Security Target

Cellcrypt Android Mobile Client v5.0

Ref:	ST-FED-MCL-And-2
Ver:	0.5.1
Date:	Nov 26, 2025

Contents

Figures

Tables

Releases

Issue	Description
0.1.0	Initial release.
0.2.0	Made several changes in response to initial comments.
0.3.0	Answered questions and modified according to second round of comments.
0.3.1	Updated FCS_COP.1.1/KeydHash and detail for FCS_TLSC_EXT.6 in TSS.
0.4.0	Updated FCS_RBG_EXT.1 and added FCS_RBG_EXT.2.
0.4.1	Updated FCS_RBG_EXT.2.1 and the TSS description.
0.4.2	Changed Samsung phone version to S23 and S21.
0.4.3	Added Android 14 reference.
0.4.4	Updated ciphersuites (FCS_TLSC_EXT.1.2). Updated TDs and TSS
0.4.5	Updated FCS_TLSC_EXT.1.4
0.4.6	Updated FCS_TLSC_EXT.2.1, FPT_LIB_EXT.1.1, and TSS
0.4.7	Updated FDP_DEC_EXT.1.1, FCS_TLSC_EXT.1.4, section 1.3 and TSS
0.4.8	Updated FCS_TLSC_EXT.1.5, FDP_DEC_EXT.1.2
0.4.9	Removed CRL capability (OCSP only). Updated FPT_LIB_EXT.1.1 (and in TSS)
0.4.10	Updated Logical scope of the TOE and removed variable settings for SRTP port
0.4.11	Added CAVP certificate number. Added TDs 945, 931, 914
0.5.0	Updated Table 2 and some TD application corrections
0.5.1	Added section 1.5 Excluded Functionality

1. ST INTRODUCTION

This Security Target (ST) specifies the requirements for the Cellcrypt Android Mobile Client Target of Evaluation (TOE) for evaluation and certification under the Common Criteria (CC).

The format and content are in accordance with Common Criteria assurance class ASE and the requirements of the Protection Profiles, Functional Packages, PP-Modules and PP-Configurations stated in Section 1.5.

1.1. ST and TOE Reference

Table 1 ST and TOE Reference information

Attribute	Description
ST Title	Cellcrypt Android Mobile Client Security Target
ST Version	0.5.1
ST Reference	ST-FED-MCL-And-2
ST Date	Nov 26, 2025
TOE Title	Cellcrypt Android Mobile Client
TOE Version (Android)	5.0.0
TOE Developer	Cellcrypt
CC Version	Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, April 2017

1.2. TOE Overview

Cellcrypt Android Mobile Client is a secure multimedia application for Android smartphones. It implements end-to-end encryption and authentication of voice, video, text messages and file attachments between two or more users of Cellcrypt Android Mobile Client and other compatible applications. The Cellcrypt system comprises a handset software application (Cellcrypt Android Mobile Client, i.e. the TOE) and the back-end support infrastructure (Cellcrypt Server). The TOE is the handset software application, Cellcrypt Android Mobile Client, on a specific hardware platform (described below).

Cellcrypt Android Mobile Client uses standard wireless packet-based connectivity that can be provided by a cellular network or a Wi-Fi data connection.

Mutually authenticated connection set-up ensures that only mobile phones on which the TOE runs can participate in secure sessions with the Cellcrypt Server, and that the users of the TOE can be assured to always connect to a legitimate Cellcrypt server. End-to-end encryption is achieved through the creation and use of session-unique encryption/decryption keys used by the TOE to encrypt and decrypt voice traffic, messages, and attachments. Long-term static

keys and other sensitive user data are stored by the TOE in an encrypted database (SQLCipher) with the SQLCipher master key being protected by the operating system.

The following prerequisites must apply in the use of the TOE:

- The TOE hardware platforms are:
    - Samsung Galaxy S23 running Android 14.0 on a Qualcomm Snapdragon 8 Gen2 processor with Processor Algorithm Accelerators (PAA).
    - Samsung Galaxy S21 running Android 14.0 on a Qualcomm Snapdragon 888 processor with Processor Algorithm Accelerators (PAA).
- The TOE runs on a NIAP-validated configuration of a mobile platform (including VPN), as defined by the Protection Profile for Mobile Device Fundamentals. The mobile platform is outside the scope of the evaluation.
- ESC Server, as defined by the PP-Module for Enterprise Session Controller (ESC) is outside the scope of this evaluation.
- The TOE operates exclusively within the mobility ecosystem specified by the associated mobility Protection Profiles and will assume that all associated resources (IPSEC VPN tunnel, SIP network) are in place.

The non-TOE components required by the TOE are the following:

- OCSP responder for use in the verification of X.509 certificates.
- Cellcrypt Server for client authentication and other services e.g. SIP, messaging/attachments and check for updated software.

1.3. TOE Description

The Target of Evaluation (TOE) is the Cellcrypt Android Mobile Client application (Figure 1), which runs on Android 14. The Cellcrypt Android Mobile Client application is a software cryptographic application for smartphones. The core function of the TOE is to allow users' voice and video calls to be encrypted with end-to-end security.

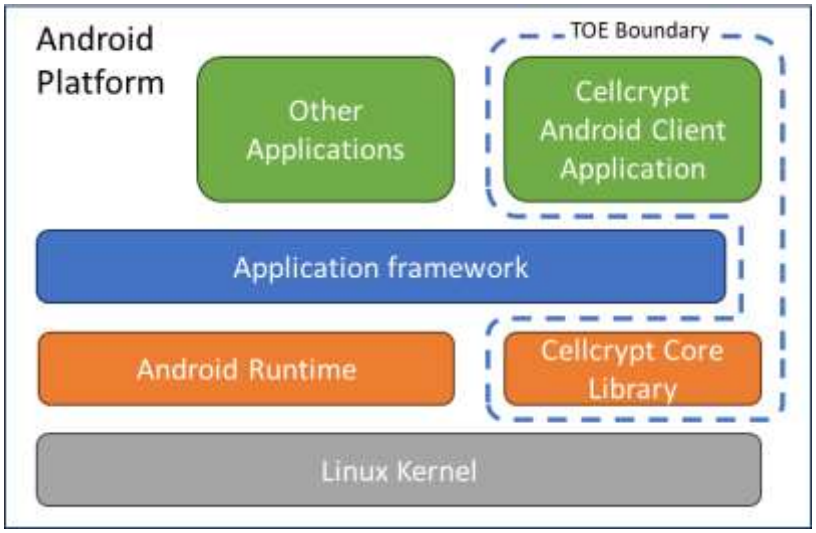

Figure 1 TOE Boundary

1.3.1. Physical Scope of the TOE

The physical scope of the TOE comprises of the following:

- The TOE Software, i.e. the Cellcrypt Android Mobile Client Version 5.0.0.
- TOE Security Guidance: Common Criteria Guidance - Cellcrypt Android Mobile Client, AG-FED-MCL-And-2, Version 0.2.0, Jun, 3 2025.

1.3.2. Logical Scope of the TOE

The logical scope of the TOE comprises of the following:

- Authenticated call set-up with the Cellcrypt Server.
- Extended authentication confirmation via the CA OCSP services.
- End-to-end encryption of secure voice and video traffic.
- Security management functions restricted to authorized personnel.
- Protection measures for ensuring the integrity and authenticity of the TOE.

1.3.2.1. Authenticated call set-up

The TOE uses X.509 Certificates for mutual authentication on the trusted channel between itself and the Cellcrypt Server. The TOE can use TLSv1.3 or TLSv1.2 protocol to protect all communications between itself and the Cellcrypt Server from modification and disclosure. In addition to X.509 Certificate authentication, the TOE also authenticates the user to the Cellcrypt Server using a password. The TOE does not store the authentication password but requests the user to enter it each time it is required.

1.3.2.2. Extended authentication

The validity of the X.509 certificate is checked by querying the authority CA's OCSP responder indicated in the Authority Information Access (AIA) X.509 extension in the certificate.

1.3.2.3. End-to-end protection of voice and video

The TOE provides end-to-end encryption using an SDES-SRTP trusted channel. The keys for the SDES-SRTP trusted channel are protected by the TLS/SIP channel during key establishment. The TOE mitigates side channel attacks by utilizing a fixed rate vocoder. This prevents an attacker from inferring information about the audio from the bitrate being transmitted.

1.3.2.4. Security management

The TOE is configured to be secure by default but also provides administrator and user interfaces for security-relevant configuration and maintenance. This includes controlled mechanisms for application deployment and updates using platform-supported deployment mechanisms. User management settings allow selection of ciphers and other secure communication options. Users can also configure access to the TOE using a PIN/password or a fingerprint.

