Secusmart

SecuSUITE Client v3.0 and Vodafone Secure Call Client v3.0

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	21 Mar 2016	L Turner	Release for evaluation.
1.1	18 Apr 2016	L Turner	Updated to address evaluator observations.
1.2	20 Jun 2016	L Turner	Updated to address certifier observations.
1.3	11 Aug 2016	L Turner	Address certification comments.
1.4	7 Dec 2016	L Turner	Additional clarifications.
1.5	6 Feb 2017	L Turner	Final for certification.
1.6	10 Feb 2017	L Turner	Address certifier observations.
1.7	01 Mar 2017	L Turner	Removed Blackberry Q5 device.
1.8	02 Mar 2017	L Turner	Added HMAC Integrity Key
1.9	07 Apr 2017	Secusmart	Address certifier observations (ETR Review)
1.10	01 May 2017	L Turner	Address additional TDs.

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 Client v3.0 and Vodafone Secure Call Client v3.0) and supporting backend infrastructure (including the SecuSUITE SIP Server).

3 This Security Target (ST) defines the SecuSUITE Client v3.0 and Vodafone Secure Call Client v3.0 Target of Evaluation (TOE) for the purposes of Common Criteria (CC) evaluation.

4 The TOE is the client application component of the SecuSUITE security solution. The client application is available as two differently branded mobile apps (SecuSUITE Client v3.0 and Vodafone Secure Call Client v3.0) with the only difference being name and branding graphics.

1.2 Identification

Table 1: Evaluation identifiers

Target of Evaluation	Secusmart SecuSUITE Client v3.0 and Vodafone Secure Call Client v3.0 Build: 3.0.17
Security Target	Secusmart SecuSUITE Client v3.0 and Vodafone Secure Call Client v3.0 Security Target, v1.10, 1 May 2017

1.3 Conformance Claims

5 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) Protection Profile for VOIP Applications, v1.3 with NIAP Technical Decisions:

i) TD0042: Removal of Low-level Crypto Failure Audit from PPs

ii) TD0068: Addition of SRTP Ciphersuites

iii) TD0079: RBG Cryptographic Transitions per NIST SP 800-131A Revision 1

iv) TD0088: Revision to FDP_VOP_EXT.1.1 in VoIP PP v1.3

v) TD0106: Removing SDES/SRTP from FIA_X509_EXT.2

vi) TD0107: FCS_CKM - ANSI X9.31-1998, Section 4.1.for Cryptographic Key Generation

vii) TD0161: FTP_ITC.1(2) - Test 2 Not Required

viii) TD0163: Update to FCS_TLSC_EXT.1.1 Test 5.4 and FCS_TLSS_EXT.1.1 Test

1.4 Terminology

Table 2: Terminology

Term	Definition
CC	Common Criteria
EAL	Evaluation Assurance Level
H(A1)	SHA-256 hash of (username-value ":" realm-value ":" SIP password)
OSP	Organizational Security Policy
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.
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.
TOE	Target of Evaluation
TSF	TOE Security Functionality
URI	Uniform Resource Identifier
SBC	Session Border Controller – logical component made up of the SIP Server and RTP proxy.
SDES	Security Descriptions for Media Streams
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

Term	Definition
	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).
VoIP PP	Protection Profile for VOIP Applications, v1.3

2 TOE Description

2.1 Type

6 The TOE, herein referred to as the SecuSUITE Client or the TOE, is a VoIP application that executes on a mobile device operating system.

2.2 Usage

7 The TOE allows users to place secure VoIP calls over data connections like EDGE, UMTS/HSPA, LTE or WiFi on mobile devices running the SecuSUITE App.

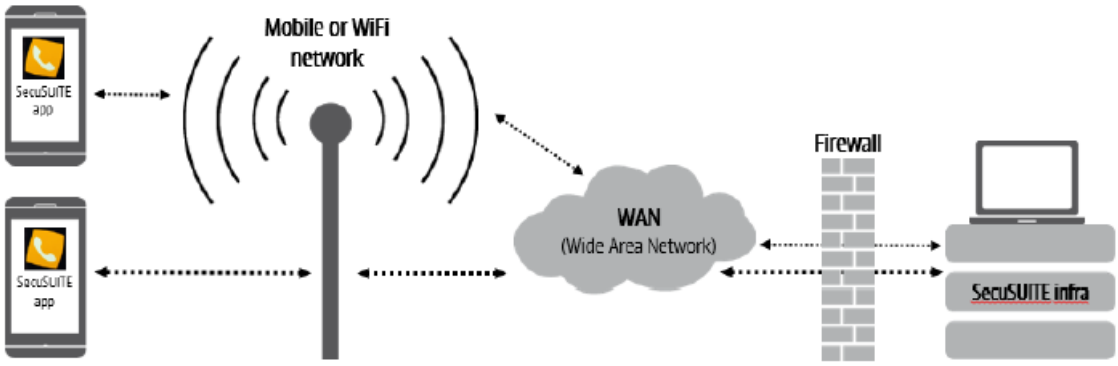


Figure 1: TOE Usage Scenario

2.2.1 User Context

8 The TOE user downloads the SecuSUITE Client from an app store (e.g. Apple Store, Google Play) or it is pushed via a Mobile Device Management (MDM) server (e.g. BlackBerry Enterprise Server) and installs the app to their mobile device. On first use of the app, the user must go through a registration process in order to register to a specified SecuSUITE infrastructure (identified by URI).

9 Once registered, the user can place secure VoIP calls using the app with largely the same interactions as with a normal phone call. End-to-end encrypted calls are only possible with other SecuSUITE app users registered to the same infrastructure. If the contact isn't a SecuSUITE user, the connection is cancelled with the message "The number is not reachable via SecuSUITE."

10 Users are typically invited to join SecuSUITE via an activation email initiated by their corporate IT administrator who adds users via the SecuSUITE infrastructure administration portal.

2.2.2 SecuSUITE Context

11 The TOE is part of the SecuSUITE Security Solution shown in Figure 2 below. The TOE does not work in isolation but relies on SecuSUITE infrastructure components to enable a secure VoIP communications.

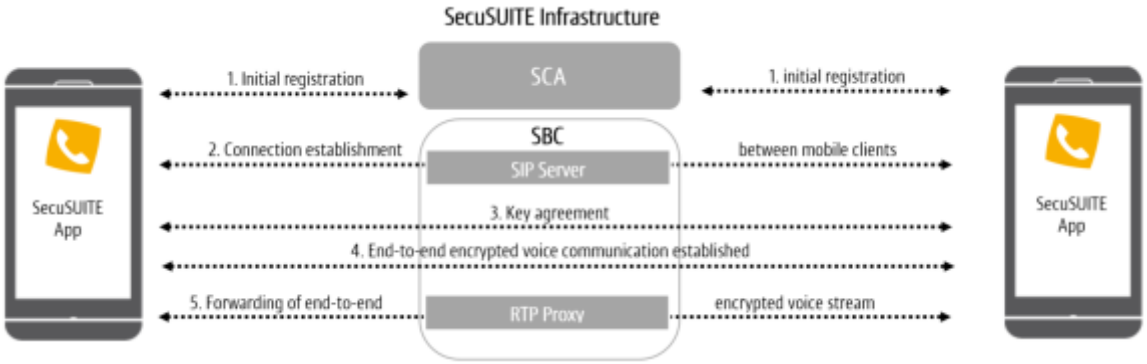


Figure 2: SecuSUITE Security Solution

12 As shown in Figure 2, 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. 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. The Secusmart SecuSUITE SIP Server Security Target defines the SIP Server TOE.

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.

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 SRTP session keys.

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.

2.2.3 VoIP Client

13 The SecuSUITE Client establishes a secure tunnel for voice communications with another SecuSUITE client. The tunnel provides confidentiality, integrity, and data authentication for information that travels across the public network. This occurs using the Secure Real-Time Transport Protocol (SRTP) that has been established using the Session Description Protocol (SDP) and the Security Descriptions for Media Streams (SDES) for SDP - the TOE supports SDES-SRTP.

14 The SecuSUITE Client also protects communications between itself and the SIP Server by using a Transport Layer Security (TLS)-protected signalling channel. To

register the TOE within the domain, the TOE is required to be password authenticated by the SIP Server. The TOE also makes use of certificates to authenticate both the SIP server end and the TOE itself through the TLS connection.

2.3 Security Functions

15 The TOE provides the following security functions:

a) **Secure Tunnels.** The TOE implements TLS and SDES-SRTP to secure communications with the SIP Server and remote VoIP applications.

b) **TOE Configuration.** The TOE provides interfaces to control the configuration of TLS and SDES-SRTP and the underlying cryptographic mechanisms, management of X.509 certificates, and updates to the TOE.

c) **Verifiable Updates.** The mobile device platform uses digital signatures to verify software updates to the TOE.

