certificate	Reason for failure
	Establishment of session with CA	Source and destination addresses Source and destination ports TOE Interface
FIA_X509_EXT.2	None.	None.
FIA_X509_EXT.3	None.	None.
FMT_MOF.1(1)/Trusted Update	Any attempt to initiate a manual update	None.
FMT_MTD.1	All management activities of TSF data.	None.
FMT_MTD.1/AdminAct	Modification, deletion, generation/import of cryptographic keys.	None.
FMT_MOF.1(1)/AdminAct	Modification of the behaviour of the TSF.	None.
FMT_SMF.1	None.	None.
FMT_SMR.2	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_TST_EXT.1	Failure of AIDE integrity test	None.

Requirement	Auditable Events	Additional Audit Record Contents
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	No additional information.
FPT_TUD_EXT.2	Failure of update	Reason for failure (including identifier of invalid certificate)
FPT_STM.1	Changes to time.	The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FTA_SSL_EXT.1	Any attempts at unlocking of an interactive session.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1(1)	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_ITC.1(2)		
FTP_ITC.1(3)		
FTP_TRP.1	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	Identification of the claimed user identity.

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

a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, *information specified in column three of ~~Table 1~~* ***the table above****.*

FAU_GEN.2 User Identity Association

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

FAU_STG_EXT.1 Protected Audit Event Storage

FAU_STG_EXT.1.1 The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC.1.

FAU_STG_EXT.1.2 The TSF shall be able to store generated audit data on the TOE itself.

FAU_STG_EXT.1.3 The TSF shall drop new audit data when the local storage space for audit data is full.

5.3.2 Cryptographic Support (FCS)

FCS_CKM.1 Cryptographic Key Generation

FCS_CKM.1.1 The TSF shall generate **asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm:

- **RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;**
- **ECC schemes using "NIST curves" P-256, P-384 that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;**

~~]and specified cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards]~~.

FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1 The TSF shall **perform** cryptographic **key establishment** in accordance with a specified cryptographic key **establishment** method:

- **RSA-based key establishment schemes that meets the following: NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography";**
- **Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";**

~~that meets the following: [assignment: list of standards].~~

FCS_CKM.4 Cryptographic Key Destruction

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method:

- For plaintext keys in volatile storage, the destruction shall be executed by a single overwrite consisting of zeroes;
- For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that:

- logically addresses the storage location of the key and performs a single-pass overwrite consisting of zeroes;

that meets the following: *No Standard*.

FCS_COP.1(1) Cryptographic Operation (AES Data Encryption/Decryption)

FCS_COP.1.1(1) The TSF shall perform *encryption/decryption* in accordance with a specified cryptographic algorithm *AES used in* CBC, GCM *mode* and cryptographic key sizes 128 bits, 256 bits that meet the following: *AES as specified in ISO 18033-3,* CBC as specified in ISO 10116, GCM as specified in ISO 19772.

FCS_COP.1(2) Cryptographic Operation (Signature Generation and Verification)

FCS_COP.1.1(2) The TSF shall perform *cryptographic signature services* (generation and verification) in accordance with a specified cryptographic algorithm:

- RSA Digital Signature Algorithm and cryptographic key sizes **(modulus)** *2048 bits*,
- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes: *256 bits, 384 bits.*

that meet the following:

- For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSAPKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,
- For ECDSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" P-256, P-384; ISO/IEC 14888-3, Section 6.4.

FCS_COP.1(3) Cryptographic Operation (Hash Algorithm)

FCS_COP.1.1(3) The TSF shall perform *cryptographic hashing services* in accordance with a specified cryptographic algorithm SHA-1, SHA-256, SHA-384, SHA-512 ~~and cryptographic key sizes [assignment: cryptographic key sizes]~~ that meet the following: *ISO/IEC 10118-3:2004*.

FCS_COP.1(4) Cryptographic Operation (Keyed Hash Algorithm)

FCS_COP.1.1(4) The TSF shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384 and cryptographic key sizes *160, 256, 384* a**nd message digest sizes 160, 256, 384 *bits*** that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

FCS_RBG_EXT.1 Random Bit Generation

FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using CTR_DRBG (AES).

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from *1* hardware-based noise source with a minimum of 256 bits of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 “Security Strength Table for Hash Functions”, of the keys and hashes that it will generate.

FCS_SSHS_EXT.1 SSH Server Protocol

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254, and 5656, 6668.

FCS_SSHS_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, password based.

FCS_SSHS_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than *32768* bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: aes128-cbc, aes256-cbc.

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH transport implementation uses ecdsa-sha2-nistp256 and no other public key algorithms as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses hmac-sha2-256 and no other MAC algorithms as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7 The TSF shall ensure that ecdh-sha2-nistp256 and no other methods are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHS_EXT.1.8 The TSF shall ensure that within SSH connections the same session keys are used for a threshold of no longer than one hour, and no more than one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

FCS_TLSC_EXT.2 TLS Client Protocol with authentication

FCS_TLSC_EXT.2.1 The TSF shall implement TLS 1.2 (RFC 5246) supporting the following ciphersuites:

Mandatory Ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268

Optional Ciphersuites:

- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

FCS_TLSC_EXT.2.2 The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.2.3 The TSF shall only establish a trusted channel if the peer certificate is valid.

FCS_TLSC_EXT.2.4 The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: P-256, P-384 and no other curves.

FCS_TLSC_EXT.2.5 The TSF shall support mutual authentication using X.509v3 certificates.

FCS_TLSS_EXT.2 TLS Server Protocol with mutual authentication

FCS_TLSS_EXT.2.1 The TSF shall implement *TLS 1.2 (RFC 5246)* supporting the following ciphersuites:

Mandatory Ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268

Optional Ciphersuites:

- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

FCS_TLSS_EXT.2.2 The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, and TLS 1.1.

FCS_TLSS_EXT.2.3 The TSF shall generate key establishment parameters using RSA with key size 2048 bits and no other size and over NIST curves *secp256r1, secp384r1* and no other curves.

FCS_TLSS_EXT.2.4 The TSF shall support mutual authentication of TLS clients using X.509v3 certificates.

FCS_TLSS_EXT.2.5 The TSF shall not establish a trusted channel if the peer certificate is invalid.

FCS_TLSS_EXT.2.6 The TSF shall not establish a trusted channel if the distinguished name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match the expected identifier for the peer.

5.3.3 Identification and Authentication (FIA)

FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1.1 The TSF shall provide the following password management capabilities for administrative passwords:

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", "_ ", "<", ">", "{", "}", "[", "]", ":", ";", "|", "+", "~", ",", "`",

b) Minimum password length shall be settable by the Security Administrator, and shall support passwords of 15 characters or greater.

FIA_SIPS_EXT.1 Session Initiation Protocol (SIP) Server

FIA_SIPS_EXT.1.1 The TSF shall implement the Session Initiation Protocol (SIP) that complies with RFC 3261 using the Session Description Protocol (SDP) complying with RFC 4566 to describe the multimedia session that will be used to carry the VOIP traffic.

FIA_SIPS_EXT.1.2 The TSF shall require password authentication for SIP REGISTER function requests as specified in section 22 of RFC 3261.