1.3.2.5. TOE integrity and authenticity

During deployment the TOE is protected by a secure digital signature which is authenticated by the platform before allowing the TOE to be installed. The TOE software takes advantage of modern smart phone platform security architectures and provides a secure runtime environment including the use of compiler/linker security options e.g. Address space layout randomization (ASLR) and stack-based overflow protections. All communications include cryptographic integrity and authentication.

1.4. TOE Cryptography

All TOE cryptography is performed using OpenSSL 3.5 in FIPS mode. CAVP Certificate references are given in Table 2. The TOE cryptographic support includes functions supporting key management, encryption and decryption, random number generation, digital signatures, secure hashing, and keyed secure hashing. Cryptographic protocol support includes TLS.

Table 2 CAVP Certificate References

Algorithms	Options	SFR Reference	Cert#
AES (FIPS 197)	Modes: CTR, GCM (SP 800-38A, SP 800-38D) Key lengths: 128, 256 bits	FCS_COP.1.1/SKC FCS_COP.1.1/SRTP	A7364
SHA-1 (FIPS 180-4)	Hash lengths: 160 bits	FCS_COP.1.1/Hash	A7364
SHA2 (FIPS 180-4)	Hash lengths: 256, 384, 512 bits	FCS_COP.1.1/Hash	A7364
HMAC (FIPS 198)	Hash lengths: 160, 256, 384, 512 bits	FCS_COP.1.1/KeyedHash	A7364
RSA (FIPS 186-4)	KeyGen Key length: 2048, 3072, and 4096 bits	FCS_CKM.1.1/AK	A7364

Algorithms	Options	SFR Reference	Cert#
RSA (FIPS 186-5)	SigGen, SigVer Key length: 2048, 3072, and 4096 bits	FCS_CKM.1.1/Sig	A7364
Safe Prime (RFC 7919)	KeyGen Key length: 2048, 3072, 4096, 6144, and 8192 bits	FCS_CKM.1/AK	A7364
KAS-FFC (SP 800-56Ar3)	KeyExch Key length: 2048, 3072, 4096, 6144, and 8192 bits	FCS_CKM.2	A7364
KAS-ECC (SP 800-56Ar3)	KeyExch Curves: P-256, P-384, P-521	FCS_CKM.2	A7364
ECDSA (FIPS 186-5)	KeyGen, SigGen, SigVer Curves P-256, P-384, P-521	FCS_CKM.1/AK FCS_COP.1.1/Sig	A7364
DRBG (SP 800-90Ar3)	CTR_DRBG: (AES-256)	FCS_RBG_EXT.2	A7364

1.5. Excluded Functionality

In the case of CellCrypt to CellCrypt device communication, there is an additional cryptographic layer that is tunnelled through the call for additional security. The presence or functionality of this is outside the scope of the Protection Profiles and therefore is outside the scope of the evaluation.

2. CONFORMANCE CLAIMS

2.1. CC Conformance Claims

The ST and the TOE it describes are conformant to the following CC versions:

- Common Criteria for Information Technology Security Evaluation Part 1, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluation Part 2, Version 3.1, Revision 5, April 2017: Part 2
- Common Criteria for Information Technology Security Evaluation Part 2, Version 3.1, Revision 5, April 2017: Part 3

The ST is Common Criteria Part 2 extended conformant and Common Criteria part 3 extended conformant.

The ST is package conformant to: **none**. The Security Assurance Components are taken exactly from the Protection Profiles, Protection Profile Modules and Functional packages. They do not constitute an Evaluation Assurance Level or other assurance package.

2.2. Protection Profile Conformance

The TOE claims exact conformance to:

- Protection Profile for Application Software Version 1.4, 2021-10-07 (AppPP)

- PP-Module for Voice and Video over IP (VVoIP) Version 1.0, 2020-10-28 (MOD_VVoIP)
- Functional Package for Transport Layer Security (TLS) Version 2.0, 2022-12-19 (TLS-PKG)

Protection Profile Conformance is claimed in accordance with the following:

- PP-Configuration for Application Software and Voice/Video over IP (VVoIP) Endpoints (CFG_APP-VVoIP_V1.1)

The following NIAP Technical Decisions (TD) apply to the AppPP, MOD_VVoIP and TLS-PKG. Their applicability to the evaluation was determined based on whether the TD is current (i.e. not superseded) and whether the SFRs referenced by the TD are included in the ST:

Table 3: Technical Decisions

TD	Applies	PP/MOD	Exclusion Rationale
0945: Adding FIPS 186-5 in PP_APP_V1.4	Yes	AppPP	
0931: Clarification when CTR_DRBG is Selected for FCS_RBG_EXT.2.2 in PP_APP_V1.4	Yes	AppPP	
0893: Addition of Recommended Configuration Locations for Windows in FMT_MEC_EXT.1.1	No	AppPP	Applies to Windows apps.
0865: Consistency of Cryptographic Key Sizes	Yes	AppPP	
0844: Addition of Assurance Package for Flaw Remediation V1.0 Conformance Claim	No	AppPP	Applies to Flaw Remediation.
0823: Update to Microsoft Windows Exploit Protection link in FPT_AEX_EXT.1.3	No	AppPP	Applies to Windows apps.
0822: Correction to Windows Manifest File for FDP_DEC_EXT.1	No	AppPP	Applies to Windows apps.
0815: Addition of Conditional TSS Activity for FPT_AEX_EXT.1.5	Yes	AppPP	
0798: Static Memory Mapping Exceptions	Yes	AppPP	
0780: FIA_X509_EXT.1 Test 4 Clarification	Yes	AppPP	
0756: Update for platform-provided full disk encryption	Yes	AppPP	

TD	Applies	PP/MOD	Exclusion Rationale
0747: Configuration Storage Option for Android	Yes	AppPP	
0743: FTP_DIT_EXT.1.1 Selection exclusivity	Yes		
0736: Number of elements for iterations of FCS_HTTPS_EXT.1	No	AppPP	FCS_HTTPS_EXT.1 is not claimed.
0719: ECD for PP APP V1.3 and 1.4	Yes	AppPP	
0717: Format Changes for PP_APP_V1.4	Yes	AppPP	
0664: Testing activity for FPT_TUD_EXT.2.2	Yes	AppPP	
0650: Conformance claim sections updated to allow for MOD_VPNC_V2.3 and 2.4	Yes	AppPP	
0628: Addition of Container Image to Package Format	Yes	AppPP	
0834: Aligning MOD_VVOIP 1.0 with NDcPP 3.0E	No	MOD_VVoIP	TOE base PP is not NDcPP.
0785: Terminology Change in MOD_VVoIP: Extended to Functional Package	Yes	MOD_VVoIP	
0730: Clarification on Trusted Update sources for VVoIP Applications	Yes	MOD_VVoIP	
0718: Format changes for MOD_VVoIP_V1.0	Yes	MOD_VVoIP	
0684: VVoIP PP-Module Updated to Allow for App PP v1.4 as Base PP	No	MOD_VVoIP	Already using AppPP v1.4 as base PP.
0589: Reliable Time for VVoIP Software TOEs	No	MOD_VVoIP	Applicable only to hardware devices.
0914: Addition of PKG_TLS_V2.0 to Conformance Claims	Yes	AppPP	Applied.
0912: TLS FP 2.0 selections in audit records table	No	TLS-PKG	FAU_GEN is not claimed.
0911: FCS_TLS_EXT.1 TSS and Guidance Evaluation Activities	Yes	TLS-PKG	
0793: Alignment of FCS_TLSS_EXT.5:2 with RFC 8446	No	TLS-PKG	FCS_TLSS_EXT.5.2 is not claimed.
0731: Clarification of TLS 2.0 FP test 22.2	No	TLS-PKG	FCS_TLSS_EXT.1 is not claimed.

TD	Applies	PP/MOD	Exclusion Rationale
0729: Self-contradictory test requirement	No	TLS-PKG	FCS_TLSS_EXT.1 is not claimed.

2.3. Conformance Rationale

The TOE is an Android application that provides voice and video communications secured using TLS. As such the TOE conformance claims in this security target claim exact Protection Profile Conformance in accordance with the following:

- PP-Configuration for Application Software and Voice/Video over IP (VVoIP) Endpoints, Version 1.1, 2022-05-31 [CFG_APP-VVoIP_V1.1].
  The PP-Configuration includes the following:
  - Protection Profile for Application Software Version 1.4, 2021-10-07 [AppPP],
  - PP-Module for Voice/Video over IP (VVoIP) Endpoints Version 1.1, 2020-10-28 [MOD_VVoIP],
- Functional Package for Transport Layer Security (TLS) Version 2.0, 2022-12-19 [TLS-PKG].