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

e) **Cryptographic Module.** The TOE includes a cryptographic module - OpenSSL FIPS Object Module Version 2.0.12.

2.4 Physical Scope

16 The TOE boundary is illustrated in Figure 3.

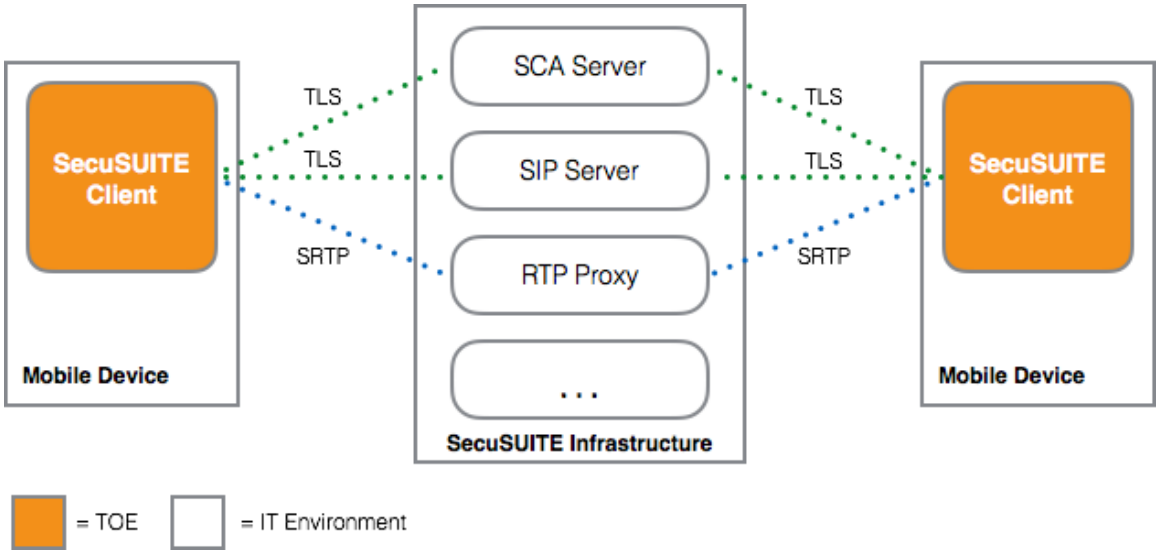


Figure 3: TOE Boundary

17 The TOE is comprised of the following software:

a) SecuSUITE Client v3.0 and Vodafone Secure Call Client v3.0

18 The TOE executes on the following mobile devices:

a) Blackberry Passport, Leap, Classic, Q10, Z30 and Z10 (Security Target: https://www.cse-cst.gc.ca/en/system/files/pdf_documents/blackberry-v1033-sec-eng.pdf)

b) Samsung Galaxy S7 and S7 Edge (Security Target: https://www.niap-ccevs.org/st/st_vid10726-st.pdf)

c) Apple iPhone 6 and 6 Plus (Apple iOS 9.3) (Security Target: https://www.niap-ccevs.org/st/st_vid10725-st.pdf)

2.4.1 Guidance Documents

19 The TOE includes the following guidance documents:

a) SecuSUITE App User Guide, v2.4

b) Vodafone Secure Call App User Manual, v2.2

c) SecuSUITE Client v3.0 and Vodafone Secure Call Client v3.0 Common Criteria Configuration Guide, v2.1

2.4.2 Non-TOE Components

20 The TOE is part of the 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 the database server.

b) **SecuSUITE Database Server v1.0.** Stores configuration data for use by SecuSUITE infrastructure components.

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.

d) **SecuSUITE SIP Server v1.0.** The SIP Server is used to establish the secure connection between the mobile devices. The use of a TLS connection, providing encryption and mutual authentication, ensures that the devices connect with authorized SIP servers only and the dialed call numbers are transmitted encrypted.

e) **SecuSUITE RTP Proxy v1.0.** The Real-time Transport Protocol (RTP) Proxy bridges media packets sent between clients. The RTP Proxy is part of the SecuSUITE SIP Server. The SIP Server creates and deletes RTP and Real-time Transport Control Protocol (RTCP) bridging sessions in the RTP Proxy.

2.5 Logical Scope

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

3 Security Problem Definition

22 The following security problem definition is reproduced from Annex A of the VoIP PP. No additional threats, assumptions or OSPs are added in this Security Target.

3.1 Threats

23 Table 3 identifies the threats addressed by the TOE.

Table 3: Threats

Identifier	Description
T.TSF_CONFIGURATION	Failure to allow configuration of the TSF may prevent its users from being able to adequately implement their particular security policy, leading to a compromise of user information.
T.TSF_FAILURE	Security mechanisms of the TOE may fail, leading to a compromise of the TSF.
T.UNAUTHORIZED_ACCESS	A user may gain unauthorized access to the TOE data. 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	Voice data may be inadvertently sent to a destination not intended because it is sent outside the voice call.

3.2 Organizational Security Policies

24 No OSPs are defined.

3.3 Assumptions

25 Table 4 identifies the assumptions related to the TOE's environment.

Table 4: Assumptions

Identifier	Description
A.AVAILABILITY	Network resources shall be available to allow VoIP clients to satisfy mission requirements and to transmit information.

Identifier	Description
A.OPER_ENV	The operational environment of the TOE appropriately addresses those requirements, threats, and policies not applicable to the TOE itself, but that are necessary to support the correct operation of the TOE.
A.TRUSTED_CONFIG	Personnel configuring the TOE and its operational environment will follow the applicable security configuration guidance.

4 Security Objectives

26 The security objectives are reproduced from Annex A of the VoIP PP. No additional objectives are added in this Security Target.

4.1 Objectives for the Operational Environment

27 Table 5 identifies the objectives for the operational environment.

Table 5: Operational environment objectives

Identifier	Description
OE.AUTHORIZED_USER	The user of the TOE is non-hostile and follows all user guidance.
OE.OPER_ENV	The operational environment will provide a SIP infrastructure to establish a VoIP connection; a PKI to provide certificates; and an execution domain to support correct operation of the TOE.
OE.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

4.2 Objectives for the TOE

28 Table 6 identifies the security objectives for the TOE.

Table 6: Security objectives

Identifier	Description
O.PROTECTED_COMMUNICATIONS	The TOE will provide protected communication channels with authorized IT entities (SIP Server and other VoIP applications).
O.TSF_SELF_TEST	The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly.

Identifier	Description
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.

5 Security Requirements

5.1 Conventions

29 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).

30 **Note:** This ST reproduces the SFRs, including applied conventions and identified operations, from the VoIP PP.

5.2 Extended Components Definition

31 Table 7 identifies the extended components that are incorporated into this ST.

Table 7: Extended Components

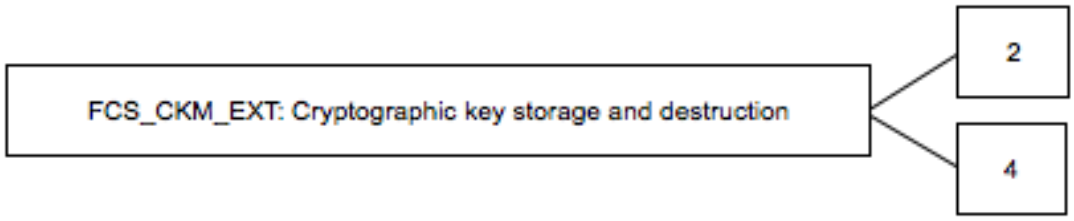
Component	Title	Rationale
FCS_CKM_EXT.2	Cryptographic Key Storage	Drawn from VoIP PP.
FCS_CKM_EXT.4	Cryptographic key material destruction (Key Material)	
FCS_RBG_EXT.1	Extended: Cryptographic operation (Random Bit Generation)	
FCS_SRTP_EXT.1	Secure Real-Time Transport Protocol (SRTP)	
FCS_TLS_EXT.1	Transport Level Security	
FDP_VOP_EXT.1	Voice Over IP Data Protection	
FIA_SIPC_EXT.1	Session Initiation Protocol (SIP) Client	
FIA_X509_EXT.1	Extended: X509 Certificate Validation	
FIA_X509_EXT.2	Extended: X509 Certificate Use and Management	
FPT_TST_EXT.1	Extended: TSF Self Test	
FPT_TUD_EXT.1	Extended: Trusted Update	

5.2.1 Cryptographic key storage and destruction (FCS_CKM_EXT)

5.2.1.1 Family Behavior

32 This family provides requirements that address cryptographic key storage and destruction.

5.2.1.2 Component Leveling



33 FCS_CKM_EXT.2 addresses storage of cryptographic keys.

34 FCS_CKM_EXT.4 addresses the destruction of key material. **Note:** The FCS_CKM family addresses components 1 and 3.

5.2.1.3 Management: FCS_CKM_EXT.2, FCS_CKM_EXT.4

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

a) None

5.2.1.4 Audit:

FCS_CKM_EXT.2

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

a) None

FCS_CKM_EXT.4

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

a) Minimal: Success and failure of the activity.

b) Basic: The object attribute(s), and object value(s) excluding any sensitive information (e.g. secret or private keys).