FIA_SIPS_EXT.1.3 The TSF shall support SIP authentication passwords that contain at least *8* characters in the set of {upper case characters, lower case characters, numbers, and the following special characters: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")", and *no additional characters*.

FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1.1 The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- no other actions

FIA_UIA_EXT.1.2 The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2.1 The TSF shall provide a local password-based authentication mechanism, **and other authentications mechanisms:** none to perform administrative user authentication.

FIA_UAU.7 Protected Authentication Feedback

FIA_UAU.7.1 The TSF shall provide only *obscured feedback* to the administrative user while the authentication is in progress at the local console.

FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1.1 The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using Certificate Revocation List (CRL) as specified in RFC 5759 Section 5.
- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - *Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.*
    - *Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*
    - *Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.*
    - *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.*

FIA_X509_EXT.1.2 The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for TLS and no additional uses.

FIA_X509_EXT.2.2 When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall accept the certificate.

FIA_X509_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.3.1 The TSF shall generate a Certificate Request Message as specified by RFC 2986 and be able to provide the following information in the request: public key and Common Name, Organization, Organizational Unit, Country.

FIA_X509_EXT.3.2 The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

5.3.4 Security Management (FMT)

FMT_MOF.1(1)/TrustedUpdate Management of security functions behaviour

FMT_MOF.1.1(1)/TrustedUpdate The TSF shall restrict the ability to enable the functions *to perform manual update to Security Administrators*.

FMT_MOF.1(1)/AdminAct Management of security functions behaviour

FMT_MOF.1.1(1)/AdminAct The TSF shall restrict the ability to modify the behaviour of the functions *TOE Security Functions to Security Administrators*.

FMT_MTD.1 Management of TSF Data

FMT_MTD.1.1 The TSF shall restrict the ability to *manage* the *TSF data* to *Security Administrators*.

FMT_MTD.1/AdminAct Management of TSF data

FMT_MTD.1.1/AdminAct The TSF shall restrict the ability to *modify, delete, generate/import the cryptographic keys **and certificates*** to *Security Administrators*.

FMT_SMF.1 Specification of Management Functions

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

- *Ability to administer the TOE locally and remotely;*
- *Ability to configure the access banner;*
- *Ability to configure the session inactivity time before session termination or locking;*
- *Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;*
- *Ability of a Security Administrator to:*
  - *Configure the SIP;*
  - *Configure mechanisms implemented with respect to FCS_TLSS_EXT.2;*
  - *Import X.509v3 certificates;*
  - *Configure SIP client password;*
  - *Ability to configure the cryptographic functionality;*
  - *Ability to configure thresholds for SSH rekeying.*

FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1 The TSF shall maintain the roles:

- *Security Administrator.*

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

FMT_SMR.2.3 The TSF shall ensure that the conditions

- *The Security Administrator role shall be able to administer the TOE locally;*
- *The Security Administrator role shall be able to administer the TOE remotely*

are satisfied.

5.3.5 Protection of the TSF (FPT)

FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys)

FPT_SKP_EXT.1.1 The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

FPT_APW_EXT.1 Protection of Administrator Passwords

FPT_APW_EXT.1.1 The TSF shall store passwords in non-plaintext form.

FPT_APW_EXT.1.2 The TSF shall prevent the reading of plaintext passwords.

FPT_TST_EXT.1 TSF testing

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests during initial start-up (on power on) to demonstrate the correct operation of the TSF:

- *OpenSSL module self-tests*
- *TOE software integrity test*

FPT_TUD_EXT.1 Extended: Trusted update

FPT_TUD_EXT.1.1 The TSF shall provide *Security Administrators* the ability to query the currently executing version of the TOE firmware/software as well as the most recently installed version of the TOE firmware/software.

FPT_TUD_EXT.1.2 The TSF shall provide *Security Administrators* the ability to manually initiate updates to TOE firmware/software and no other update mechanism.

FPT_TUD_EXT.1.3 The TSF shall provide a means to authenticate firmware/software updates to the TOE using a digital signature mechanism and no other functions prior to installing those updates.

FPT_STM.1 Reliable Time Stamps

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

5.3.6 TOE Access (FTA)

FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1.1 The TSF shall, for local interactive sessions:

- terminate the session

after a Security Administrator-specified time period of inactivity.

FTA_SSL.3 TSF-initiated Termination

FTA_SSL.3.1 **Refinement:** The TSF shall terminate **a remote** interactive session after a *Security Administrator-configurable time interval of session inactivity*.

FTA_SSL.4 User-initiated Termination

FTA_SSL.4.1 **Refinement:** The TSF shall allow **Administrator**-initiated termination of the **Administrator's** own interactive session.

FTA_TAB.1 Default TOE Access Banners

FTA_TAB.1.1 **Refinement:** Before establishing **an administrative user** session the TSF shall **display a Security Administrator-specified** advisory **notice and consent** warning message regarding use of the TOE.

5.3.7 Trusted path/channels (FTP)

FTP_ITC.1(1) Inter-TSF trusted channel

FTP_ITC.1.1(1) The TSF shall be **capable of using TLS** to provide a trusted communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, *database server, update server*** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

FTP_ITC.1.2(1) The TSF shall permit **the TSF, or the authorized IT entities** to initiate communication via the trusted channel.

FTP_ITC.1.3(1) The TSF shall initiate communication via the trusted channel for

- *Syslog*
- *Accessing the database server*
- *Accessing the update server*

FTP_ITC.1(2) Inter-TSF Trusted Channel (TLS/SIP)

FTP_ITC.1.1(2) **Refinement:** The TSF shall provide a communication channel between itself and **a SIP Client using TLS and no other protocol as specified in FCS_TLSS_EXT.2 only**, that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification and ~~or~~ disclosure.

FTP_ITC.1.2(2) The TSF shall permit the TSF Client to initiate communication via the trusted channel

FTP_ITC.1.3(2) The TSF Client shall initiate communication via the trusted channel for *all communications with the SIP serve*r.

FTP_ITC.1(3) Inter-TSF Trusted Channel (Protection from Modification or Disclosure – SIP Server)

FTP_ITC.1.1(3) **Refinement:** The TSF shall provide a communication channel between itself and another **SIP Server using TLS** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification and disclosure.

FTP_ITC.1.2(3) The TSF shall permit **the TOE or the peer SIP Server** to initiate communication via the trusted channel

FTP_ITC.1.3(3) The TSF shall initiate communication via the trusted channel to *pass SIP data to a SIP Server Peer*.

FTP_TRP.1 Trusted Path

FTP_TRP.1.1 The TSF shall be **capable of using SSH** to provide a communication path between itself and **authorized remote administrators** that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from *disclosure and provides detection of modification of the channel data*.

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

FTP_TRP.1.3 The TSF shall require the use of the trusted path for **initial administrator authentication and all remote administration actions**.