The security problem definition, security objectives and security requirements in this Security Target are all taken from the Protection Profile, the PP-Modules, and the Functional Package, applied in accordance with the applicable PP-Configurations. Only operations defined therein are performed on the security functional and security assurance components.

3. SECURITY PROBLEM DEFINITION

This section describes the assumptions and threats that are relevant to both the TOE and its environment.

The security problem definition has been taken from [AppPP] and [MOD_VVoIP] and is reproduced for the convenience of the reader. [TLS-PKG] does not state additional Security Problem Definition elements.

3.1. Threats Addressed by the TOE

Table 4: Threats Addressed by the TOE

Threat	Description
T.NETWORK_ATTACK	An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may engage in communications with the application software or alter communications between the application software and other endpoints in order to compromise it.
T.NETWORK_EAVSDROP	An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may

Threat	Description
	monitor and gain access to data exchanged between the application and other endpoints.
T.LOCAL_ATTACK	An attacker can act through unprivileged software on the same computing platform on which the application executes. Attackers may provide maliciously formatted input to the application in the form of files or other local communications.
T.PHYSICAL_ACCESS	An attacker may try to access sensitive data at rest.
T.UNDETECTED_TRANSMISSION	An attacker may cause the TOE to exfiltrate audio and/or video media over a remote channel while in a state where the user has a reasonable expectation that no media is being transmitted.
T.MEDIA_DISCLOSURE	An attacker can use the encrypted variable rate vocoder frames to their advantage to decode transmitted data.

3.2. Organizational Security Policies

The [AppPP], [MOD_VVoIP] and [TLS-PKG] do not specify any organizational security policies.

3.3. Assumptions

Table 5: Assumptions on the Operational Environment

Assumption	Description
A.PLATFORM	The TOE relies upon a trustworthy computing platform with a reliable time clock for its execution. This includes the underlying platform and whatever runtime environment it provides to the TOE.
A.PROPER_USER	The user of the application software is not wilfully negligent or hostile, and uses the software in compliance with the applied enterprise security policy.
A.PROPER_ADMIN	The administrator of the application software is not careless, wilfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy.
A.UPDATE_SOURCE	It is assumed that TOE software/firmware updates will be made available on either the call control server that the TOE connects to or a separate file server managed by the organization.

4. SECURITY OBJECTIVES

This section defines the security objectives for the TOE and its supporting environment. The security objectives are intended to counter identified threats, comply with defined organizational security policies, and address applicable assumptions.

The security objectives are taken from [AppPP], [MOD_VVoIP] and are reproduced for the convenience of the reader. There are no additional security objectives stated in [TLS-PKG].

4.1. Security Objectives for the TOE

Table 6: Security Objectives for the TOE

Security Objective	Description
O.INTEGRITY	Conformant TOEs ensure the integrity of their installation and update packages, and also leverage execution environment-based mitigations. Software is seldom if ever shipped without errors, and the ability to deploy patches and updates to fielded software with integrity is critical to enterprise network security. Processor manufacturers, compiler developers, execution environment vendors, and operating system vendors have developed execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems. Application software can often take advantage of these mechanisms by using APIs provided by the runtime environment or by enabling the mechanism through compiler or linker options.
O.QUALITY	To ensure quality of implementation, conformant TOEs leverage services and APIs provided by the runtime environment rather than implementing their own versions of these services and APIs. This is especially important for cryptographic services and other complex operations such as file and media parsing. Leveraging this platform behaviour relies upon using only documented and supported APIs.
O.MANAGEMENT	To facilitate management by users and the enterprise, conformant TOEs provide consistent and supported interfaces for their security-relevant configuration and maintenance. This includes the deployment of applications and application updates through the use of platform-supported deployment mechanisms and formats, as well as providing mechanisms for configuration. This also includes providing control to the user regarding disclosure of any PII.

Security Objective	Description
O.PROTECTED_STORAGE	To address the issue of loss of confidentiality of user data in the event of loss of physical control of the storage medium, conformant TOEs will use data-at-rest protection. This involves encrypting data and keys stored by the TOE in order to prevent unauthorized access to this data. This also includes unnecessary network communications whose consequence may be the loss of data.
O.PROTECTED_COMMS	To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant TOEs will use a trusted channel for sensitive data. Sensitive data includes cryptographic keys, passwords, and any other data specific to the application that should not be exposed outside of the application.
O.ENCRYPTION	To prevent data disclosure from decryption, conformant TOEs will transmit and store sensitive data using mechanisms that provide adequate protections.
O.NO_UNATTENDED_TRANSMISSION	To prevent undetected transmissions, conformant TOEs will not transmit unattended voice or video data when streaming media is not in use.
O.TOE_ADMINISTRATION	To support the enforcement of other security functionality, a conformant TOE will provide a management capability that allows for configuration of the TSF.

4.2. Security Objectives for the Operational Environment

Table 7: Security Objectives for the Operational Environment

Security Objective	Description
OE.PLATFORM	The TOE relies upon a trustworthy computing platform for its execution. This includes the underlying operating system and any discrete execution environment provided to the TOE.
OE.PROPER_USER	The user of the application software is not wilfully negligent or hostile and uses the software within compliance of the applied enterprise security policy.

OE.PROPER_ADMIN	The administrator of the application software is not careless, wilfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy.
OE.UPDATE_SOURCE	The operational environment will have TOE software/firmware made available on either the call control server that the TOE connects to or a separate file server managed by the organization.

4.3. Security Objectives Rationale

The security objectives rationale is identical to [AppPP], [MOD_VVoIP] and [TLS-PKG]. It is not repeated herein.

5. SECURITY REQUIREMENTS

This section identifies the Security Functional Requirements for the TOE, including the security requirements rationale (minus the dependency rationale) and the security assurance requirements. The Security Functional Requirements included in this section are derived from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, dated: April 2017, all applicable international interpretations, and [AppPP], [MOD_VVoIP] and [TLS-PKG].

5.1. Conventions

The CC defines operations on Security Functional Requirements: iterations, assignments, selections, assignments within selections and refinements. This document uses the following typographic conventions to identify the operations performed in the ST:

- Iterations following the precise notation used in the source of the relevant SFR, for example FCS_COP.1(1) or FPT_COP.1/SRTP;
- Assignment: Indicated with ***bold italics*** text;
- Selection: Indicated with underlined text;
- Assignment within a Selection: Indicated with ***underlined bold italics*** text.

Formatting conventions for operations performed by the [AppPP], [MOD_VVoIP], or [TLS-PKG] are carried forward without modification.

5.2. TOE Security Functional Requirements

This section identifies the Security Functional Requirements for the TOE. The TOE Security Functional Requirements are summarized in *Table 8* and stated in the following subsections.

Table 8: Security Functional Requirements

Class	SFR	PP/MOD	Required Component
FCS: Cryptographic Support	FCS_CKM_EXT.1	AppPP	Cryptographic Key Generation Services
	FCS_CKM.1/AK	AppPP	Cryptographic Key Generation Services
	FCS_CKM.2	AppPP	Cryptographic Key Establishment
	FCS_RBG_EXT.1	AppPP	Random Bit Generation Services
	FCS_RBG_EXT.2	AppPP	Random Bit Generation from Application
	FCS_STO_EXT.1	AppPP	Storage of Credentials
	FCS_COP.1/Hash	AppPP	Cryptographic Operation - Hashing
	FCS_COP.1/KeyedHash	AppPP	Cryptographic Operation - Keyed-Hash Message Authentication
	FCS_COP.1/Sig	AppPP	Cryptographic Operation - Signing
	FCS_COP.1/SKC	AppPP	Cryptographic Operation - Encryption/Decryption
	FCS_COP.1/SRTP	MOD_VVoIP	Cryptographic Operation (Encryption/Decryption for SRTP)
	FCS_SRTP_EXT.1	MOD_VVoIP	Secure Real-Time Transport Protocol
	FCS_TLS_EXT.1	TLS-PKG	TLS Protocol
	FCS_TLSC_EXT.1	TLS-PKG	TLS Client Protocol
	FCS_TLSC_EXT.2	TLS-PKG	TLS Client Support for Mutual Authentication
	FCS_TLSC_EXT.3	TLS-PKG	TLS Client Support for Signature Algorithms Extension
	FCS_TLSC_EXT.4	TLS-PKG	TLS Client Support for Renegotiation
	FCS_TLSC_EXT.5	TLS-PKG	TLS Client Support for Supported Groups Extension
	FCS_TLSC_EXT.6	TLS_PKG	TLS Client TLS 1.3 resumption refinements
FCO: Communications	FCO_VOC_EXT.1	MOD_VVoIP	Fixed-Rate Vocoder
FDP: User Data Protection	FDP_IFC.1	MOD_VVoIP	Subset Information Flow Control
	FDP_IFF.1	MOD_VVoIP	Simple Security Attributes
	FDP_DEC_EXT.1	AppPP	Access to Platform Resources