FCS_CKM_EXT.2 Cryptographic Key Storage

Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM_EXT.4 Cryptographic key material destruction
FCS_CKM_EXT.2.1	The VoIP client application shall store persistent secrets and private keys when not in use in platform-provided key storage.

FCS_CKM_EXT.4 Cryptographic key material destruction (Key Material)

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]

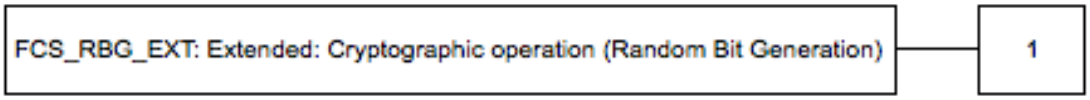
FCS_CKM_EXT.4.1 The [selection, choose at least one of: VoIP client application, client device platform] shall zeroize all plaintext secret and private cryptographic keys and Critical Security Parameters (CSPs) when no longer required.

5.2.2 Extended: Cryptographic operation (Random Bit Generation) (FCS_RBG_EXT)

5.2.2.1 Family Behavior

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

5.2.2.2 Component Leveling



39 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.

5.2.2.3 Management: FCS_RBG_EXT.1

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

a) None

5.2.2.4 Audit: FCS_RBG_EXT.1

41 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 [selection, choose at least one of: VoIP client application, client device platform] shall perform all deterministic random bit generation services in accordance with [selection, choose one of: NIST Special Publication 800-90A using [selection: Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES)].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by an entropy source that accumulates entropy from [selection: a software-based noise source, a platform-based RBG] with a minimum of [selection: 128 bits, 256 bits] of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.

5.2.3 Secure Real-Time Transport Protocol (SRTP) (FCS_SRTP_EXT)

5.2.3.1 Family Behavior

42 This family provides requirements that address the Secure Real-Time Transport Protocol (SRTP).

5.2.3.2 Component Leveling

43 FCS_SRTP_EXT.1 addresses SRTP requirements.

5.2.3.3 Management: FCS_SRTP_EXT.1

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

a) None

5.2.3.4 Audit: FCS_SRTP_EXT.1

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

a) None

FCS_SRTP_EXT.1 Secure Real-Time Transport Protocol (SRTP)

Hierarchical to: No other components.

Dependencies: None

FCS_SRTP_EXT.1.1 The VoIP client application shall implement the Secure Real-Time Transport Protocol (SRTP) that complies with RFC 3711, and use Security Descriptions for Media Streams (SDES) in compliance with RFC 4568 to provide key information for the SRTP connection.

FCS_SRTP_EXT.1.2 The VoIP client application shall implement SDES-SRTP supporting the following ciphersuites: AES_CM_128_HMAC_SHA1_80 in accordance with RFC 4568 and [selection: AES_256_CM_HMAC_SHA1_80 in accordance with RFC 6188, AEAD_AES_256_GCM in accordance with RFC 7714, no other].

FCS_SRTP_EXT.1.3 The VoIP client application shall ensure the SRTP NULL algorithm can be disabled.

FCS_SRTP_EXT.1.4 The VoIP client application shall allow the SRTP ports to be used for SRTP communications to be specified by an Authorized Administrator.

5.2.4 Transport Level Security (FCS_TLS_EXT)

5.2.4.1 Family Behavior

46 The component in this family addresses protecting data between a client and a server using the TLS protocol.

5.2.4.2 Component Leveling

47 FCS_TLS_EXT.1 addresses protecting data between a client and a server using the TLS protocol.

5.2.4.3 Management: FCS_TLS_EXT.1

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

a) None

5.2.4.4 Audit: FCS_TLS_EXT.1

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

a) None

FCS_TLS_EXT.1 Transport Level Security

Hierarchical to: No other components.

Dependencies: None

FCS_TLS_EXT.1.1 The [selection, choose at least one of: VoIP client application, client device platform] shall implement one or more of the following protocols [selection: TLS 1.0 (RFC 2246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)] using mutual authentication with certificates and supporting the following ciphersuites:

Mandatory Ciphersuites in accordance with RFC 3268:

- TLS_RSA_WITH_AES_128_CBC_SHA

Optional Ciphersuites: [selection:

- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246

- TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH _AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 6460
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 6460
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- no other ciphersuite]

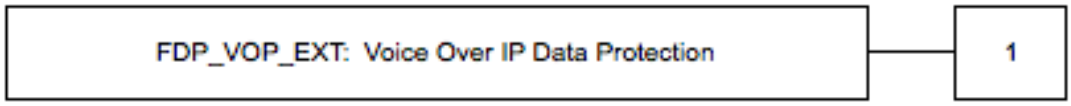
FCS_TLS_EXT.1.2 The [selection, choose at least one of: VoIP client application, client device platform] shall not establish a trusted channel if the distinguished name (DN) contained in a certificate does not match the expected DN for the peer.

5.2.5 Voice Over IP Data Protection (FDP_VOP_EXT)

5.2.5.1 Family Behavior

50 This family provides requirements that address Voice Over IP Data Protection.

5.2.5.2 Component Leveling



51 FDP_VOP_EXT.1 provides requirements that address Voice Over IP Data Protection.

5.2.5.3 Management: FDP_VOP_EXT.1

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

a) None

5.2.5.4 Audit: FDP_VOP_EXT.1

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

a) None

FDP_VOP_EXT.1 Voice Over IP Data Protection

Hierarchical to: No other components.

Dependencies: None

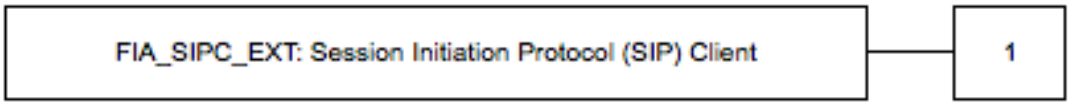
FDP_VOP_EXT.1.1 The VoIP Client Application shall stop the transmission of voice data when a VoIP call is placed on mute, a VoIP call is not connected, [selection: a VoIP call is placed on hold, no other selections] and [assignment: other actions, no other actions].

5.2.6 Session Initiation Protocol (SIP) Client (FIA_SIPC_EXT)

5.2.6.1 Family Behavior

54 This family provides requirements that address Session Initiation Protocol (SIP) clients.

5.2.6.2 Component Leveling



55 FIA_SIPC_EXT.1 provides requirements for SIP clients.

5.2.6.3 Management: FIA_SIPC_EXT.1

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

a) None

5.2.6.4 Audit: FIA_SIPC_EXT.1

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

a) None

FIA_SIPC_EXT.1 Session Initiation Protocol (SIP) Client

Hierarchical to: No other components.

Dependencies: None

FIA_SIPC_EXT.1.1 The VoIP client application 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_SIPC_EXT.1.2 The VoIP client application shall require the user to enter a password to support the use of password authentication for SIP REGISTER function requests as specified in section 22 of RFC 3261.

FIA_SIPC_EXT.1.3 The VoIP client application 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]}.

FIA_SIPC_EXT.1.4 The password entered by the user as per FIA_SIPC_EXT.1.2 shall be cleared by the VoIP client application once the VoIP client application is notified that the REGISTER request was successful.

5.2.7 Extended: X509 Certificates (FIA_X509_EXT)

5.2.7.1 Family Behavior

58 This family provides requirements that address X.509 certificates.

5.2.7.2 Component Leveling

59 FIA_X509_EXT.1 addresses X.509 certificate validation.

60 FIA_X509_EXT.2 addresses X.509 certificate use and management.

5.2.7.3 Management: FIA_X509_EXT.1, FIA_X509_EXT.2

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

a) None

5.2.7.4 Audit: FIA_X509_EXT.1, FIA_X509_EXT.2

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

a) None

FIA_X509_EXT.1 Extended: X509 Certificate Validation

Hierarchical to: No other components.