5.4 Assurance Requirements

90 The TOE security assurance requirements are summarized in Table 14.

Table 14: Assurance Requirements

Assurance Class	Components	Description
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative User Guidance
Life Cycle Support	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM Coverage
Security Target Evaluation	ASE_CCL.1	Conformance Claims
	ASE_ECD.1	Extended Components Definition
	ASE_INT.1	ST Introduction
	ASE_OBJ.1	Security Objectives for the operational environment
	ASE_REQ.1	Stated Security Requirements
	ASE_SPD.1	Security Problem Definition
	ASE_TSS.1	TOE Summary Specification
Tests	ATE_IND.1	Independent Testing - conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability Analysis

6 TOE Summary Specification

6.1 Session Initiation Protocol (SIP) Server

91 The TOE implements SIP with the Session Description Protocol (SDP) to describe multimedia sessions that are used to carry VoIP traffic. The TOE requires VoIP clients to authenticate with a password (of 8 or more characters) for SIP REGISTER function requests.

Table 15: SIP Server SFRs

SFR	Fulfilment
FIA_SIPS_EXT.1	**Client registration to SecuSUITE (external to the TOE)** Before a SecuSUITE client can exchange messages with the TOE, it must first be registered to the SecuSUITE infrastructure via the SCA Server. This initial client registration, which occurs externally to the TOE, is briefly described below to provide context to the reader: 1. SecuSUITE administrator adds a user to the SecuSUITE via the Admin Portal which generates activation and validation codes that are delivered to the user via some out of band method (e.g. email / SMS) 2. User downloads the SecuSUITE client application from supported app store and launches client app. The user is prompted to create an application password. 3. Client app running on mobile device prompts the user to enter the activation code and initiates a TLS connection to SCA Server. 4. SCA Server validates client for registration via activation code (this is not the SIP password) 5. Client generates multiple certificate signing requests and submits to SCA Server 6. SCA server's embedded CA creates, signs and returns the certificates 7. Client gets its SIP settings from SCA server (which retrieves settings from the database server). Settings include: a. E.164 telephone number (SIP alias) b. SIP Server URI c. TLS version (TLS 1.2) d. SIP domain to which client belongs e. SIP user name f. SIP password (displayed to the user once – the user must remember and acknowledge the password) g. Validation code (not displayed to user) 8. User performs the following: a. Come up with and enter password for client application b. Enter unique activation code

SFR	Fulfilment
	c. Enter validation code d. Memorize / acknowledge SIP password **Note:** The above process is occurs only for initial client provisioning. The SCA Server is not involved in client authentication to the SIP Server with the SIP password. User must enter the SIP password on start / restart of the application (e.g. device reboot). The TOE calculates the H(A1) from the SIP password for digest access authentication to the SIP server as described below. **Client Registration with SIP Server** When: Whenever authenticated and configured client connects to the SIP Server, e.g. after: • client app was installed and SCA procedure was successfully passed, or • client was restarted, or • client had lost TLS connection to SIP server (e.g. because of network change or problems) Procedure: • Client opens two-way authenticated TLS session with SIP server • Client registers using SIP REGISTER requests regularly with SIP server for keeping the TLS connection in the firewall (of the IP network to which the client is currently connected) open • In times of inactivity the firewall would otherwise close the port again which it had allocated for the client's TLS connection, and any further SIP messages of the server would then be blocked and would not reach the client anymore • SIP server authenticates client's SIP REGISTER request messages with SIP username and password / digest access authentication. **Digest Access Authentication** • The SIP username and password are used to authenticate SIP REGISTER and INVITE messages using digest access authentication per RFC 3261 as follows: • Client and server have a shared secret (H(A1) of SIP password) • Client sends request message to server • Server rejects request with request message containing challenge ("nonce") • Client calculates digest from challenge and H(A1) of SIP password • Client sends request message again with request message now containing digest • Server also calculates digest and compares this with value received from client • If digest values match, server accepts request

SFR	Fulfilment
	Call Setup Preconditions: • Client A (“Alice”) and client B (“Bob”) have registered with SCA server. • Alice and Bob have running TLS sessions with the SIP server Alice calls Bob: • The SIP Server routes SIP messages between Alice and Bob. • Alice and Bob do not exchange media packets (RTP/RTCP) directly. The SecuSUITE encompasses an RTP proxy which works as an RTP bridge. Alice sends her media packets to the RTP proxy which forwards them to Bob, and vice versa. During connection signalling, the SIP server sets up the RTP/RTCP packet bridging in the RTP proxy for this connection. The messages are as follows (refer to Annex A: Call Signaling diagram): • Alice’s SIP INVITE message includes: o SIP username and password for digest access authentication o Alice‘s VoIP Encryption Certificate • Bob‘s SIP 200 OK message includes within the SDP: o Bob‘s VoIP Encryption Certificate o Bob’s SDP message o Bob‘s SRTP master uplink key and salt (i.e. the key and salt Bob is using when sending RTP and RTCP packets to Alice, see (RFC4568, 2006) section 5.1.1) in a message block containing a CMS EnvelopedData ASN.1 structure. • Alice’s SIP ACK includes: o Alice’s SDP message o Alice’s SRTP master uplink key and salt (i.e. the key and salt Alice is using when sending RTP and RTCP packets to Bob; similar encoding as Bob) **User Plane (Media)** The main purpose of the RTP Proxy is to make the communication between SIP user agents behind NAT/NAPT possible: • Typically, the client has an internal (non-routable) IP address and will select some UDP port for RTP and another one for RTCP. NAPT will change the IP address and UDP ports to external values. The internal values however appear in the SDP, and the remote client would use them as destination IP address and ports, which would not work. The solution is to replace the IP address and ports in the SDP: The new IP address is a routable IP address of an RTP proxy, and the RTP/RTCP ports are replaced and used as session identifiers. This replacement happens in the SIP server during call establishment: • When the SIP server receives the first SIP message with SDP content during call setup (e.g. 200 OK), it extracts the Call-ID, selects an RTP

SFR	Fulfilment
	proxy, and sends the Call ID to this RTP Proxy using the RTPproxy Control Protocol. • The RTP Proxy creates a new session by allocating randomly two subsequent unused UDP ports from a range of UDP ports to that session, and returns these port numbers to the SIP server via the RTP Proxy Control Protocol. The first port is for RTP, and the second one for RTCP. • After receiving the reply from the RTP Proxy, the SIP server replaces the RTP and RTCP media IP addresses and UDP ports in the SDP content of the message with the RTP Proxy IP address and the UDP ports the RTP Proxy has allocated. • Then the SIP server forwards this modified SIP message as usually to the intended destination. • When the SIP server receives a SIP follow-up message (e.g. ACK) containing SDP information from the other peer, it sends again the Call-ID to the RTP proxy via the RTPproxy Control Protocol. • Using the Call-ID as a key, the RTP proxy performs a lookup among existing sessions, allocates randomly another pair of subsequent UDP ports to this session and returns these port numbers to the SIP server. • After receiving the second pair of port numbers from the RTP proxy, the SIP server replaces the media IP address and Ports in the SDP content of the SIP follow-up message so that it now also points to the RTP proxy. The SIP server forwards the SIP message as usually to the intended destination. • For RTP, the RTP proxy now listens on the two ports it has allocated for that session and waits for receiving at least one UDP message from Alice and one from Bob. When such a packet is received, the proxy fills one of two IP address/UDP port structures associated to this call with the source IP address and the source UDP port of that packet. When both structures are filled in, the RTP proxy starts relaying UDP/RTP packets between the Alice and Bob. • The same happens for RTCP. • The RTP proxy tracks idle time for each of the existing sessions (i.e. the time within which there were no packets relayed), and automatically cleans up a sessions whose idle times exceed a specified value (e.g. 60 seconds). **Call Termination** Users can terminate an ongoing call anytime by pushing the “End call” button. The client sends a SIP BYE message and the other party confirms with a SIP OK message. The SIP server then terminates the SRTP session by sending a Delete message for that call to the RTP Proxy. Clients will also terminate a call when no RTP data is received for more than 10 seconds. **Session Description Protocol (SDP)** The TOE uses SDP to describe multimedia sessions that are used to carry VoIP traffic. Refer to Annex B: SDP Example for a description of the TOE SDP implementation.