Class	SFR	PP/MOD	Required Component
	FDP_NET_EXT.1	AppPP	Network Communications
	FDP_DAR_EXT.1	AppPP	Encryption Of Sensitive Application Data
FIA: Identification and Authentication	FIA_X509_EXT.1	AppPP	X.509 Certificate Validation
	FIA_X509_EXT.2	AppPP	X.509 Certificate Authentication
FMT: Security Management	FMT_MEC_EXT.1	AppPP	Supported Configuration Mechanism
	FMT_CFG_EXT.1	AppPP	Secure by Default Configuration
	FMT_SMF.1	AppPP	Specification of Management Functions
	FMT_SMF.1/VVoIP	MOD_VVoIP	Specification of Management Functions (VVoIP Communications)
FPR: Privacy	FPR_ANO_EXT.1	AppPP	User Consent for Transmission of Personally Identifiable Information
FPT: Protection of the TSF	FPT_API_EXT.1	AppPP	Use of Supported Services and APIs
	FPT_AEX_EXT.1	AppPP	Anti-Exploitation Capabilities
	FPT_IDV_EXT.1	AppPP	Software Identification and Versions.
	FPT_LIB_EXT.1	AppPP	Use of Third Party Libraries
	FPT_TUD_EXT.1	AppPP	Integrity for Installation and Update
	FPT_TUD_EXT.2	AppPP	Integrity for Installation and Update
FTA: TOE Access	FTA_SSL.3/Media	MOD_VVoIP	TSF-Initiated Termination (Media Channel)
FTP: Trusted Path/Channel (FTP)	FTP_DIT_EXT.1	MOD_VVoIP	Protection of Data in Transit
	FTP_ITC.1/Control	MOD_VVoIP	Inter-TSF Trusted Channel (Signaling Channel)
	FTP_ITC.1/Media	MOD_VVoIP	Inter-TSF Trusted Channel (Media Channel)

5.2.1. Cryptographic Support (FCS)

5.2.1.1. FCS_CKM_EXT.1 Cryptographic Key Generation Services

FCS_CKM_EXT.1.1[1] The application shall [

- Implement asymmetric key generation

[1] In accordance with TD0717

].

5.2.1.2. FCS_CKM.1/AK Cryptographic Asymmetric Key Generation

FCS_CKM.1.1/AK[2,3] The **application** shall [

- *implement functionality*

] **to** 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-384 and [P-256, P-521]]** that meet the following: **"[FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.2]"***
- *[FFC schemes] **using ["safe-prime" groups],** that meet the following: **[NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"** and [RFC 7919]],*

].

5.2.1.3. FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1 The application shall [*implement functionality*] to perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [

- ***[Elliptic curve-based key establishment schemes]** that meets the following: **[NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"]***,
- *[**FFC schemes**] **using "safe-prime" groups** that meet the following: **NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [RFC 7919],***

].

5.2.1.4. FCS_RBG_EXT.1 Random Bit Generation Services

FCS_RBG_EXT.1.1 The application shall [

- *implement DRBG functionality*

] for its cryptographic operations.

5.2.1.5. FCS_RBG_EXT.2 Random Bit Generation from Application

FCS_RBG_EXT.2.1 The application shall perform all deterministic random bit generation (DRBG) services in accordance with NIST Special Publication 800-90A using [*CTR_DRBG (AES)*].

[2] In accordance with TD0717
[3] In accordance with TD0945

FCS_RBG_EXT.2.2[4] The deterministic RBG shall be seeded by an entropy source that accumulates entropy from a platform-based DRBG and [

- *no other noise source,*

] 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.

5.2.1.6. FCS_STO_EXT.1 Storage of Credentials

FCS_STO_EXT.1.1[5] The application shall [

- *invoke the functionality provided by the platform to securely store [**database encryption key**],*
- *implement functionality to securely store [**X.509 certificates and associated private keys**] according to [FCS_COP.1/SKC]*

] to non-volatile memory.

5.2.1.7. FCS_COP.1/Hash[6] Cryptographic Operation – Hashing

FCS_COP.1.1/Hash The **application** shall perform [*cryptographic hashing services*] in accordance with a specified cryptographic algorithm [

- *SHA-1,*
- *SHA-256,*
- *SHA-384,*
- *SHA-512*

] and **message digest** sizes [

- *160,*
- *256,*
- *384,*
- *512*

] **bits** that meet the following: [FIPS Pub 180-4].

5.2.1.8. FCS_COP.1/KeyedHash[7] Cryptographic Operation - Keyed-Hash Message Authentication

FCS_COP.1.1/KeyedHash The **application** shall perform [*keyed-hash message authentication*] in accordance with a specified cryptographic algorithm:

- HMAC-SHA-256,
- HMAC-SHA-384,

[4] In accordance with TD0931
[5] In accordance with TD0865 (Selections unaffected)
[6] In accordance with TD0717
[7] In accordance with TD0717

- HMAC-SHA-512

and [

- *HMAC-SHA-1,*

] **with** key sizes [***160 bits, 128-bits, 256-bits, 384-bits, 512-bits***] **and message digest sizes** [*256, 384, 512*] **and** [*160*] **bits** that meet the following: [*FIPS Pub 198-1, 'The Keyed-Hash Message Authentication Code' and FIPS Pub 180-4 'Secure Hash Standard'*].

5.2.1.9. FCS_COP.1/Sig[8] Cryptographic Operation – Signing

FCS_COP.1.1/Sig[9] The **application** shall perform [*cryptographic signature services (generation and verification)*] in accordance with a specified cryptographic algorithm [

- ***RSA schemes** using cryptographic key sizes of [2048-bit or greater] that meet the following: [FIPS PUB 186-4 or FIPS 186-5, "Digital Signature Standard (DSS)", Section 4],*
- ***ECDSA schemes** using ["NIST curves" P-256, P-384 and [P-521]] that meet the following: [FIPS PUB 186-4 or FIPS 186-5, "Digital Signature Standard (DSS)", Section 5]*

].

5.2.1.10. FCS_COP.1/SKC[10] Cryptographic Operation - Encryption/Decryption

FCS_COP.1.1/SKC The **application** shall perform [*encryption/decryption*] in accordance with a specified cryptographic algorithm [

- *AES-GCM (as defined in NIST SP 800-38D) mode,*
- *AES-CTR (as defined in NIST SP 800-38A) mode,*

] and cryptographic key sizes [*128-bit, 256-bit*].

5.2.1.11. FCS_COP.1/SRTP Cryptographic Operation (Encryption/Decryption for SRTP)

FCS_COP.1.1/SRTP: The TSF shall perform [encryption/decryption to support SDES-SRTP] in accordance with a specified cryptographic algorithm [*AES-CTR (as defined in NIST SP 800-38A), AES-GCM (as defined in NIST SP 800-38D)*] and cryptographic key sizes [*128-bit, 256-bit*].

5.2.1.12. FCS_SRTP_EXT.1 Secure Real-Time Transport Protocol (MOD_VVoIP)

FCS_SRTP_EXT.1.1 The TSF 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 TSF shall implement SDES-SRTP supporting the following ciphersuites [

- *AES_CM_128_HMAC_SHA1_80, in accordance with RFC 4568,*

[8] In accordance with TD0717
[9] In accordance with TD0945
[10] In accordance with TD0717

- *AES_CM_128_HMAC_SHA1_32, in accordance with RFC 4568,*
- *AES_256_CM_HMAC_SHA1_80, in accordance with RFC 6188,*
- *AES_256_CM_HMAC_SHA1_32, in accordance with RFC 6188,*
- *AEAD_AES_128_GCM, in accordance with RFC7714,*
- *AEAD_AES_256_GCM, in accordance with RFC 7714*].

FCS_SRTP_EXT.1.3 The TSF shall ensure the SRTP NULL algorithm can be disabled.

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

5.2.1.13. FCS_TLS_EXT.1 TLS Protocol (PKG_TLS)

FCS_TLS_EXT.1.1 The TSF shall implement [

- *TLS as a client,*

].