Dependencies: None

FIA_X509_EXT.1.1 The [selection, choose at least one of: VoIP client application, client device platform] shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- Validate the certificate path by ensuring the basicConstraints extension is present and the CA flag is set to TRUE for all CA certificates.
- 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 5759].
- 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).
    - 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.

FIA_X509_EXT.1.2 The [selection, choose at least one of: VoIP client application, client device platform] 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 Extended: X509 Certificate Use and Management

FIA_X509_EXT.2.1 The [selection, choose at least one of: VoIP client application, client device platform] shall use X.509v3 certificates as defined by RFC 5280 to support authentication for TLS, and [selection: code signing for software updates, code signing for software integrity verification, no additional uses].

FIA_X509_EXT.2.2 When the [selection, choose at least one of: VoIP client application, client device platform] cannot establish a connection to determine the validity of a certificate, the [selection, choose at least one of: VoIP client Application, client device platform] shall [selection: allow the administrator to choose whether to establish or not establish the trusted channel in these cases, accept the certificate, not accept the certificate].

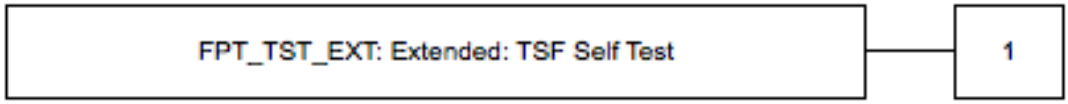
FIA_X509_EXT.2.3 The [selection, choose at least one of: VoIP client Application, client device platform] shall not establish a trusted communication channel if the peer certificate is deemed invalid.

5.2.8 Extended: TSF Self Test (FPT_TST_EXT)

5.2.8.1 Family Behavior

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

5.2.8.2 Component Leveling



64 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.

5.2.8.3 Management: FPT_TST_EXT.1

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

a) None

5.2.8.4 Audit: FPT_TST_EXT.1

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

a) None

FPT_TST_EXT.1 Extended: TSF Self Test

Hierarchical to: No other components.

Dependencies: None

FPT_TST_EXT.1.1 The [selection, choose at least one of: VoIP Client Application, client device platform] shall run a suite of self tests during initial start-up (on power on) to demonstrate correct operation of the TSF.

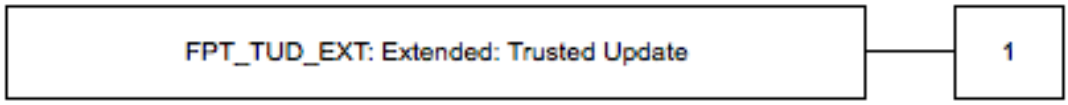
FPT_TST_EXT.1.2 The [selection, choose at least one of: VoIP Client Application, client device platform] shall provide the capability to verify the integrity of stored TSF executable code when it is loaded for execution through the use of the TSF-provided cryptographic services.

5.2.9 Extended: Trusted Update (FPT_TUD_EXT)

5.2.9.1 Family Behavior

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

5.2.9.2 Component Leveling



68 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.

5.2.9.3 Management: FPT_TUD_EXT.1

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

a) None

5.2.9.4 Audit: FPT_TUD_EXT.1

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

a) None

FPT_TUD_EXT.1 Extended: Trusted Update

Hierarchical to: No other components.

Dependencies: None

FPT_TUD_EXT.1.1 The TSF shall provide the client device platform the ability to query the current version of the TOE firmware/software.

FPT_TUD_EXT.1.2 The [selection, choose at least one of: VoIP client application, client device platform] shall provide authorized administrators the ability to initiate updates to the TOE firmware/software.

FPT_TUD_EXT.1.3 The [selection, choose at least one of: VoIP client application, client device platform] shall provide a means to verify firmware/software updates to the TOE using a digital signature mechanism and [selection: published hash, no other functions] prior to installing those updates.

5.3 Functional Requirements

Table 8: Summary of SFRs

Requirement	Title
FCS_CKM.1(1)	Cryptographic Key Generation (Asymmetric Keys)
FCS_CKM.1(2)	Cryptographic Key Generation
FCS_CKM_EXT.2(1)	Cryptographic Key Storage
FCS_CKM_EXT.4(1)	Cryptographic key material destruction (Key Material)
FCS_CKM_EXT.4(2)	Cryptographic key material destruction (Key Material)
FCS_COP.1(1)	Cryptographic Operation (Data Encryption/Decryption)
FCS_COP.1(2)	Cryptographic Operation (for cryptographic signature)
FCS_COP.1(3)	Cryptographic Operation (for cryptographic hashing)
FCS_COP.1(4)	Cryptographic Operation (For keyed-hash Message Authentication

Requirement	Title
FCS_RBG_EXT.1	Extended: Cryptographic operation (Random Bit Generation)
FCS_SRTP_EXT.1	Secure Real-Time Transport Protocol (SRTP)
FCS_TLS_EXT.1	Transport Level Security
FDP_VOP_EXT.1	Voice Over IP Data Protection
FIA_SIPC_EXT.1	Session Initiation Protocol (SIP) Client
FIA_X509_EXT.1	Extended: X509 Certificate Validation
FIA_X509_EXT.2	Extended: X509 Certificate Use and Management
FMT_SMF.1(1)	Specification of Management Functions
FMT_SMF.1(2)	Specification of Management Functions
FPT_TST_EXT.1	Extended: TSF Self Test
FPT_TUD_EXT.1	Extended: Trusted Update
FTP_ITC.1(1)	Inter-TSF Trusted Channel (SDES-SRTP)
FTP_ITC.1(2)	Inter-TSF Trusted Channel (TLS/SIP)
FTP_ITC.1(3)	Inter-TSF Trusted Channel (Refined)

5.3.1 Cryptographic Support (FCS)

FCS_CKM.1(1) Cryptographic Key Generation (Asymmetric Keys)

FCS_CKM.1.1(1) **Refinement:** The VoIP client application and client device platform shall generate **asymmetric** cryptographic keys **used for key establishment** in accordance with:

- NIST Special Publication 800-56B, “Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography” for RSA-based key establishment schemes and
- NIST Special Publication 800-56A, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” for elliptic curve-based key establishment schemes and implementing “NIST curves” P-256, P-384 and no other curves (as defined in FIPS PUB 186-4, “Digital Signature Standard”);

and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits.

FCS_CKM.1(2) Cryptographic Key Generation

FCS_CKM.1.1(2) **Refinement:** The VoIP client application shall generate **asymmetric** cryptographic keys **used for authentication** in accordance with a specified cryptographic key generation algorithm:

- FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 for RSA schemes;
- FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4 for ECDSA schemes and implementing "NIST curves" P-256, P-384 and no other curves;

and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits.

FCS_CKM_EXT.2(1) Cryptographic Key Storage

FCS_CKM_EXT.2.1(1) The VoIP client application shall store persistent secrets and private keys when not in use in platform-provided key storage.

FCS_CKM_EXT.4(1) Cryptographic key material destruction (Key Material)

FCS_CKM_EXT.4.1(1) **Refinement:** The VoIP client application shall zeroize all plaintext secret and private cryptographic keys and Critical Security Parameters (CSPs) when no longer required.

FCS_CKM_EXT.4(2) Cryptographic key material destruction (Key Material)

FCS_CKM_EXT.4.1(2) **Refinement:** The client device platform shall zeroize all plaintext secret and private cryptographic keys and Critical Security Parameters (CSPs) when no longer required.

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

FCS_COP.1.1 **Refinement:** The VoIP client application shall perform *encryption and decryption* in accordance with a specified cryptographic algorithm *AES operating in* **CTR, CBC**, *and GCM (as defined in NIST SP800-38D)* and cryptographic key sizes 128-bits, and 256-bits that meets the following:

- FIPS PUB 197, "Advanced Encryption Standard (AES)"
- NIST SP 800-38A, NIST SP 800-38D

FCS_COP.1(2) Cryptographic Operation (for cryptographic signature)

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

- ***FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 for RSA schemes***
- ***FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4 for Elliptic Curve Digital Signature Algorithm (ECDSA) schemes and implementing "NIST curves" P-256, P-384, and no other curves***

and cryptographic key sizes [***equivalent to, or greater than, a symmetric key strength of 112 bits***].