6.2 Protected Communications

92 The TOE protects the integrity and confidentiality of communications with VoIP clients, remote administrators, remote audit servers, update server, database server and peer SIP servers.

Table 16: Protected Communications SFRs

SFR	Fulfilment

FCS_TLSS_EXT.2	**Summary**
	The TOE uses TLS (OpenSSL) for communication with SIP clients. This implementation has the following characteristics:
	• TLS 1.2 allowed, connection requests with other SSL/TLS versions denied.
	• Supported ciphersuites:
	  ○ TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
	  ○ TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
	  ○ TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
	Key Agreement and Server Authentication
	Server and client establish a shared TLS premaster secret with the TLS key exchange
	All key exchange methods use the same algorithm then to convert the premaster secret into the master secret
	Together with client random (in ClientHello message) and server random (in ServerHello message), client and server generate session encryption and MAC keys from the master secret with the TLS PRF.
	• RSA:
	  ○ The TOE uses RSA encryption of the premaster secret in accordance with (RFC5246, 2008) section 7.4.7.1
	  ○ SIP Server sends its TLS certificate to the client (server Certificate message). This contains the SIP server's public key
	  ○ The client uses the contained public key to encrypt the TLS premaster secret (client and SIP server calculate TLS master secret and session keys from this) before sending this to the SIP server.
	  ○ The SIP server can only decrypt the TLS premaster secret and finalize the TLS handshake when it possesses the corresponding private key.
	  ○ The client thus authenticates the SIP server
	  ○ The server's certificate contains a public RSA key authorized for encryption

SFR	Fulfilment

	• ECDHE_ECDSA
	  ○ SecuSUITE uses ECDHE_ECDSA in accordance with (RFC4492, 2006) section 2
	  ○ SIP Server sends its TLS certificate to the client (server Certificate message). This contains the server's static public key (an elliptic curve point) authorized for ECDH and is signed with ECDSA.
	  ○ Server creates an ephemeral elliptic-curve key-pair
	  ○ Using ECDSA, server signs the ephemeral public key (an elliptic curve point) and the corresponding elliptic curve parameters with its static private key (an integer) corresponding to the static public key in its TLS certificate.
	  ○ The TLS ciphersuites do not specify a concrete hash function for ECDSA-based server authentication. SecuSUITE selects SHA-256.
	  ○ Server sends the signed ephemeral public key and the elliptic curve parameters to the client (ServerKeyExchange message)
	  ○ Client verifies signature (with the server's static public key from the server's certificate)
	  ○ Client generates also an ephemeral key pair on the same curve as the server and sends its ephemeral public key (an elliptic curve point) to the server (ClientKeyExchange message)
	  ○ Server and client execute ephemeral ECDH. The resulting curve point is their shared secret.
	  ○ Server and client generate the TLS premaster secret from their secret curve point: The x-coordinate is represented as an octet string and fed into a KDF.
	• ECDHE_RSA
	  ○ The TOE uses ECDHE_RSA in accordance with (RFC5289, 2008) section 2.4
	  ○ Similar to ECDHE_ECDSA except that the server signs its ephemeral public key and the elliptic curve parameters with RSASSA-PKCS1-v1_5.
	  ○ The TLS ciphersuites do not specify a concrete hash function for RSA-based server authentication. SecuSUITE selects SHA-256.
	  ○ Server has a certificate signed with RSA containing a public RSA key authorized for signing
	Client Authentication
	The server requests client authentication with the ClientCertificateRequest message, and the client answers then with ClientCertificate and CertificateVerify messag

SFR	Fulfilment
	TLS authentication is certificate-based (see X.509 requirements) using two-way authentication (in the CertificateVerify message, client sends a signed hash of all the TLS messages up to this point): • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256: o Server requests and client provides authentication with ECDSA and SHA-256 o Client needs a certificate signed with ECDSA containing an ECDSA public key authorized for signing o Certificate complies with (RFC5759, 2010) • TLS_RSA_WITH_AES_128_CBC_SHA: o Server requests and client provides authentication with RSASSA-PKCS1-v1_5 and SHA-1 o Client needs a certificate with an RSA public key applicable for signing with RSASSA-PKCS1-v1_5 and hashing with SHA-1 • TLS_ECDHE_RSA_WITH _AES_256_GCM_SHA384: o Server requests and client provides authentication with RSASSA-PKCS1-v1_5 and SHA-256 o Client needs a certificate with an RSA public key applicable for signing with RSASSA-PKCS1-v1_5 and hashing with SHA-256 **Certificate Validation** Certificates are validated in accordance with FIA_X509 requirements in section 6.7.
FCS_TLSC_EXT.2	**Summary** The TOE uses TLS (OpenSSL) for communication with a Syslog Server, Update Server, database server and peer SIP Server. This implementation in which the TOE is a TLS client has the following characteristics: • TLS v1.2 • Supported ciphersuites: o TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268 o TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289 o TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 o Supported NIST curves: P-256 and P-384 (enabled by default) **Key Agreement and Server Authentication** Server and client (TOE) establish a shared TLS premaster secret with the TLS key exchange

<table>
<tr><th>SFR</th><th>Fulfilment</th></tr>
<tr><td></td><td>
All key exchange methods use the same algorithm then to convert the premaster secret into the master secret.

Together with client random (in ClientHello message) and server random (in ServerHello message), client and server generate session encryption and MAC keys from the master secret with the TLS PRF.

• RSA:
  ○ The TOE uses RSA encryption of the premaster secret in accordance with (RFC5246, 2008) section 7.4.7.1
  ○ Server sends its TLS certificate to the TOE (server Certificate message). This contains the server's public key
  ○ The TOE uses the contained public key to encrypt the TLS premaster secret (TOE and server calculate TLS master secret and session keys from this) before sending this to the server.
  ○ The server can only decrypt the TLS premaster secret and finalize the TLS handshake when it possesses the corresponding private key.
  ○ The TOE thus authenticates the server
  ○ The server's certificate contains a public RSA key authorized for encryption

Client Authentication

The server requests client (TOE) authentication with the ClientCertificateRequest message, and the TOE answers then with ClientCertificate and CertificateVerify message.