5.2.1.14. FCS_TLSC_EXT.1 TLS Client Protocol (PKG-TLS)

FCS_TLSC_EXT.1.1 The TSF shall implement TLS 1.2 (RFC 5246) and [*TLS 1.3 (RFC 8446)*] as a client that supports additional functionality for session renegotiation protection and [

- Mutual authentication
- Supplemental downgrade protection
- Session resumption

] and shall abort attempts by a server to negotiate all other TLS or SSL versions.

FCS_TLSC_EXT.1.2 The TSF shall be able to support the following TLS 1.2 ciphersuites: [

- ***TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 and RFC 8422***
- ***TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 and RFC 8422***
- ***TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289***
- ***TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288***

], the following PP-specific ciphersuites using pre-shared secrets:

- ***no ciphersuites using pre-shared secrets***

], and the following TLS 1.3 ciphersuites: [

- ***TLS_AES_256_GCM_SHA384 as defined in RFC 8446***
- ***TLS_AES_128_GCM_SHA256 as defined in RFC 8446***

] offering the supported ciphersuites in a client hello message in preference order: [

- ***TLS_AES_256_GCM_SHA384 as defined in RFC 8446***
- ***TLS_AES_128_GCM_SHA256 as defined in RFC 8446***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 and RFC 8422***
- ***TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 and RFC 8422***
- ***TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288***

- ***TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289***
- ***TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288***

].

FCS_TLSC_EXT.1.3 The TSF shall not offer ciphersuites indicating the following:

- the null encryption component
- support for anonymous servers
- use of deprecated or export-grade cryptography including DES, 3DES, RC2, RC4, or IDEA for encryption
- use of MD

and shall abort sessions where a server attempts to negotiate ciphersuites not enumerated in the client hello message.

FCS_TLSC_EXT.1.4 The TSF shall be able to support the following TLS client hello message extensions:

- signature_algorithms extension (RFC 8446) indicating support for [
  - *ecdsa-secp384r1_sha384 (RFC 8446)*
  - *rsa_pkcs1_sha384 (RFC 8446)*

  ], and [
  - *rsa_pss_rsae_sha384 (RFC 8603)*

  [
  - ***ecdsa-secp256r1_sha256 (RFC 8446)***
  - ***ecdsa-secp521r1_sha512 (RFC 8446)***
  - ***rsa_pkcs1_sha256 (RFC 8446)***
  - ***rsa_pkcs1_sha512 (RFC 8446)***
  - ***rsa_pss_rsae_sha256 (RFC 8446)***
  - ***rsa_pss_rsae_sha512 (RFC 8446)***

  ]

  ]
- extended_master_secret extension (RFC 7627) enforcing server support
- the following other extensions: [
  - *supported_versions extension (RFC 8446) indicating support for TLS 1.3*
  - *supported_groups extension (RFC 7919, RFC 8446) indicating support for [*
    - *secp256r1*
    - *secp384r1*
    - *secp521r1*

    ]
  - *key_share extension (RFC 8446)*

  ] and shall not send the following extensions:
  - early_data
  - psk_key_exchange_mode indicating PSK only mode.

FCS_TLSC_EXT.1.5 The TSF shall be able to [

- *verify that a presented identifier of name type: [*
  - *DNS name type according to RFC 6125*
  - *Common Name conversion to DNS name according to RFC 6125*

]

] matches a reference identifier of the requested TLS server and shall abort the session if no match is found.

FCS_TLSC_EXT.1.6 The TSF shall not establish a trusted channel if the server certificate is invalid [*with no exceptions*].

5.2.1.15. FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication (PKG-TLS)

FCS_TLSC_EXT.2.1 The TSF shall support mutual authentication using X.509v3 certificates during the handshake and [*at no other time*], in accordance with [*RFC 5246, Section 7.4.4, RFC 8446, section 4.3.2*].

5.2.1.16. FCS_TLSC_EXT.3 TLS Client Downgrade Protection (PKG-TLS)

FCS_TLSC_EXT.3.1 The TSF shall not establish a TLS channel if the server hello message includes [*TLS 1.2 downgrade indicator, TLS 1.1 or below downgrade indicator*] in the server random field.

5.2.1.17. FCS_TLSC_EXT.4 TLS Client Support for Renegotiation (PKG-TLS)

FCS_TLSC_EXT.4.1 The TSF shall support secure renegotiation through use of [*the "renegotiation info" TLS extension*] in accordance with RFC 5746 and shall terminate the session if an unexpected server hello is received or [*hello request message is received*].

5.2.1.18. FCS_TLSC_EXT.5 TLS Client Support for Session Resumption (PKG-TLS)

FCS_TLSC_EXT.5.1 The TSF shall support session resumption as a client via the use of [*session ID in accordance with RFC 5246, tickets in accordance with RFC 5077, PSK and tickets in accordance with RFC 8446*].

5.2.1.19. FCS_TLSC_EXT.6 TLS Client TLS 1.3 Resumption Refinements (PKG-TLS)

FCS_TLSC_EXT.6.1 The TSF shall send a psk_key_exchange_mode extension with the value psk_dhe_ke when TLS 1.3 session resumption is offered.

FCS_TLSC_EXT.6.2 The TSF shall not send early data in TLS 1.3 sessions.

5.2.2. Communications (FCO)

5.2.2.1. FCO_VOC_EXT.1 Fixed-Rate Vocoder (MOD_VVoIP)

FCO_VOC_EXT.1.1 The TSF shall transmit voice media using a constant bit rate vocoder.

5.2.3. User Data Protection (FDP)

5.2.3.1. FDP_IFC.1 Subset Information Flow Control (MOD_VVoIP)

FDP_IFC.1.1 The TSF shall enforce the [*media transmission policy*] on [*voice/video media transmitted by the TOE*].

5.2.3.2. FDP_IFF.1 Simple Security Attributes (MOD_VVoIP)

FDP_IFF.1.1 The TSF shall enforce the [*media transmission policy*] based on the following types of subject and information security attributes: [*TOE hook state, VVoIP call connection status, and VVoIP call control server status*].

FDP_IFF.1.2[11] The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: [

- *The TOE is [registered with a VVoIP call control server],*
- *A call has been established with a telephony device (VVoIP endpoint),*
- *The TOE is in the off-hook state,*
- *The TOE is not in the mute state,*
- *[No other rules]*].

FDP_IFF.1.3 The TSF shall enforce [*no additional information flow control policy rules*].

FDP_IFF.1.4 The TSF shall explicitly authorize an information flow based on the following rules: [*no additional rules*].

FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: [*all TCP and UDP ports used by the TOE are closed when not in active use*].

5.2.3.3. FDP_DEC_EXT.1 Access to Platform Resources

FDP_DEC_EXT.1.1 The application shall restrict its access to [

- *network connectivity,*
- *camera,*
- *microphone,*
- *Bluetooth,*
- *Biometric sensor*
- *Vibration motor*

].

FDP_DEC_EXT.1.2 The application shall restrict its access to [

- *address book,*
- ***[Android file system]***

].

[11] In accordance with TD0718

5.2.3.4. FDP_NET_EXT.1 Network Communications

FDP_NET_EXT.1.1 The application shall restrict network communication to [

- *user-initiated communication for [**connecting to a SIP server, connecting to a VVoIP endpoint, checking for updates**],*
- *[**certificate validation with OCSP, fetch timeout configuration from the configuration server**]*

].

5.2.3.5. FDP_DAR_EXT.1 Encryption Of Sensitive Application Data

FDP_DAR_EXT.1.1 The application shall [

- *protect sensitive data in accordance with FCS_STO_EXT.1*

] in non-volatile memory.

5.2.4. Identification and Authentication (FIA)

5.2.4.1. FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1.1 The application shall [*implement functionality*] to 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 application 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, and that any path constraints are met.
- The application shall validate that any CA certificate includes caSigning purpose in the key usage field.
- The application shall validate the revocation status of the certificate using *[OCSP as specified in RFC 6960*].
- The application shall validate the extendedKeyUsage (EKU) 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 EKU 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 EKU field.
    - S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the EKU 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 EKU field.

- Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field.

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

5.2.4.2. FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1 The application shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [*TLS*].

FIA_X509_EXT.2.2 When the application cannot establish a connection to determine the validity of a certificate, the application shall [*not accept the certificate*].

5.2.5. Security Management (FMT)

5.2.5.1. FMT_MEC_EXT.1 Supported Configuration Mechanism

FMT_MEC_EXT.1.1 The application shall [

- *invoke the mechanisms recommended by the platform vendor for storing and setting configuration options*][12].