FCS_COP.1(3) Cryptographic Operation (for cryptographic hashing)

FCS_COP.1.1(3) **Refinement:** The VoIP client application shall perform **cryptographic hashing** in accordance with a specified cryptographic algorithm SHA-1 and SHA-256, SHA-384 and **message digest sizes** 160 bits and 256, 384 bits that meet the following: *FIPS PUB 180-3, "Secure Hash Standard."*

FCS_COP.1(4) Cryptographic Operation (For keyed-hash Message Authentication)

FCS_COP.1.1(4) **Refinement:** The VoIP client application shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm *HMAC-SHA-1 and HMAC-SHA-256* and cryptographic key sizes *160, 256 bits*, **and message digest sizes 160 and 256 bits** that meet the following: *FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code, and FIPS Pub 180-3, "Secure Hash Standard."*

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

FCS_RBG_EXT.1.1 The VoIP client application shall perform all deterministic random bit generation services in accordance with *NIST Special Publication 800-90A using CTR_DRBG (AES)*.

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by an entropy source that accumulates entropy from *a platform-based RBG* with a minimum of *256 bits* of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.

FCS_SRTP_EXT.1 Secure Real-Time Transport Protocol (SRTP)

FCS_SRTP_EXT.1.1 The VoIP client application shall implement the Secure Real-Time Transport Protocol (SRTP) that complies with RFC 3711, and use Security Descriptions for Media Streams (SDES) in compliance with RFC 4568 to provide key information for the SRTP connection.

FCS_SRTP_EXT.1.2 The VoIP client application shall implement SDES-SRTP supporting the following ciphersuites: AES_CM_128_HMAC_SHA1_80 in accordance with RFC 4568 and no other.

FCS_SRTP_EXT.1.3 The VoIP client application shall ensure the SRTP NULL algorithm can be disabled.

FCS_SRTP_EXT.1.4 The VoIP client application shall allow the SRTP ports to be used for SRTP communications to be specified by an Authorized Administrator.

FCS_TLS_EXT.1 Transport Level Security

FCS_TLS_EXT.1.1 The VoIP client application shall implement one or more of the following protocols TLS 1.2 (RFC 5246) using mutual authentication with certificates and supporting the following ciphersuites:

Mandatory Ciphersuites in accordance with RFC 3268:

- TLS_RSA_WITH_AES_128_CBC_SHA

Optional Ciphersuites:

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 6460

FCS_TLS_EXT.1.2 The VoIP client application shall not establish a trusted channel if the distinguished name (DN) contained in a certificate does not match the expected DN for the peer.

5.3.2 User Data Protection (FDP)

FDP_VOP_EXT.1 Voice Over IP Data Protection

FDP_VOP_EXT.1.1 The VoIP Client Application shall stop the transmission of voice data when a VoIP call is placed on mute, a VoIP call is not connected, no other selections and *no other actions*.

5.3.3 Identification and Authentication (FIA)

FIA_SIPC_EXT.1 Session Initiation Protocol (SIP) Client

FIA_SIPC_EXT.1.1 The VoIP client application 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_SIPC_EXT.1.2 The VoIP client application shall require the user to enter a password to support the use of password authentication for SIP REGISTER function requests as specified in section 22 of RFC 3261.

FIA_SIPC_EXT.1.3 The VoIP client application 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_SIPC_EXT.1.4 The password entered by the user as per FIA_SIPC_EXT.1.2 shall be cleared by the VoIP client application once the VoIP client application is notified that the REGISTER request was successful.

FIA_X509_EXT.1 Extended: X509 Certificate Validation

FIA_X509_EXT.1.1 The VoIP client application shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- Validate the certificate path by ensuring the basicConstraints extension is present and the cA flag is set to TRUE for all CA certificates.
- Validate the revocation status of the certificate using Certificate Revocation List (CRL) as specified in RFC 5759.
- 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).
    - 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.

FIA_X509_EXT.1.2 The VoIP client application 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 Extended: X509 Certificate Use and Management

FIA_X509_EXT.2.1 The VoIP client application 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 VoIP client application cannot establish a connection to determine the validity of a certificate, the VoIP client Application shall accept the certificate.

FIA_X509_EXT.2.3 The VoIP client Application shall not establish a trusted communication channel if the peer certificate is deemed invalid.

5.3.4 Security Management (FMT)

FMT_SMF.1(1) Specification of Management Functions

FMT_SMF.1.1(1) The VoIP client application shall be capable of performing the following management functions:

- Specify the SIP Server to use for connections,
- Specify VoIP client credentials to be used for connections,
- Specify password requirements for SIP authentication,
- Ability to configure all security management functions identified in other sections of this PP,
- no other functions.

FMT_SMF.1(2) Specification of Management Functions

FMT_SMF.1.1(2) The VoIP client Application, client device platform shall be capable of performing the following management functions:

- Configure cryptographic algorithms associated with protocols mandated in this PP,
- Load X5.09v3 certificates used for security functions in this PP,
- Configure certificate revocation check,
- Ability to update the TOE, and to verify the updates
- Ability to configure all security management functions identified in other sections of this PP,
- no other actions.

Application Note: Ability to update the TOE, and to verify the updates is performed by the client device platform.

5.3.5 Protection of the TSF (FPT)

FPT_TST_EXT.1 Extended: TSF Self Test

FPT_TST_EXT.1.1 The VoIP Client Application, client device platform shall run a suite of self tests during initial start-up (on power on) to demonstrate correct operation of the TSF.

FPT_TST_EXT.1.2 The VoIP Client Application shall provide the capability to verify the integrity of stored TSF executable code when it is loaded for execution through the use of the TSF-provided cryptographic services.

FPT_TUD_EXT.1 Extended: Trusted Update

FPT_TUD_EXT.1.1 The TSF shall provide the client device platform the ability to query the current version of the TOE firmware/software.

FPT_TUD_EXT.1.2 The client device platform shall provide authorized administrators the ability to initiate updates to the TOE firmware/software.

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

5.3.6 Trusted Path/Channel (FTP)

FTP_ITC.1(1) Inter-TSF Trusted Channel (SDES-SRTP)

FTP_ITC.1.1(1) **Refinement:** The VoIP Client Application shall provide a communication channel between itself and a **remote VoIP application using SDES-SRTP as specified in FCS_SRTP_EXT.1** 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(1) The VoIP Client Application shall permit the TSF or the remote VoIP application to initiate communication via the trusted channel.

FTP_ITC.1.3(1) The VoIP Client Application shall initiate communication via the trusted channel for [all communications between the two devices].

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

FTP_ITC.1.1(2) **Refinement:** The VoIP Client Application shall provide a communication channel between itself and **a SIP Server using TLS and no other protocol as specified in FCS_TLS_EXT.1 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 disclosure.

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

FTP_ITC.1.3(2) The VoIP Client Application shall initiate communication via the trusted channel for [all communications with the SIP server].

FTP_ITC.1(3) Inter-TSF trusted channel (Refined)

FTP_ITC.1.1(3) Refinement: The **VoIP Client Application** shall be **capable of using TLS to** provide a trusted communication channel between itself and **authorized IT entities supporting the following capabilities: configuration management 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(3) The **VoIP Client Application** shall permit the **TSF, or the authorized IT entities** to initiate communication via the trusted channel.

FTP_ITC.1.3(3) The **VoIP Client Application** shall initiate communication via the trusted channel for *SCA Server services*.

5.4 Assurance Requirements

71 The TOE security assurance requirements are summarized in Table 9.

Table 9: 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 Secure Tunnels

72 The TOE implements TLS and SDES-SRTP to secure communications with the SIP Server and remote VoIP applications.

Table 10: Secure Tunnels SFRs

SFR	Fulfilment
FIA_SIPC_EXT.1 FCS_SRTP_EXT.1	**Client registration to SecuSUITE** Before a SecuSUITE client can exchange messages with the SIP Server, it must first be registered to the SecuSUITE infrastructure via the SCA Server. This initial client registration 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 (or it is pushed via an MDM) and launches client app. 3. Client app running on mobile device prompts the user to enter the activation code and initiates a TLS connection to SCA Server 4. The user is notified to define a device password in case no device password is defined yet. 5. SCA Server validates client for registration via activation code (this is not the SIP password) 6. Client generates multiple certificate signing requests and submits to SCA Server 7. SCA server's embedded CA creates, signs and returns the certificates 8. Client gets its client configuration settings from SCA server 9. 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 and password (displayed to the user who needs to memorize the password) 10. User performs the following: a. Come up with and enter password for client application b. Enter unique activation code c. Enter validation code d. Memorize / acknowledge SIP password

SFR	Fulfilment
	11. 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. The clear text SIP password is immediately cleared from RAM. The TOE keeps the H(A1) value in RAM per section 6.5.1. **Client Registration with SIP Server** The client registers with the SIP server every time a new connection with the SIP server is established. For example, 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 Server messages 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 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 **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