TLS authentication is certificate-based (see X.509 requirements) using two-way authentication (in the CertificateVerify message, the TOE sends a signed hash of all the TLS messages up to this point):

• Server requests and the TOE provides authentication with RSASSA-PKCS1-v1_5 and SHA-1
• The TOE uses a certificate with an RSA public key applicable for signing with RSASSA-PKCS1-v1_5 and hashing with SHA-1

Certificate Validation

Certificates are validated in accordance with FIA_X509 requirements in section 6.7. The TOE does not support certificate pinning. IP addresses in certificates are supported and wildcards are supported.
</td></tr>
<tr><td>FCS_SSHS_EXT.1</td><td>
The SIP server supports these SSH algorithms:
<table>
<tr><th>Algorithm</th><th>Supported methods</th><th>Reference</th><th>Usage</th></tr>
<tr><td>KEX algorithm</td><td>ecdh-sha2-nistp256</td><td>(RFC5656, 2009)</td><td>Generation of session keys for encryption</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>SFR</th><th>Fulfilment</th></tr>
<tr><td></td><td>
<table>
<tr><td>server host key algorithm</td><td>ecdsa-sha2-nistp256</td><td>(RFC5656, 2009)</td><td>SIP server authentication</td></tr>
<tr><td>Encryption</td><td>AES128-CBC, AES256-CBC</td><td></td><td>SSH Transport Layer Encryption</td></tr>
<tr><td>MAC algorithm</td><td>HMAC-SHA256</td><td>(RFC6668, 2012)</td><td>Integrity verification</td></tr>
<tr><td>Client authentication algorithm</td><td>public key-based (ecdsa-sha2-nistp256)
password-based</td><td>(RFC4252, 2006)</td><td>Client authentication</td></tr>
</table>
Packets greater than 32768 bytes in an SSH transport connection are dropped.
The TOE enforces default rekey thresholds of one hour and one gigabyte of transmitted data. After either of the thresholds are reached a rekey is performed. The threshold values may be changed by the administrator.
</td></tr>
<tr><td>FTP_ITC.1(1)</td><td>The TOE implements TLS with authentication as specified at FCS_TLSC_EXT.2 for connections with the Syslog server, update server and database server.</td></tr>
<tr><td>FTP_ITC.1(2)</td><td>The TOE implements TLS with mutual authentication as specified at FCS_TLSS_EXT.2 for connections with SIP clients.</td></tr>
<tr><td>FTP_ITC.1(3)</td><td>The TOE supports TLS connections with peer SIP servers and may do so as a client (FCS_TLSC_EXT.2) or server (FCS_TLSS_EXT.2).</td></tr>
<tr><td>FTP_TRP.1</td><td>TOE administrators use SSHv2 for remote administration as specified at FCS_SSHS_EXT.1.</td></tr>
</table>

6.3 Trusted Update

93 The TOE ensures the authenticity and integrity of software updates through digital signatures.

Table 17: Trusted Update SFRs

SFR	Fulfilment
FPT_TUD_EXT.1	Software updates are made available via an update server hosted within Secusmart cloud services. Each update is a tar file signed with a private Secusmart key dedicated for software package signing (ECDSA P-256) – the update package includes this digital signature. The SIP server has the corresponding public key in its filesystem, and only root can access it. With this public key the software update function of the SIP server can verify the signature using the OpenSSL module.

SFR	Fulfilment
	When the security administrator starts the software update process, the software update function: • copies the update package form the update server to the SIP server (via Stunnel using TLS) • checks the Secusmart signature • unpacks the tar file • starts the installation script included in the tar file Installation of the update fails if the digital signature verification fails. In this case an error message is displayed to the administrator, a log event is generated, the update is aborted and the original software remains unchanged.

6.4 System Monitoring

94 The TOE keeps local and remote audit records of security relevant events.

Table 18: System Monitoring SFRs

SFR	Fulfilment
FAU_GEN.1	The TOE uses the Linux Audit System (auditd) to log the events and information identified at FAU_GEN.1.
FAU_GEN.2	TOE audit events include the related user identity when applicable.
FAU_STG_EXT.1	The Linux auditd daemon process receives audit data from applications and the kernel. The daemon runs as a root process and writes audit data to an audit log file on the local machine. Only the root and the security administrator have read and write access to the locally saved audit log. Linux auditd can be configured by the security administrator to write audit logs additionally to the local rsyslog that can forward the logs to an external syslog server via TLS. Audit logs are saved in a dedicated disk partition. The default maximum size of the audit logs is 1 GB. The TOE will drop new audit data when the maximum log size is reached.
FPT_STM.1	Linux auditd makes use of time for timestamps in audit records. The TOE implements an internal clock provided by the OS to keep reliable time. Time can be set manually by the Administrator. Time is also used for certificate validation as described by FIA_X509_EXT.1 at section 6.7.

6.5 Secure Administration

95 The TOE enables secure local and remote management of its security functions.

Table 19: Secure Administration SFRs

SFR	Fulfilment
FIA_PMG_EXT.1	The TOE supports a defined character set for password-based authentication. Minimum password length is configurable by the TOE administrator and passwords must be 15 characters or greater.

FIA_UIA_EXT.1	The TOE requires entities to perform identification and authentication before performing any actions other than displaying a warning banner.
	The following login methods are supported:
	• **Local console.** Administrators may login locally with a correct username and password combination.
	• **SSH.** Administrators may login via SSH with either:
	  ○ Correct username and password combination, or
	  ○ Recognized ECDSA certificate (per FCS_SSHS specifications at section 6.2)

FIA_UAU_EXT.2	The TOE implements password-based authentication for administrators.
FIA_UAU.7	The TOE obscures feedback during password-based authentication.
FMT_MOF.1(1)/ TrustedUpdate	Access to management functions is restricted to TOE administrators, including restricting access to: • the ability to perform software updates • the ability to manage all configuration parameters • the ability to modify, delete, generate/import cryptographic keys and certificates Access Control is enforced by OS user and group management. No functions are available to users prior to authentication.
FMT_MOF.1(1)/AdminAct	
FMT_MTD.1	
FMT_MTD.1/AdminAct	
FMT_SMF.1	The TOE provides the management functions: • Ability to administer the TOE locally and remotely; • Ability to configure the access banner; • Ability to configure the session inactivity time before session termination or locking; • Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates; • Configure the SIP; • Configure mechanisms implemented with respect to FCS_TLSS_EXT.2; • Import X.509v3 certificates; • Configure SIP client password; • Ability to configure audit behavior;

SFR	Fulfilment
	• Ability to configure the cryptographic functionality; • Ability to configure thresholds for SSH rekeying.
FMT_SMR.2	The term 'Security Administrator' and 'TOE administrator' are used in this ST to refer to any user which is permitted to perform the identified management functions. The TOE supports local administration via direct console cable and remote administration via SSHv2. **Note:** TOE administrative users do not have root privilege,
FPT_SKP_EXT.1	The TOE stores all private keys in a secure directory that is not readily accessible to administrators. Refer to section 6.7.1 for detail on keys and CSPs.
FPT_APW_EXT.1	The TOE ensures that plaintext user passwords will not be disclosed even to administrators. Refer to section 6.7.1 for detail on stored passwords.
FTA_SSL_EXT.1	The TOE terminates console sessions after a configured period of inactivity.
FTA_SSL.3	The TOE terminates SSHv2 sessions after a configured period of inactivity.
FTA_SSL.4	TOE administrators are able to terminate an interactive session.
FTA_TAB.1	The TOE displays a configurable advisory notice and consent warning message regarding use of the TOE when connection via local console or SSHv2.