5.2.5.2. FMT_CFG_EXT.1 Secure by Default Configuration

FMT_CFG_EXT.1.1 The application shall provide only enough functionality to set new credentials when configured with default credentials or no credentials.

FMT_CFG_EXT.1.2 The application shall be configured by default with file permissions which protect the application binaries and data files from modification by normal unprivileged users.

5.2.5.3. FMT_SMF.1 Specification of Management Functions

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

- [***Register and unregister with a SIP server,***
- ***Load X.509 Certificates and private keys,***
- ***Select X.509 Certificate,***
- ***Check for software updates,***
- ***Specify ESC server URL,***
- ***Select TLS cipher suites,***
- ***Enable/disable call history***]

].

[12] In accordance with TD0437

5.2.5.4. FMT_SMF.1/VVoIP Specification of Management Functions (VVoIP Communications) (MOD_VVoIP)

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

- Ability to *[register the TOE to an ESC [manually]];*

  [
  - *Ability to configure the termination period for idle calls;*
  - *Ability to specify the vocoder used;*

].

5.2.6. Privacy (FPR)

5.2.6.1. FPR_ANO_EXT.1 User Consent for Transmission of Personally Identifiable Information

FPR_ANO_EXT.1.1 The application shall [*not transmit PII over a network*].

5.2.7. Protection of the TSF (FPT)

5.2.7.1. FPT_API_EXT.1 Use of Supported Services and APIs

FPT_API_EXT.1.1 The application shall use only documented platform APIs.

5.2.7.2. FPT_AEX_EXT.1 Anti-Exploitation Capabilities

FPT_AEX_EXT.1.1 The application shall not request to map memory at an explicit address except for [***no exceptions***].

FPT_AEX_EXT.1.2 The application shall [

- *not allocate any memory region with both write and execute permissions*

].

FPT_AEX_EXT.1.3 The application shall be compatible with security features provided by the platform vendor.

FPT_AEX_EXT.1.4 The application shall not write user-modifiable files to directories that contain executable files unless explicitly directed by the user to do so.

FPT_AEX_EXT.1.5 The application shall be built with stack-based buffer overflow protection enabled.

5.2.7.3. FPT_IDV_EXT.1 Software Identification and Versions

FPT_IDV_EXT.1.1 The application shall be versioned with [***[a visible version string accessible in the Settings menu]***].

5.2.7.4. FPT_LIB_EXT.1 Use of Third Party Libraries

FPT_LIB_EXT.1.1 The application shall be packaged with only **[**

- ***libcrashlytics-common.so (see TSS)***
- ***libcrashlytics-handler.so (see TSS)***
- ***libcrashlytics-trampoline.so (see TSS)***
- ***libcrashlytics.so (see TSS)***
- ***libfips.so***
- ***libimage_processing_util_jni.so***
- ***libjnidispatch.so***
- ***libmessaging_vphone.so***
- ***libsqlcipher.so***

].

5.2.7.5. FPT_TUD_EXT.1 Integrity for Installation and Update

FPT_TUD_EXT.1.1 The application shall [*provide the ability*] to check for updates and patches to the application software.

FPT_TUD_EXT.1.2 The application shall [*provide the ability*] to query the current version of the application software.

FPT_TUD_EXT.1.3 The application shall not download, modify, replace or update its own binary code.

FPT_TUD_EXT.1.4 Application updates shall be digitally signed such that the application platform.

can cryptographically verify them prior to installation.

FPT_TUD_EXT.1.5 The application is distributed [*as an additional software package to the platform OS*].

5.2.7.6. FPT_TUD_EXT.2 Integrity for Installation and Update

FPT_TUD_EXT.2.1 The application shall be distributed using [the format of the platform-supported package manager].

FPT_TUD_EXT.2.2 The application shall be packaged such that its removal results in the deletion of all traces of the application, with the exception of configuration settings, output files, and audit/log events.

FPT_TUD_EXT.2.3 The application installation package shall be digitally signed such that its platform can cryptographically verify them prior to installation.

5.2.8. TOE Access (FTA)

5.2.8.1. FTA_SSL.3/Media TSF-Initiated Termination (Media Channel)

FTA_SSL.3.1/Media The TSF shall terminate **voice/video transmission** after [*inactivity longer than* [***[60]*** *seconds, an administrator configurable interval]*].

5.2.9. Trusted Path/Channel (FTP)

5.2.9.1. FTP_DIT_EXT.1[13] Protection of Data in Transit (PPApp+MOD_VoIP)

FTP_DIT_EXT.1.1[14] The application shall [

- encrypt all transmitted [data] with [**TLS as a client as defined in the Functional Package for TLS**] and /***Secure Real-Time Transport Protocol (SRTP)***/

] between itself and another trusted IT product.

5.2.9.2. FTP_ITC.1/Control Inter-TSF Trusted Channel (Signaling Channel) (MOD_VVoIP)

FTP_ITC.1.1/Control[15] The TSF shall **be capable of using [*Session Initiation Protocol (SIP)*] to** provide a **trusted** communication channel between itself and **a VVoIP call control 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/Control[16] The TSF shall permit [the TSF, **the VVoIP call control server**] to initiate communication via the trusted channel.

FTP_ITC.1.3/Control The TSF shall initiate communication via the trusted channel for [*establishment of call control*].

5.2.9.3. FTP_ITC.1/Media Inter-TSF Trusted Channel (Media Channel) (MOD_VVoIP)

FTP_ITC.1.1/Media[17] The TSF shall **be capable of using [*SRTP*] to** provide a **trusted** communication channel between itself and **another VVoIP endpoint or other telephony device** 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/Media The TSF shall permit **[*the TSF, another VVoIP endpoint or other telephony device*]** to initiate communication via the trusted channel.

FTP_ITC.1.3/Media The TSF shall initiate communication via the trusted channel for [*transmission of voice/video media*].

5.3. Security Assurance Requirements

The Security Assurance Requirements applicable to the TOE are those stated in Sect. 5.2 of [AppPP] and in Sect. 2.2 of [CFG_APP-VVoIP_V1.1]. Neither [MOD_VVoIP] nor [TLS-PKG]

[13] In accordance with TD0743
[14] In accordance with TD0785
[15] In accordance with TD0718
[16] In accordance with TD0718
[17] In accordance with TD0718

defines additional Security Assurance Requirements. For the sake of compactness, the Security Assurance Requirements are not repeated herein.

5.4. Security Requirements Rationale

The security requirements rationale is identical to [AppPP], [MOD_VVoIP] and [TLS-PKG]. It is not repeated herein.

6. TOE SUMMARY SPECIFICATION

This chapter identifies and describes how the Security Functional Requirements are met by the TOE. CAVP Certificate references for all cryptographic algorithms are given in Table 2.

Table 9: TOE Summary Specification Description

Requirement	Rationale
FCS_CKM_EXT.1 FCS_CKM.1/AK FCS_CKM.2	The TOE generates asymmetric keys for digital signature computation and X.509 certificate-based authentication using the following algorithms: • RSA schemes in accordance with FIPS PUB 186-4 using key sizes of 2048-bit or greater • ECC schemes for digital signatures in accordance with FIPS PUB 186-5 using NIST curves P-256, P-384, and P-521 The TOE implements the following asymmetric algorithms for use with TLS Key establishment: • Elliptic curve-based key establishment schemes in accordance with NIST SP800-56Ar3 • FFC schemes using “safe-prime” groups in accordance with NIST SP800-56Ar3 and RFC 7919
FCS_RBG_EXT.1	All random bit generation in the TOE provided by DRBG (CAVP #A7364), seeded by the TOE using the Java SecureRandom class. The random bit generation is used by the following functions of the TOE: • Generation of asymmetric and symmetric cryptographic keys • Generation of Initialization Vectors for symmetric cryptography • Generation of TLS RSA PreMasterSecret.
FCS_RBG_EXT.2	The TOE DRBG (CAVP #A7364) is a NIST SP800-90A compliant CTR_DRBG(AES) with derivation function, which is seeded with 256 bits of entropy via the Java SecureRandom class.
FCS_STO_EXT.1	The user credentials and other sensitive information is stored in an encrypted database (SQLCipher). The following data is stored in the database: • User profile • User keys and certificates • User contacts The TOE invokes the Android Keystore API to store the AES 256-bit database encryption key. The database is encrypted according to FCS_COP.1/SKC. MODE_PRIVATE flag is also set on the database file.