SFR	Fulfilment
	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 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 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

SFR	Fulfilment
	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 RTPProxy. Clients will also terminate a call when no RTP data is received for more than 15 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. **SRTP/SRTCP** Clients send voice data via SRTP/SRTCP in compliance with (RFC3711, 2004). Clients exchange key and salt information for SRTP/SRTCP in an SDP crypto attribute in compliance with (RFC4568, 2006). The only supported crypto suite is AES_CM_128_HMAC_SHA1_80 (see (RFC4568, 2006) section 6.2) which includes:

SFR	Fulfilment
	• AES128 Counter Mode stream cipher • HMAC-SHA1 message authentication with an 80-bit authentication tag and a 160-bit authentication key Clients only offer this crypto suite in their SDP. Therefore, clients cannot negotiate other crypto suites using the offer/answer model. Clients exchange their respective master uplink key and master uplink salt during call setup in a body part of a SIP message containing a CMS/ASN.1 envelope-data structure. From the Master uplink key and salt, the key derivation function (KDF) specified in (RFC3711, 2004) section 4.3 derives • session keys for encryption • session keys for authentication • session salts The KDF includes SRTP and SRTCP index numbers into the respective key and salt generation. Maximum number of packets that can be sent with the same Master uplink key is • 2^{48} for SRTP • 2^{31} for SRTCP The SecuSUITE SIP Server administrator may specify the SRTP destination ports to be used by the SecuSUITE Client for SRTP communications. **SRTP NULL Algorithm** The TOE is hardcoded to only use AES_CM_128_HMAC_SHA1_80. The NULL algorithm cannot be selected. This behaviour cannot be changed by the user.
FCS_TLS_EXT.1 FTP_ITC.1(2) FTP_ITC.1(3)	The TOE implements TLS 1.2 using mutual authentication with certificates and supports the following ciphersuites: • TLS_RSA_WITH_AES_128_CBC_SHA • TLS_ECDHE_RSA_WITH _AES_256_GCM_SHA384 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 The TOE compares the server hostname/IP with the common name in the presented certificate and rejects the connection if there is a mismatch.
FDP_VOP_EXT.1	The TOE implement VoIP data protection in the following manner: • **Mute.** The TOE mute's the mobile device input microphone and thereby prevents audio data from being transmitted (i.e. there is not audio input to transmit). While no audio data is transmitted, RTP/RTCP traffic continues to keep the call alive. • **VoIP call not connected.** Voice data transmission is stopped in the following conditions: ○ Call party has terminated to call by sending "BYE". The terminating party stops sending RTP traffic after sending the BYE, the other party stops sending traffic as soon as the BYE is

SFR	Fulfilment
	received. Furthermore, the SIP Server ends the RTP session on the RTP proxy after the SIP OK message has been forwarded. o Network connection is interrupted. In case the network connection is interrupted, the client is not able to send any data. Clients will terminate a call when no RTP data is received for more than 15 seconds. The TOE does not support call hold.
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 issuer name is checked to ensure that it equals the subject name of the previous certificate in the path; • 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 client's X.509v3 certificates are generated during registration with the SCA server as follows (once the client has authenticated using the activation code – see Client registration to SecuSUITE in Table 10): • Client requests certificate template from the SCA server • SCA server returns message containing requested certificate generation information • Using the platform keystore, the client creates a new key pair according to the received key parameter / algorithm in the certificate generation information

SFR	Fulfilment

	• Client submits a certificate signing request containing
	  ○ certification request information:
	  ▪ client's subject name (a distinguished name)
	  ▪ client's subject alternative name (e.g. telephone number or email address)
	  ▪ key usage
	  ▪ extended key usage (e.g. id_kp_clientAuth, i.e. TLS WWW client authentication)
	  ▪ the new public key
	  ○ signature algorithm identifier
	  ○ digital signature on the certification request information
	• SCA server embedded CA creates and signs the certificate
	• SCA server returns the certificate
	• Client stores certificate in the platform keystore
	The above process is performed for the following relevant certificates:
	• **Client SIP TLS certificate** – used for TLS communication with the SIP server.
	• **Client SCA TLS certificate** – used for TLS communication with the SCA server.
	The TOE maintains a reference to the keystore location for each certificate and thereby able to select the correct certificate for a given usage.
	When a connection cannot be established during the validity check of a certificate the TOE will accept the certificate. The TOE rejects the connection if the peer certificate is deemed invalid

FTP_ITC.1(1)	The TOE implements SDES-SRTP as described above at SRTP/SRTCP (FCS_SRTP_EXT.1).

6.2 TOE Configuration

73 The TOE provides interfaces to control the configuration of TLS and SDES-SRTP and the underlying cryptographic mechanisms, management of X.509 certificates, and updates to the TOE.

Table 11: TOE Configuration SFRs

SFR	Fulfilment
FMT_SMF.1(1)	The TOE supports the following management functions*: • **Specify the SIP Server to use for connections.** User can enter SCA Server address during the activation phase which determines the SIP server. **Note:** This is the only parameter that the user can configure via the TOE.
FMT_SMF.1(2)	

SFR	Fulfilment
	• **Specify VoIP client credentials to be used for connections.** SCA server pushes configuration settings including username and password during registration: ○ The SIP server can be used to manually specify an alternative SIP password • **Specify password requirements for SIP authentication.** Password can be set manually from SIP server. Admin can define minimum password length, which must be 8 or more. • **Configure cryptographic algorithms associated with protocols mandated in this PP.** The algorithms are selected by the SIP server during TLS handshake. • **Load X5.09v3 certificates used for security functions in this PP** – the client's X.509v3 certificates are generated during registration with the SCA server.as described in FIA_X509_EXT.2. • **Configure certificate revocation check**. There are no configurable parameters for revocation checking. The client device platform performs the following management functions: • **Ability to update the TOE, and to verify the updates.** The user may uninstall or update the TOE using the mobile device operating system and app store. Downloaded TOE updates (apps) are verified using digital signatures in accordance with supported mobile device Security Targets (see FPT_TUD_EXT.1 in Table 12). *Per VoIP PP FMT_SMF.1 application note: *There may be some instances where a SIP Server "pushes" configuration information down to the VoIP client application. This is an acceptable form of management…*

6.3 Verifiable Updates

74 The TOE uses digital signatures to verify software updates.

Table 12: Verifiable Updates SFRs

SFR	Fulfilment
FPT_TUD_EXT.1	The user may install or update the TOE using the mobile device operating system and app store. Alternatively, an MDM may be used to push the TOE app and updates to the user's mobile device – in such cases user interaction / acceptance is still required. TOE updates are signed by with the Secusmart software signing key associated with each platform. Downloaded TOE updates (apps) are verified using digital signatures in accordance with supported mobile device Security Targets. Please refer to the following documents for additional information regarding the update verification mechanism and for details on how the certificates used by the update verification mechanism are contained on the device: • **Blackberry.** Refer to section 7.6.11 of the BlackBerry Smartphones with OS 10.3.3 Security Target. • **Samsung.** Additional details relating to Samsung / Android application signing (that was excluded from the related Security Target) is provided at http://developer.android.com/tools/publishing/app-signing.html • **Apple.** Refer to FPT_TUD_EXT.2 in Table 9 of Apple iOS 9.3 Security Target. The user may determine the installed version of the TOE by navigating to the 'about' section of the settings menu within the app.

6.4 Self Test

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

Table 13: Self Test SFRs

SFR	Fulfilment
FPT_TST_EXT.1	The TOE relies on the mobile device to perform start-up self-tests to ensure correct operation of the OS and supporting hardware, enabling the correct operation of the TSF. Refer to the following documents for platform specific details: • **Blackberry.** Refer to sections 7.6.8 and 7.6.9 of the BlackBerry Smartphones with OS 10.3.3 Security Target. • **Samsung.** Refer to FPT_TST_EXT.1 on page 51 of the Samsung Galaxy Devices on Android 6 (MDFPP20) Security Target. • **Apple.** Refer to FPT_TST_EXT.1 in Table 9 of Apple iOS 9.3 Security Target. The TOE's OpenSSL cryptographic module performs start-up self-testing to ensure the correct operation of the cryptographic module. Refer to Annex C: OpenSSL Crypto Module Self-tests for detail of these tests. The TOE's OpenSSL cryptographic module is used to perform an integrity check (HMAC_SHA1) of the TSF executable code. If the integrity check succeeds the client continues with the start-up procedure. If the integrity check fails the TOE does not start-up at all. As described above, self-testing ensures the integrity and correct operation of the TOE platform, the cryptographic module and the TOE application code. These combined tests ensure the integrity of all TSF critical components, demonstrating that the TSF is operating correctly.