6.6 Self Test

96 The TOE performs a suite of self-tests to ensure the correct operation and enforcement of its security functions.

Table 20: Self Test SFRs

SFR	Fulfilment
FPT_TST_EXT.1	The TOE implements the Advanced Intrusion Detection Environment (AIDE) file and directory integrity checker to confirm the integrity of critical files and directories on start-up. AIDE is configured to perform the following: • construct a database of TSF critical files • uses OpenSSL to create a SHA-256 hash of each protected file • perform an integrity check of protected files at start-up (utilizing OpenSSL to generate hashes for comparison to database)

SFR	Fulfilment
	• if the integrity test fails the TOE will generate an audit event The TOE incorporates the OpenSSL FIPS Object Module as specified in section 6.7 which runs start-up self tests to confirm the correct operation of the cryptographic functions of the TOE. OpenSSL performs the following power on self-tests: • Software integrity – HMAC-SHA1 • HMAC Known Answer Tests (KAT) • AES KATs • TDES KATs • RSA KAT • DSA KAT • DRBG KATs • ECDSA pairwise consistency test • ECC CDH KAT Together, these tests ensure that the TSF is operating correctly.

6.7 Cryptographic Module

97 The TOE makes use of the OpenSSL FIPS Object Module Version 2.0.12.

98 The following CAVP certificates are relevant to the TOE: AES (#4381), CVL (#1077, #1078), DRBG (#1407), ECDSA (#1045), HMAC (#2909), RSA (#2367), and SHA (#3609).

Table 21: Cryptographic Module SFRs

SFR	Fulfilment
FCS_CKM.1	The TOE OpenSSL module performs key generation in accordance with the RSA (2048-bit) and ECDSA (P-256, P-384) schemes. These schemes are used in support of TLS and SSH (ECDSA only). The TOE acts as both sender and receiver (i.e. depending on the channel) when using RSA schemes in TLS.
FCS_CKM.2	The TOE OpenSSL module performs key establishment in accordance with ECC / NIST SP 800-56A in support of TLS and SSH. The TOE OpenSSL module performs RSA key establishment in accordance with NIST SP 800-56B in support of TLS. The TOE acts as both sender and receiver (i.e. depending on the channel) when using RSA schemes in TLS. In accordance with NIST SP 800-56B, the TOE does not reveal specific error details but raises generic errors during TLS handshake.
FCS_CKM.4	None of the TOE's symmetric keys, pre-shared keys, or private keys are stored in plaintext form.

<table>
<tr><th>SFR</th><th>Fulfilment</th></tr>
<tr><td></td><td>See section 6.7.1 for details of CSPs and related zeroization.</td></tr>
<tr><td>FCS_COP.1(1)</td><td>The TOE OpenSSL module performs AES encryption and decryption in CBC and GCM mode.</td></tr>
<tr><td>FCS_COP.1(2)</td><td>The TOE OpenSSL module performs RSA and ECDSA cryptographic signature services (generation and verification).</td></tr>
<tr><td>FCS_COP.1(3)</td><td>The TOE OpenSSL module performs SHA-1, SHA-256, SHA-384 and SAH-512 cryptographic hashing in support of the following functions:
SHA-1
TLS client authentication with RSA
Certificates: subjectKeyIdentifier
SHA-256
TLS server authentication with ECDHE-RSA
TLS pseudorandom function (PRF) with AES128-GCM
TLS PRF with AES128-CBC
SSH server and client authentication with ecdsa-sha2
SSH key exchange with ecdh-sha2
Digest Access Authentication
AIDE file integrity
SHA-384
Certificate signature
TLS PRF with AES256-GCM
SHA-512
Admin password hashing
SSH password hashing</td></tr>
<tr><td>FCS_COP.1(4)</td><td>The TOE OpenSSL module performs HMAC-SHA-1, HMAC-SHA-256 and HMAC-SHA-384 keyed-hash message authentication in support of the following functions:
TLS (all)
SSH (HMAC-SHA-256)
Details are as show in the table below.
<table>
<tr><th>HMAC</th><th>Key Length</th><th>Block Size</th><th>Output Length</th></tr>
<tr><td>SHA-1</td><td>160 bit</td><td>512 bit</td><td>160 bit</td></tr>
<tr><td>SHA-256</td><td>256 bit</td><td>512 bit</td><td>256 bit</td></tr>
<tr><td>SHA-384</td><td>384 bit</td><td>1024 bit</td><td>384 bit</td></tr>
</table>
</td></tr>
</table>

SFR	Fulfilment
FCS_RBG_EXT.1	The TOE leverages Intel's RdRand TRNG to seed the OpenSSL CTR_DRBG (AES). Additional details included in separate Entropy Assessment Report.

FIA_X509_EXT.1	The TOE performs certificate path validation in accordance with section 6.1 of RFC5280 – implementing the defined validation algorithm. The following steps are performed for each certificate in the path, starting from the trust anchor summarized as follows:
	• The public key algorithm and parameters are checked;
	• The current date/time is checked against the validity period of the certificate;
	• The revocation status is checked, by CRL, to ensure the certificate is not revoked. The revocation status of the peer certificate is checked each time a connection is made for the following communication flows:
	  ○ TLS connections with SIP clients (TOE is server)
	  ○ TLS connections with other components – database server, update server, syslog (TOE is client)
	  ○ TLS connections with peer SIP servers (TOE may be client and/or server)
	• The issuer name is checked to ensure that it equals the subject name of the previous certificate in the path;
	• The reference identifier is compared to the provided identifier
	• Name constraints are checked, to make sure the subject name is within the permitted subtrees list of all previous CA certificates and not within the excluded subtrees list of any previous CA certificate;
	• The asserted Certificate Policy OIDs are checked against the permissible OIDs as of the previous certificate, including any policy mapping equivalencies asserted by the previous certificate;
	• Policy constraints and basic constraints are checked, to ensure that any explicit policy requirements are not violated and that the certificate is a CA certificate, respectively.
	• The path length is checked to ensure that it does not exceed any maximum path length asserted in this or a previous certificate;
	• The key usage extension is checked; and
	• Any other critical extensions are recognized and processed.
	Path validation must start from a trusted root certificate (trust anchor).

FIA_X509_EXT.2	The TOE maintains a reference to the keystore location for each certificate and is thereby able to select the correct certificate for a given usage.

SFR	Fulfilment
	When the TOE cannot establish a connection to determine the validity of a certificate the TOE accepts the certificate. This behaviour applies to the trusted channels identified by FTP_ITC.1(1), FTP_ITC.1(2) and FTP_ITC.1(3).
FIA_X509_EXT.3	The TOE OpenSSL module is able to generate certificate signing requests and validate the CA response.