Requirement	Rationale
FCS_COP.1/Hash	The TOE implements the following hashing algorithms: SHA-1, SHA2-256, SHA2-384, SHA2-512.
FCS_COP.1/KeyedHash	The TOE implements HMAC message authentication (CAVP #A7364). HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384 and HMAC_SHA-512 are supported.
FCS_COP.1/Sig	The TOE implements RSA and ECDSA signature generation and verification (CAVP #A7364). RSA keys of 2048 bits and 4096 bits are supported. ECDSA keys using P-256, P-384 and P-521 are supported.
FCS_COP.1/SKC	The TOE implements AES for encryption and decryption on TLS and SRTP links. For TLS the TOE implements GCM modes. For SRTP, the TOE implements CM, and GCM modes, all using 128-bit and 256-bit keys. All cryptographic functions are NIST-validated (CAVP #A7364).
FCS_COP.1/SRTP FCS_SRTP_EXT.1	The TOE implements the Secure Real-Time Transport Protocol (SRTP) in the libSRTP v.1.5.4 library. libSRTP is compatible with SRTP (RFC 3711) and SRTP SDES (RFC 4568). libSRTP uses our NIST-validated cryptography (CAVP #A7364). The TOE supports the following SRTP ciphersuites: • AES_CM_128_HMAC_SHA1_80 (RFC 4568) • AES_CM_128_HMAC_SHA1_32 (RFC 4568) • AES_256_CM_HMAC_SHA1_80 (RFC 6188) • AES_256_CM_HMAC_SHA1_32 (RFC 6188) • AEAD_AES_128_GCM (RFC 7714) • AEAD_AES_256_GCM (RFC 7714) The TOE establishes SRTP sessions (for both incoming and outgoing calls) using SIP, described in FTP_ITC.1/Control. The SRTP keying material and ciphersuites are negotiated using SDES (SDP attachment to a SIP message). The ciphersuites listed above are fixed and not configurable. These are the only ciphersuites available to the TOE. Therefore, the TOE cannot use the NULL ciphersuite or any other ciphersuite not listed above. SRTP ports can be changed in the app settings under the direction of an Authorized Administrator.

FCS_TLS_EXT.1 FCS_TLSC_EXT.1 FCS_TLSC_EXT.2 FCS_TLSC_EXT.3 FCS_TLSC_EXT.4 FCS_TLSC_EXT.5 FCS_TLSC_EXT.6	The TOE implements TLS v1.2 (RFC 5246) and TLS v1.3 (RFC 8446). With mutual authentication, downgrade protection, and session resumption. The TOE supports the following TLS v1.2 and 1.3 ciphersuites: • TLS_AES_256_GCM_SHA384 (RFC 8446) • TLS_AES_128_GCM_SHA256 (RFC 8446) • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (RFC 5289 & RFC 8422) • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (RFC 5289 & RFC 8422) • TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (RFC 5288) • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (RFC 5289) • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (RFC 5289) • TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (RFC 5288) Note that the ciphersuites are listed in the same order as listed in the TLS Hello message The ciphersuites listed above are fixed and not configurable. These are the only ciphersuites available to the TOE. Therefore, the TOE cannot use the NULL ciphersuite or any other ciphersuite not listed above.. The TOE sends a client certificate to the SIP server when requested as part of the TLS handshake. The SIP Server validates the certificate to establish the authenticity of the client. The SIP server also matches a reference identifier by parsing the TOE client certificate X.509 Common Name (CN). The reference identifier is matched against the SIP server's configured reference identifier and the SIP server terminates the session if a match is not found. The TOE verifies the SIP server certificate's signature using the configured root certificate. The TOE establishes a reference identifier by parsing the DNS Name for the connected SIP server. The reference identifier is matched against the SIP server certificate SAN, if present. If the SAN is not present, the referenced identifier is matched against the CN and the TLS session is terminated if a match is not found. The TOE will not establish a connection if the server hello message includes downgrade indicators in the server random field. This is the default response and is not configurable. With TLS 1.3 the TOE sends a psk_key_exchange_mode extension with the value psk_dhe_ke when session resumption is offered (see extensions below). The TOE does not support "early data" (RFC8446) in TLS 1.3 sessions. Certificate pinning is supported. The SIP server certificate is matched based on its public key. The pinned public key can be updated after a successful sign-in with user password. The TOE presents the following extensions in the Hello message: • Renegotiation Info – Renegotiation based on session ID. Renegotiation support is provided by default and is not configurable. • Elliptic Curves Extension – Showing the P-256, P-384 and P-521 curves. This is the default TOE behaviour and cannot be modified.

Requirement	Rationale
	• Signature Algorithms extension – Indicating support for SHA-256, SHA-384 signature hashes. This is the default TOE behaviour and cannot be modified. • Key Share extension – Lists supported keys. • Supported Versions – Supported versions are 1.2 and 1.3 • Pre-Shared Key Exchange – Supported mode is psk_dhe_ke when session resumption is offered. Resumption support is provided by default and is not configurable.
FCO_VOC_EXT.1	The TOE uses the Opus, G.711(PCMA) and G.711(PCMU) vocoders to transmit voice media. The Opus vocoder generates a Constant Bit-Rate (CBR) stream of 48 Kbps and the G.711 codecs generate a CBR stream of 8 Kbps. In low-bandwidth mode the TOE uses Opus at 12 Kbps (CBR).
FDP_IFC.1 FDP_IFF.1	The TOE transmits media data when it is registered with the ESC and a call has been established between the TOE and a remote telephony device. The TOE does not transmit any media data when it is not on a call or when it is muted. On a video call there is a separate “mute” button for video. The TOE does not implement a “hold” state.
FDP_DEC_EXT.1	The TOE accesses network connectivity, camera, microphone and Bluetooth resources. Bluetooth is an optional setting for hands-free audio. The TOE also accesses its database through the Android file system and can read from the Android filesystem for importing the root certificate and user’s .p8 credentials file. Here is the full list of permissions requested by the app: • ACCESS_NETWORK_STATE • ACCESS_WIFI_STATE • BLUETOOTH (android:maxSdkVersion="30") • BLUETOOTH_CONNECT (usesPermissionFlags="neverForLocation" tools:targetApi="s") • BROADCAST_STICKY • INTERNET • MODIFY_AUDIO_SETTINGS • RECEIVE_BOOT_COMPLETED • VIBRATE • WAKE_LOCK • FOREGROUND_SERVICE • USE_BIOMETRIC • USE_FULL_SCREEN_INTENT • READ_PHONE_STATE • CAMERA • RECORD_AUDIO • READ_EXTERNAL_STORAGE • WRITE_EXTERNAL_STORAGE • RECEIVE • REQUEST_IGNORE_BATTERY_OPTIMIZATIONS • C2D_MESSAGE (for Android push notifications) • READ_CONTACTS

Requirement	Rationale
FDP_NET_EXT.1	The TOE performs the following user-initiated network communications: • Communicating with a SIP server • Communicating with a VVoIP endpoint • Communicating with an Update check server (on the ESC) The TOE automatically initiates the following network communications: • OCSP certificate validation
FDP_DAR_EXT.1	Sensitive data consisting of user credentials and contact information are stored in an encrypted database according to FCS_COP.1/SKC with keys stored according to FCS_STO_EXT.1.
FIA_X509_EXT.1 FIA_X509_EXT.2	The TSF uses X.509v3 certificates to authenticate the user to the SIP server via a mutually authenticated TLS connection. Only one client-side certificate is supported and is sent by the client during the TLS handshake. The TOE performs validity checks on the CA path and confirms that either the SubjectAltName or CN match what was provided on the distant connection certificate. The TSF also performsan OCSP validity check on the server certificate. Connection attempts are made only if the certificate is deemed valid. The TSF performs checks for RFC 5280 validation, validates certificate path by ensuring the basicConstraints extension is present and the CA flag is set to True for all CA certificates. The TOE also verifies the path terminates with a trust anchor that was manually imported into the TOE. The TSF validates the extendedKeyUsage field according to the following rules: • Server certificates presented for TLS shall have the TLS Web Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field • In the case where it is not possible to check the validity of the Certificate via an online check the TOE does not establish a TLS connection except when override is authorized in the case where valid revocation information is not available. Override is configured during registration with the Cellcrypt Server.
FMT_MEC_EXT.1[18]	The TOE stores its security-related configuration settings in an encrypted SQLCipher database stored at /data/data/com.cellcrypt.federal/databases/[identifier] where identifier is based on the user ID. The encrypted database key is generated and stored using the Android Keystore system using the Android Keystore provider (KeyStore.getInstance("AndroidKeyStore")) referenced here: https://developer.android.com/privacy-and-security/keystore. Other general settings are stored in Android “SharedPreferences” referenced here: https://developer.android.com/reference/android/content/SharedPreferences.