6.5 Cryptographic Modules

76 The TOE makes use of the following cryptographic modules:

a) **TOE (app) module.** OpenSSL FIPS Object Module Version 2.0.12. Relevant CAVP certificates: AES (#4382), CVL (#1079 & #1080), DRBG (#1408), ECDSA (#1046), HMAC (#2910), RSA (#2368), KAS (#112) and SHA (#3610).

b) **Platform modules.** The TOE makes use of platform provided modules for TLS related ECDSA key generation, sign and verify operations. These modules are:

   i) **Blackberry.** Blackberry Cryptographic Library Version 5.6.2 as described in section 7.2 of the BlackBerry Smartphones with OS 10.3.3 Security Target. Relevant CAVP certificates: ECDSA (#199)

ii) **Samsung.** BoringSSL as described in section 6.1 of the Samsung Galaxy Devices on Android 6 (MDFPP20) Security Target. Relevant CAVP certificates: ECDSA (#857)

iii) **Apple.** Apple iOS CoreCrypto Module v6.0 as described in section 1.5.2.1 of the Apple iOS 9.3 Security Target. Relevant CAVP certificates: ECDSA (#783).

77

Table 14: Cryptographic Module SFRs

SFR	Fulfilment
FCS_CKM.1(1)	The TOE OpenSSL module performs key generation for use in key establishment and authentication in RSA and ECDSA schemes with the exception that static ECDSA key pairs for SIP and SCA TLS are created using the platform provided cryptographic module. The TOE's RSA key establishment scheme generally fulfils all of the NIST SP 800-56B requirements without extensions, Table 15 specifically identifies the "should", "should not", and "shall not" conditions from the publication along with an indication of how the TOE conforms to those conditions. The TOE's ECDSA key establishment scheme complies with the following sections of NIST Special Publication 800-56A: • 5.6.1.2 ECC Key-Pair Generation • 5.7.1.2 Elliptic Curve Cryptography Cofactor Diffie-Hellman (ECC CDH) Primitive • 6.1.1 C(2e, 2s) Schemes
FCS_CKM.1(2)	
FCS_CKM_EXT.2(1)	The TOE stores all persistent secrets and private keys when not in use in client device platform-provided key storage as follows: • **BlackBerry.** Cert Manager. • **Samsung.** AndroidKeyStore. • **Apple.** iOS keychain. See section 6.5.1 for details of CSPs and related storage.
FCS_CKM_EXT.4(1)	None of the TOE's symmetric keys, pre-shared keys, or private keys are stored in plaintext form. See section 6.5.1 for details of CSPs and related zeroization.
FCS_CKM_EXT.4(2)	Please refer to FCS_CKM_EXT.4 in mobile device platform STs for details of zeroization for platform provided key storage.
FCS_COP.1(1)	The TOE OpenSSL module performs AES encryption and decryption in CTR, CBC and GCM mode.
FCS_COP.1(2)	The TOE OpenSSL module performs RSA and ECDSA cryptographic signature services (generation and verification).

<table>
<tr><th>SFR</th><th>Fulfilment</th></tr>
<tr><td>FCS_COP.1(3)</td><td>The TOE OpenSSL module performs SHA-1, SHA-256 and SHA-384 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
Signed-data signature
Digest Access Authentication
SHA-384
Certificate signature
TLS PRF with AES256-GCM</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)
RTP/SRTCP integrity protection (HMAC-SHA-1)
TSF integrity check
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
RTP/SRTCP truncates to 80 bits in accordance with RFC 3711 section 4.2.1</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 the client device platform RBG to seed the OpenSSL CTR_DRBG (AES). The client device platform RBG functionality is invoked as follows: • **BlackBerry.** Invoked via /dev/random/ • **Samsung.** Invoked via OpenSSL RAND API • **Apple.** Invoked via SecRandomCopyBytes that reads random from /dev/random

Table 15: SP800-56B Conformance (FCS_CKM.1(2))

SP800-56B Section	Requirement	Fulfilment
5.4	Should (1st)	Met (relevant for TLS_RSA_WITH_AES_128_ CBC_SHA)
5.4	Should (2nd)	
5.4	Should (3rd)	
5.4	Should (4th)	
5.4	Should (5th)	
5.5	Shall not	
5.5.1.2	Should (1st)	
5.5.1.2	Should (2nd)	
5.5.1.2	Should (3rd)	
5.5.1.2.3	Should	
5.6.1.1	Shall not (1st)	
5.6.1.1	Shall not (2nd)	
6.1	Shall not (1st)	Not Applicable – the TOE does not have RSA key pairs.
6.1	Shall not (2nd)	
6.1	Should not	
6.4.1.5	Should	
6.4.2.3	Should (1st)	Met (relevant for TLS_RSA_WITH_AES_128_ CBC_SHA)
6.4.2.3	Should (2nd)	

SP800-56B Section	Requirement	Fulfilment
6.4.2.3.1	Should	
7.1.2	Should	Not Applicable – the TOE does not use RSA-OAEP
7.2.1.3	Should	
7.2.1.3	Should not	
7.2.2.2	Shall not	
7.2.2.4	Shall not	
7.2.2.4	Should (1st)	
7.2.2.4	Should (2nd)	
7.2.2.4	Should (3rd)	
7.2.2.4	Should (4th)	
7.2.2.4	Should not	
7.2.3.2.1	Should	Not Applicable – the TOE does not use KTS-KEN-KWS
7.2.3.2.2	Should	
7.2.3.2.3	Shall not	
7.2.3.2.3	Should	
7.2.3.2.4	Should	
7.2.3.4	Should (1st)	
7.2.3.4	Should (2nd)	
7.2.3.4	Should (3rd)	
7.2.3.4	Should (4th)	
7.2.3.4	Should not	
8	Should	Not Met – the TOE only uses RSA for TLS_RSA_WITH_AES_128_CBC_SHA. This TLS cipher suite does not use key confirmation.
8.3.2	Should not	

SP800-56B Section	Requirement	Fulfilment
9.1	Should	Not Applicable – does not apply to RSA key transport schemes.

6.5.1 Cryptographic Keys and CSPs

78 Table 16 describes the keys and CSPs utilized by the TOE. **Note:** In all cases key destruction (referred to by 'destroys') is performed by overwrite with read-after-write verify.

Table 16: Cryptographic Keys and CSPs

Name	Description / Storage / Destruction
SIP Digest Access Authentication	User must enter SIP password when starting the client application. TOE calculates H(A1) hash of SIP password. TOE application destroys SIP password from application controlled RAM when H(A1) has been calculated. The TOE relies on the client device platform to destroy the SIP password on the Java layer (user interface). TOE application keeps H(A1) in RAM for later re-registration with SIP server.
TLS ECDSA static private key	BB10 Platform keystore is responsible for creation, storage and destruction. ECDSA sign and verify are calculated using platform APIs and private keys are not extracted from the platform keystore. Platform methods used for key generation and storage: • hu_ECCParamsCreate • tp_StoreObjAdd • tp_StoreObjGenKey
	iOS Platform keystore is responsible for creation, storage and destruction. ECDSA sign and verify are calculated using platform APIs and private keys are not extracted from the platform keystore Platform methods used for key generation and storage: • SecKeyGeneratePair • CFDictionaryAddValue
	Android The ECDSA static private key is created using OpenSSL module and persistently stored to the platform keystore. The TOE destroys the private key in RAM after key has been persistently stored.

Name	Description / Storage / Destruction
	ECDSA sign and verify are calculated using platform APIs and private keys are not extracted from the platform keystore
TLS ECDSA random ephemeral secret	Created and destroyed by OpenSSL module (RAM).
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	
RTP/SRTP – SRTP master key	During call setup, client application creates own and receives peer SRTP master key. The SRTP master key is stored in RAM only and the TOE application destroys these when call has ended.
RTP/SRTP - Session keys	TOE application creates session keys in RAM when call starts and destroys these when call has ended.
CSPRNG – Seed	TOE application reads seed from /dev/random into RAM and passes this to OpenSSL. Application destroys seed in RAM after this.
HMAC Integrity Key	The HMAC integrity key is used by the OpenSSL module to perform the software integrity self-check. The key is stored as a constant value to the source code.

7 Rationale

7.1 Conformance Claim Rationale

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

a) **TOE type.** As identified in section 2.1, the TOE is a network device, consistent with the TOE type identified by the VoIP PP.

b) **Security problem definition.** As shown in section 3, the threats, OSPs and assumptions are identical to those of the VoIP PP.

c) **Security objectives.** As shown in section 4, the security objectives are identical to those of the VoIP PP.

d) **Security requirements.** As shown in section 5, the security requirements are reproduced from the VoIP PP. No additional requirements have been specified.