6.7.1 Cryptographic Keys and Passwords

6.7.1.1 Keys

99 Table 22 describes the keys and CSPs utilized by the TOE.

Table 22: Cryptographic Keys

Name	Use	Generation	Storage & Destruction
SIP TLS key (OpenSIPS)	TLS ECDSA ECC static private key Or TLS RSA static private key for RSASSA-PKCS1-v1_5	OpenSSL module	RAM: OpenSSL destroys (overwrite with zeros with read-after-write verify) private key when application calls function to clear the ssl context. OpenSIPS clears ssl context when it exits. File: Saved in /etc/pki/private directory with root access only. Zeroized by 3 times overwrite with a random pattern and final overwrite of zeroes with read-after-write verify.
Audit client TLS key Database server client TLS key SIP Server client TLS key	TLS RSA static private key for RSASSA-PKCS1-v1_5	OpenSSL module	RAM: OpenSSL destroys (overwrite with zeros with read-after-write verify) private key when application calls function to clear the ssl context at end of session. File: Saved in /etc/pki/private directory with root access only. Zeroized by 3 times overwrite with a random pattern and final overwrite of zeroes with read-after-write verify.

Name	Use	Generation	Storage & Destruction
TLS ECDSA random ephemeral secret	TLS	OpenSSL module	Stored in RAM. Destroyed by overwrite with zeroes with read-after-write verify when session is closed.
TLS ECC ephemeral private key for key exchange (ECDHE)			
TLS shared ECDHE secret			
TLS premaster secret			
TLS master secret			
TLS record layer AES and MAC keys			
PWD store key	AES key to encrypt database server password	OpenSSL module	File: Saved in /etc/pki/private directory with root access only. Zeroized by 3 times overwrite with a random pattern and final overwrite of zeroes with read-after-write verify. RAM: OpenSSL module destroys (overwrite with zeros with read-after-write verify) key after use.
CSPRNG – Seed	Seed OpenSSL RNG.	TOE application reads seed from /dev/random into RAM and passes this to OpenSSL module.	TOE application destroys (overwrite with zeroes with read-after-write verify) seed in RAM after this.
SSH ECDSA Private Key / server host key	ECDSA Suite B P-256 Server authentication	OpenSSL module	File: Saved in directory with root access only. Zeroized by 3 times overwrite with a random pattern and final overwrite of zeroes with read-after-write verify. RAM: OpenSSL module destroys (overwrite with

Name	Use	Generation	Storage & Destruction
			zeroes with read-after-write verify) key after use.
SSH EC Diffie-Hellman private key	EC DH P-256 Generation of session keys for encryption	OpenSSL module	RAM: OpenSSL module destroys (overwrite with zeroes with read-after-write verify) when the session is closed
SSH session keys	AES 128/256 Encryption		
SSH session HMAC key	HMAC-SHA256 Integrity		

6.7.1.2 Passwords

100 Table 22 describes passwords utilized by the TOE.

Table 23: Cryptographic Keys

Name	Use	Generation	Storage
SIP Password	Client authentication to the SIP Server. Stored as H(A1) – SHA-256 hash of (username-value ":" realm-value ":" password)	Externally generated by SCA at registration or manually input by SIP server administrator (i.e. change password).	Not stored in clear text. Persistent: H(A1) stored externally in database server. RAM: H(A1) Temporarily in RAM during client authentication.
Admin Passwords SSH passwords	System login. Managed by OS user password management.	Administrator input	File: SHA-512 hash. RAM: User inputted passwords are temporarily in RAM during authentication.
Database server password	Access database server.	Administrator input	File: Encrypted with AES256-CBC (see PWD store key). File accessible only to the processes that require access. RAM: Temporarily in RAM when used by calling applications.

7 Rationale

7.1 Conformance Claim Rationale

101 The following rationale is presented with regard to the PP conformance claims:

a) **TOE type.** As identified in section 2.1, the TOE is SIP server, consistent with the NDcPP and SIP EP.

b) **Security problem definition.** As shown in section 3, the threats, OSPs and assumptions are incorporated into this ST by reference to the NDcPP and SIP EP.

c) **Security objectives.** As shown in section 4, the security objectives are incorporated into this ST by reference to the NDcPP and SIP EP.

d) **Security requirements.** As shown in section 5, the security requirements are reproduced from the NDcPP and SIP EP. No additional requirements have been specified.

7.2 Security Objectives Rationale

102 All security objectives are drawn directly from the NDcPP and SIP EP.

7.3 Security Requirements Rationale

103 All security requirements are drawn directly from the NDcPP and SIP EP.

7.4 TOE Summary Specification Rationale

104 Table 24 provides a coverage mapping showing that all SFRs are mapped to the security functions described in the TSS.

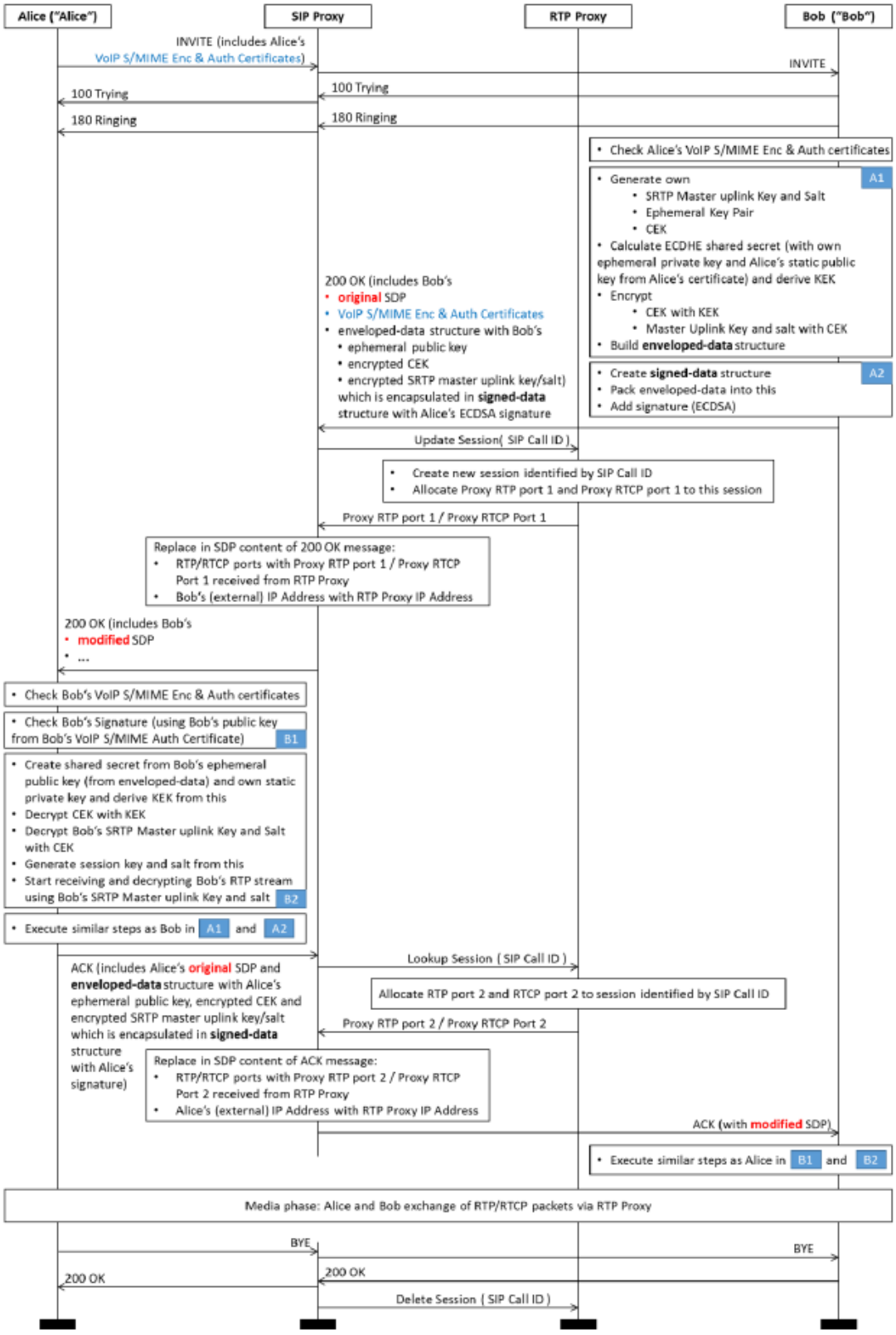
Table 24: Map of SFRs to TSS Security Functions