[18] In accordance with TD0747

Requirement	Rationale
FMT_CFG_EXT.1	The TOE does not contain any default credentials when it is installed. TOE credentials (certificates and keys) must be configured before the TOE can connect to the ESC. User credentials (username and password) must be registered with the ESC before the user can make secure calls. Users can also enable and set an app PIN via the app settings to prevent unauthorised access to the app itself.
FMT_SMF.1	The TOE has a GUI which allows users to perform the following management functions in its settings: • Register/unregister with a SIP server (“Online” slider switch) • Specify the ESC server URL when signing in • Select and Load X.509 Certificates and private keys from the file system • Check for software updates • Select the TLS cipher-suites from a list (check boxes) • Enable/disable call history (slider switch) These settings are stored in the encrypted SQLCipher database as described in FMT_MEC_EXT.1.
FMT_SMF.1/VVoIP	The TOE provides the following VVoIP management settings: • Manually register and unregister to the ESC (slider control switch). Default is register to the ESC. • Configure the termination period for idle calls (text edit box control for seconds). Default is 30s. • Specify the vocoder from a list of supported vocoders (radio buttons). Default is all codecs enabled for offer.
FPR_ANO_EXT.1	The TOE does not transmit PII.
FPT_API_EXT.1	The TOE uses the following Android APIs: android, android.annotation, android.app, android.bluetooth, android.content, android.content.res, android.database, android.database.sqlite, android.graphics, android.hardware, android.media, android.media.audiofx, android.net, android.os, android.preference, android.provider, android.security, android.telephony, android.test, android.text, android.text.method, android.util, android.view, android.view.inputmethod, android.widget, java.io, java.lang, java.lang.annotation, java.lang.ref, java.lang.reflect, java.math, java.net, java.nio, java.nio.channels, java.nio.charset, java.security, java.security.cert, java.security.interfaces, java.text, java.util, java.util.concurrent, java.util.concurrent.atomic, java.util.concurrent.atomic.locks, java.util.regex and javax.crypto, javax.crypto.spec.

Requirement	Rationale
FPT_AEX_EXT.1[19]	The app's encrypted database key is stored in the Android Keystore which makes use of the hardware security element. All sensitive information is stored within the app-protected space and can only be exported to the Download folder if the User selects the app's Export option. The TOE is built with the -fstack-protector-all compiler flag which enables stack-based buffer overflow protection. ASLR is enabled using the -fPIC compiler flag.
FPT_IDV_EXT.1	The TOE version information is visible through the version string accessible in the Settings menu. The TOE version may also include the release info concatenated and separated by full stops, i.e. as in 5.0.0.
FPT_LIB_EXT.1	The TOE is packaged with following libraries: Note: The Firebase libraries are included in the common code base, but Crashlytics is disabled. The Firebase libraries are only used to fetch a Push token from Google. Google's Push service is required to wake up the app when it is out of memory and an incoming call is pending. The token is sent in encrypted form to the Cellcrypt server which uses it to request Push from Google when an incoming call is destined for that user. No user identifiable information is contained in the Push request. • libcrashlytics-common.so • libcrashlytics-handler.so • libcrashlytics-trampoline.so • libcrashlytics.so • libfips.so – OpenSSL FIPS 140-3 module library. • libimage_processing_util_jni.so – Incuded with Android JetPack CameraX library. • libjnidispatch.so - Android native library used by the Java Native Access (JNA) library to facilitate communication between Java code and native code. • libmessaging_vphone.so – Cellcrypt library providing communications and security. Consists of openssl, mp3lame, pjsip, srtp and rust libraries – tokio, serde, request, rayon, hyper, thiserror. • libsqlcipher.so – Database library from Zetetic

[19] In accordance with TD0798 and TD0815

Requirement	Rationale
FPT_TUD_EXT.1 FPT_TUD_EXT.2	By accessing a setting on the TOE, the TOE user can check for updates by querying the Update query server, which returns the most recent version number. The TOE only indicates an update is available when the returned version is greater than the current TOE version. The TOE allows the user to query the TOE's current version in the user settings. The TOE is distributed as an additional package to the OS. TOE updates are initially provided by a Cellcrypt App Store uniquely accessible to each customer organisation and then typically delivered to user devices via a Mobile Device Management (MDM) system. Updates completely replace the existing TOE application except for the TOE configuration settings and the TOE database. Updates to the TOE are distributed in the Android Package (APK) format (.apk). Cellcrypt is the authorized source of TOE updates. All TOE APK files are signed with the Cellcrypt private key. Upon initial installation, Android trusts the associated public key. Android uses the associated public key to verify the authenticity of all subsequent updates to the TOE software.
FTA_SSL.3/Media	The TOE terminates idle voice connections. The TOE considers voice connection idle when the TOE is not receiving data from the peer. The idle time is 30 seconds by default. The Administrator can update the idle time through a configuration file the TOE downloads from the configuration server.
FTP_DIT_EXT.1	The TOE encrypts all data using TLS or SRTP. TLS is used to encrypt the control channel described by FTP_ITC.1/Control while SRTP is used to encrypt the media channel described by FTP_ITC.1/Media.
FTP_ITC.1/Control FTP_ITC.1/Media	The TOE implements the Session Initiation Protocol (SIP) in the PJSIP library. The PJSIP library uses our NIST-validated cryptography (CAVP #A7364). The TOE uses TLS v1.2 and TLS v1.3 (FCS_TLSC_EXT) to protect the SIP communications. The media channel connecting two or more peers is protected by SRTP according to RFC 3711 with key establishment using SRTP SDES according to RFC 4568. The application makes use of libSRTP (ciphersuites described in which in turn using our NIST-validated cryptography (CAVP #A7364) for all cryptographic operations including confidentiality and integrity based on the ciphersuites listed under FCS_SRTP_EXT.1.

Requirement	Rationale
ALC_TSU_EXT.1	The developer's process for providing timely security updates involves accepting reports about potential vulnerabilities on their webpage at https://www.cellcrypt.com/threats. The use of https protects the reports from unauthorized disclosure. Upon receipt of a report, the developer identifies remedial action. Once the remedial action has been implemented, the TOE undergoes normal production testing before being released. There are no partial updates, any update shall include the entire TOE. Each customer identified security officer is notified via email when a security update is available. The updates are distributed in Android Package (APK) format (.apk). TOE APK files are digitally signed with the Cellcrypt private key. The signature is verified by the Android installation routine. The time between disclosure of a vulnerability and availability of a security update varies from two weeks to 90 days. Cellcrypt gets weekly OS updates/vulnerability notifications via OS subscriptions, and our customers are notified of the same via their own subscriptions. Support contracts can also be arranged allowing Cellcrypt to notify customers directly via email. Third-party libraries are treated in the same way and unlicenced open-source third-party library vulnerabilities are monitored via MITRE CVE and NIST NCD feeds.

APPENDIX A – TERMINOLOGY AND ACRONYMS

Table 10: Terminology and Acronyms

Term	Description
AES	Advanced Encryption Standard
AIA	Authority Information Access
ACVP	Automated Cryptographic Validation Protocol
API	Application Programming Interface
APK	Android Package
ASLR	Address space layout randomization (ASLR)
CA	Certificate Authority
CAVP	Cryptographic Algorithm Validation Program
CC	Common Criteria for Information Technology Security Evaluation
CN	Common Name
CTR	Counter mode
DRBG	Deterministic Random Bit Generator
ECDSA	Elliptic Curve Digital Signature Algorithm
EKU	Extended Key Usage
ESC	Enterprise Session Controller
FIPS	Federal Information Processing Standard
GUI	Graphical User Interface
GCM	Galois Cipher Mode
HMAC	Hash-based Message Authentication Code
IP	Internet Protocol
IV	Initial Vector
KEK	Key Encryption Key
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards and Technology
OID	Object Identifier
OS	Operating System
PAA	Processor Algorithm Accelerators
PII	Personal Identification Information
PP	NIAP Protection Profiles
PRNG	Pseudo Random Number Generator
RNG	Random Number Generator

RSA	Rivest Shamir Adleman
RTP	Real Time Protocol (RFC 3550)
SAN	Subject Alternative Name
SDES	SDP Security Descriptions for Media Streams (RFC 4568)
SDP	Session Description Protocol (RFC 4566)
SIP	Session Initiation Protocol (RFC 3261)
SRTP	Secure Real Time Protocol (RFC 3711)
TOE	Target Of Evaluation
TSF	TOE Security Functions
TSS	TOE SFR Summary
VPN	Virtual Private Network
VoIP	Voice over Internet Protocol
VVoIP	Video and Voice over Internet Protocol