7.2 Security Objectives Rationale

80 All security objectives are drawn directly from the VoIP PP.

7.3 Security Requirements Rationale

81 All security requirements are drawn directly from the VoIP PP.

7.4 TOE Summary Specification Rationale

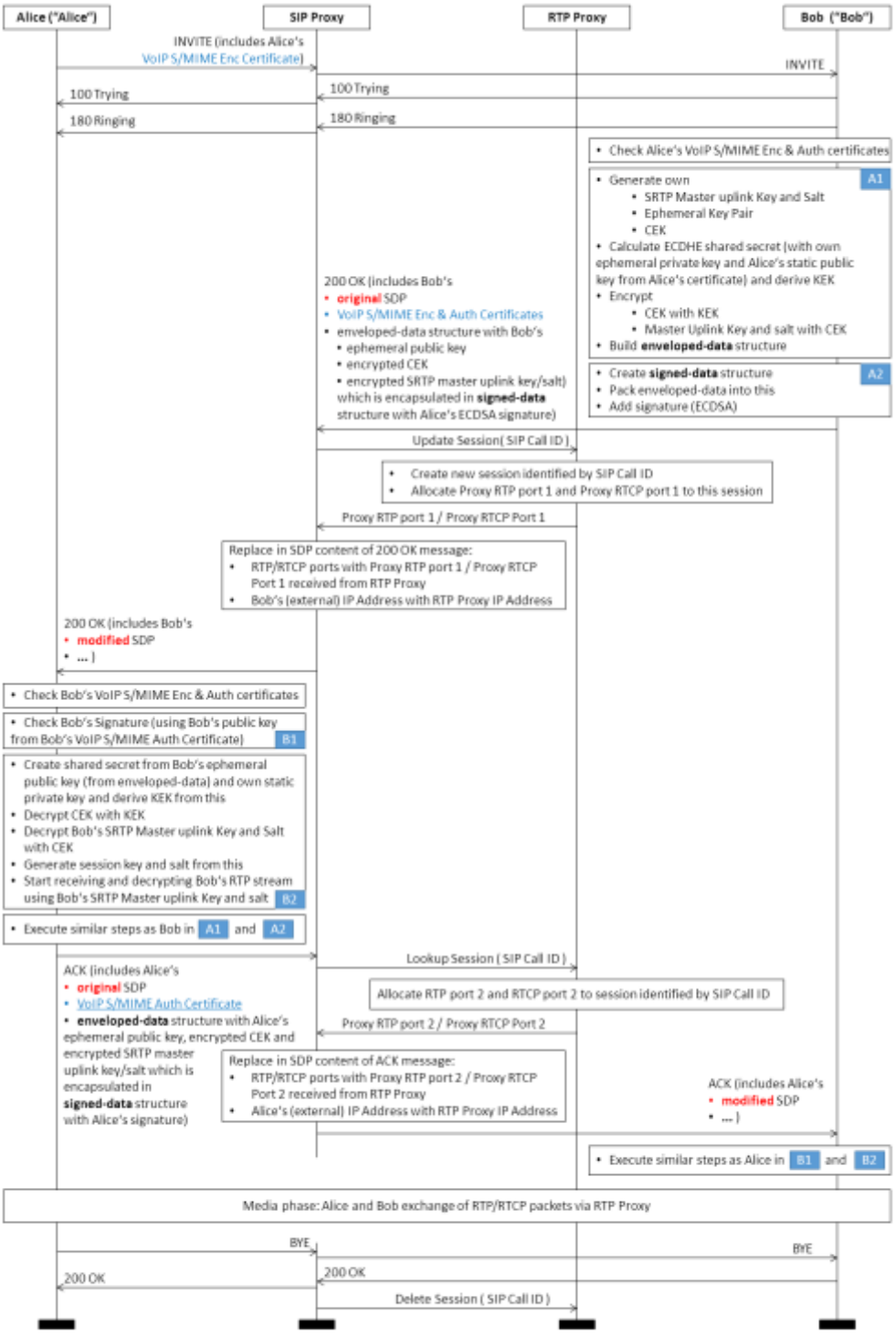
82 Table 17 provides a coverage mapping showing that all SFRs are mapped to the security functions described in the TSS.

Table 17: Map of SFRs to TSS Security Functions

	Secure Tunnels	TOE Configuration	Verifiable Updates	Self Tests	Cryptographic Modules
FCS_CKM.1(1)					X
FCS_CKM.1(2)					X
FCS_CKM_EXT.2(1)					X
FCS_CKM_EXT.4(1)					X

FCS_CKM_EXT.4(2)					X
FCS_COP.1(1)					X
FCS_COP.1(2)					X
FCS_COP.1(3)					X
FCS_COP.1(4)					X
FCS_RBG_EXT.1					X
FCS_SRTP_EXT.1	X				
FCS_TLS_EXT.1	X				
FDP_VOP_EXT.1	X				
FIA_SIPC_EXT.1	X				
FIA_X509_EXT.1	X				
FIA_X509_EXT.2	X				
FMT_SMF.1(1)		X			
FMT_SMF.1(2)		X			
FPT_TST_EXT.1				X	
FPT_TUD_EXT.1			X		
FTP_ITC.1(1)	X				
FTP_ITC.1(2)	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)

Annex C: OpenSSL Crypto Module Self-tests

83 The Module performs the self-tests listed below on invocation of Initialize or Self-test.

Algorithm	Type	Test Attributes
Software integrity	KAT	HMAC-SHA1
HMAC	KAT	One KAT per SHA1, SHA224, SHA256, SHA384 and SHA512 Per IG 9.3, this testing covers SHA POST requirements.
AES	KAT	Separate encrypt and decrypt, ECB mode, 128 bit key length
AES CCM	KAT	Separate encrypt and decrypt, 192 key length
AES GCM	KAT	Separate encrypt and decrypt, 256 key length
XTS-AES	KAT	128, 256 bit key sizes to support either the 256-bit key size (for XTS-AES-128) or the 512-bit key size (for XTS-AES-256)
AES CMAC	KAT	Sign and verify CBC mode, 128, 192, 256 key lengths
TDES	KAT	Separate encrypt and decrypt, ECB mode, 3-Key
TDES CMAC	KAT	CMAC generate and verify, CBC mode, 3-Key
RSA	KAT	Sign and verify using 2048 bit key, SHA-256, PKCS#1
DSA	PCT	Sign and verify using 2048 bit key, SHA-384
DRBG	KAT	CTR_DRBG: AES, 256 bit with and without derivation function HASH_DRBG: SHA256 HMAC_DRBG: SHA256 Dual_EC_DRBG: P-256 and SHA256
ECDSA	PCT	Keygen, sign, verify using P-224, K-233 and SHA512. The K-233 self-test is not performed for operational environments that support prime curve only (see Table 2).
ECC CDH	KAT	Shared secret calculation per SP 800-56A §5.7.1.2, IG 9.6
X9.31 RNG	KAT	128, 192, 256 bit AES keys

Power On Self Tests (KAT = Known answer test; PCT = Pairwise consistency test)

84 The HMAC-SHA-1 of the Module is verified during installation of the Module file.

The `FIPS_mode_set()` function performs all power-up self-tests listed above with no operator intervention required, returning a "1" if all power-up self-tests succeed, and a "0" otherwise. If any component of the power-up self-test fails an internal flag is set to prevent subsequent invocation of any cryptographic function calls. The module will only enter the FIPS Approved mode if the module is reloaded and the call to `FIPS_mode_set()` succeeds.

The power-up self-tests may also be performed on-demand by calling FIPS_selftest(), which returns a "1" for success and "0" for failure. Interpretation of this return code is the responsibility of the calling application.

The Module also implements the following conditional tests:

Algorithm	Test
DRBG	Tested as required by [SP800-90] Section 11
DRBG	FIPS 140-2 continuous test for stuck fault
DSA	Pairwise consistency test on each generation of a key pair
ECDSA	Pairwise consistency test on each generation of a key pair
RSA	Pairwise consistency test on each generation of a key pair
ANSI X9.31 RNG	Continuous test for stuck fault

In the event of a DRBG self-test failure the calling application must uninstantiate and re-instantiate the DRBG per the requirements of [SP 800-90]; this is not something the Module can do itself.

Pairwise consistency tests are performed for both possible modes of use, e.g. Sign/Verify and Encrypt/Decrypt.