	SIP Server	Protected Communications	Trusted Update	System Monitoring	Secure Administration	Cryptographic Module
FAU_GEN.1				X		
FAU_GEN.2				X		
FAU_STG_EXT.1				X		
FCS_CKM.1						X
FCS_CKM.2						X

FCS_CKM.4						X
FCS_COP.1(1)						X
FCS_COP.1(2)						X
FCS_COP.1(3)						X
FCS_COP.1(4)						X
FCS_SSHS_EXT.1		X				
FCS_TLSC_EXT.2		X				
FCS_TLSS_EXT.2		X				
FCS_RBG_EXT.1		X				
FIA_PMG_EXT.1					X	
FIA_SIPS_EXT.1	X					
FIA_UIA_EXT.1					X	
FIA_UAU_EXT.2					X	
FIA_UAU.7					X	
FIA_X509_EXT.1						X
FIA_X509_EXT.2						X
FIA_X509_EXT.3						X
FMT_MOF.1(1)/Trusted Update					X	
FMT_MOF.1(1)/AdminAct					X	
FMT_MTD.1					X	
FMT_MTD.1/AdminAct					X	
FMT_SMF.1					X	
FMT_SMR.2					X	
FPT_SKP_EXT.1					X	
FPT_APW_EXT.1					X	
FPT_TST_EXT.1					X	

FPT_TUD_EXT.1			X			
FPT_STM.1				X		
FTA_SSL_EXT.1					X	
FTA_SSL.3					X	
FTA_SSL.4					X	
FTA_TAB.1					X	
FTP_ITC.1(1)		X				
FTP_ITC.1(2)		X				
FTP_ITC.1(3)		X				
FTP_TRP.1		X				

Annex A: Call Signaling



Annex B: SDP Example

The following is an example of SDP content from SIP message during call setup:

```
v=0
o=- 3650186066 3650186066 IN IP4 10.137.89.193
s=pjmedia
b=AS:54
t=0 0
a=X-nat:0
m=audio 4000 RTP/SAVP 102 100 105 96
c=IN IP4 10.137.89.193
b=TIAS:36000
a=rtcp:4001 IN IP4 10.137.89.193
a=sendrecv
a=rtpmap:102 SILK/16000
a=fmtp:102 useinbandfec=0
a=rtpmap:100 SILK/8000
a=fmtp:100 useinbandfec=0
a=rtpmap:105 AMR/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-16
```

- v=0 > SDP version number
- o=- 3650186066 3650186066 IN IP4 10.137.89.193 > originator and session identifier:
  - username = - > originating host does not support the concept of user ID
  - sess-id = 3650186066
  - sess-version = 3650186066
  - nettype = IN ~ Internet
  - addrtype = IPv4
  - unicast-address = 10.137.89.193 (address of machine from which the session was created)
- s=pjmedia > session name: PJMEDIA stack enters its name here
- b=AS:54 > bandwidth (including transport):
  - bwtype = AS ~ application specific (~ RTP session bandwidth ~ aggregate limit down to IP layer, might be reserved and enforced by the network)
  - bandwidth = 54 kbps
- a=X-nat:0 > NAT-type is unknown. Client using STUN can detect and communicate the NAT type with the X-nat attribute (0:unknown, 1: full cone, ... , 6: symmetric)
- m=audio 4000 RTP/SAVP 102 100 105 96 > media description:
  - media = audio
  - transport port = 4000
  - protocol = RTP/SAVP ~Secure Real-time Transport Protocol running over UDP
  - format = 102, 100, 105 and 96 (~ RTP payload types as specified below with rtpmap attributes, 102 is default
- c=IN IP4 10.137.89.193 > Connection:

  - nettype = IN ~ Internet
  - end point IP address = 10.137.89.193
- b=TIAS:36000 > bandwidth (excluding transport, RFC-3890t): TIAS ~ Transport Independent Application Specific maximum ~ Maximum media codec rate = 36000 bit (i.e. IP/UDP/RTP overhead not considered)
- a=rtcp:4001 IN IP4 10.137.89.193 > rtcp attribute (see (RFC3605, 2003)):
  - port = 4001
  - nettype = IN ~ Internet
  - addrtype = IPv4
  - connection address = 10.137.89.193
- a=sendrecv > sendrecv: can transmit and receive media data
- a=rtpmap:102 SILK/16000 > Codec for dynamic RTP payload type 102 is SILK with sampling rate 16000 Hz
- a=fmtp:102 useinbandfec=0 > Format parameter for codec 102: no inband FEC
- a=rtpmap:100 SILK/8000 > Codec for dynamic RTP payload type 100 is SILK with sampling rate 8000 Hz
- a=fmtp:100 useinbandfec=0 > Format parameter for codec 100 : no inband FEC
- a=rtpmap:105 AMR/8000 > Codec for dynamic RTP payload type 105 is AMR with sampling rate 8000 Hz
- a=rtpmap:96 telephone-event/8000 > Codec for dynamic RTP payload type 96 is telephone-event
- a=fmtp:96 0-16 > Format parameter for codec 96: supported telephone events
- A sending gateway can recognize tones such as ringing or busy tone or DTMF digit '0', and transmit a code that identifies them using the telephone-event payload
- DTMF-related named events within the telephone-event payload format (see (RFC4733, 2006):

DTMF Event	encoding (decimal)
0-9	0-9
*	10
#	11
A-D	12-15
Flash	16

- Note: When the SIP server changes SDP connection information (IP addresses and ports) so that it points to an RTP proxy server it will add the SDP attribute a=nortpproxy:yes. This marks that the SDP connection information in the SIP message has already been overwritten.
- See (RFC4566, 2006)