COMSec Admin+ Client Security Target Lite

Proyecto / *Project*: COMSec Admin+ Evaluation

Programa / *Programme*: COMSec Admin+

Expediente/Contrato / *Contract*: N/A

Subtitulo / *Subtitle*: COMSec Admin+ Client Security Target Lite

	INDRA			
	Nombre *Name*	Firma *Signature*	Fecha *Date*	Cargo *Responsibility*
Preparado *Prepared*	Juan Luis López			System Engineer
Revisado *Revised*	David Domingo			Technical Responsible
Aprobado *Approved*	Francisco Sánchez			Quality Manager
Autorizado *Authorized*	Francisco Sánchez			Project Manager

REGISTRO DE EDICIONES Y REVISIONES DE PÁGINAS
Record of editions and page revisions

This document contains the following pages, in the editions and revisions indicated:

Capítulo *Chapter*	Edic./Rev. *Edit./Rev..*	Capítulo *Chapter*	Edic./Rev. *Edit./Rev.*	Capítulo *Chapter*	Edic./Rev. *Edit./Rev.*	Capítulo *Chapter*	Edic./Rev. *Edit./Rev..*
All	A/0						

REGISTRO DE CAMBIOS EN EL DOCUMENTO
Document changes record

Edic./Rev. *Edit./Rev.*	Fecha *Date*	Capítulos *Chapters*	Razón del Cambio *Reason for change*
A/0	03/09/2018	All	Initial Release Version

HOJA DE DISTRIBUCIÓN
Distribution sheet

Nº Copia *Copy no.*	Empresa / Organismo *Company / Organization*	Departamento *Department*	Nombre y Apellidos *Name and surname*

ÍNDICE GENERAL
Table of contents

ÍNDICE GENERAL
Table of contents

ÍNDICE FIGURAS
Figures index

ÍNDICE TABLAS
Tables index

1 SECURITY TARGET INTRODUCTION

This section identifies the Security Target (ST) and the Target of Evaluation (TOE) identification, including the document organization, ST conformance claims, and ST conventions.

The TOE is the Android COMSec client, component of the COMSec Secure Communications suite of products.

This Security Target including the security problem definition and the description of the security requirements is based on the *Protection Profile for Application Software (Version 1.2, 22 April 2016)*, although does not claim conformity with this protection profile as is described in section 2.2.

The ST contains the following additional sections:

- Section 2: Conformance Claims
- Section 3: Security Problem Definition
- Section 4. Security Objectives
- Section 5: Security Requirements
- Section 6: TOE Summary Specification

1.1 ST REFERENCE

Category	Id.
ST Title	COMSec Admin+ Client Security Target Lite
ST Version	A/0
ST Date	03/09/2018

Table 1: ST Reference

1.2 TOE REFERENCE

Category	Id.
TOE Title	COMSec Admin+
TOE Software Version	3.1.11a_CA+
TOE Developer	Indra Sistemas de Comunicaciones Seguras

Table 2: TOE Reference

1.3 TOE OVERVIEW

The whole system provides the secure communications through a Virtual Operator (IMS – IP Multimedia System) responsible to manage the communications and its security:

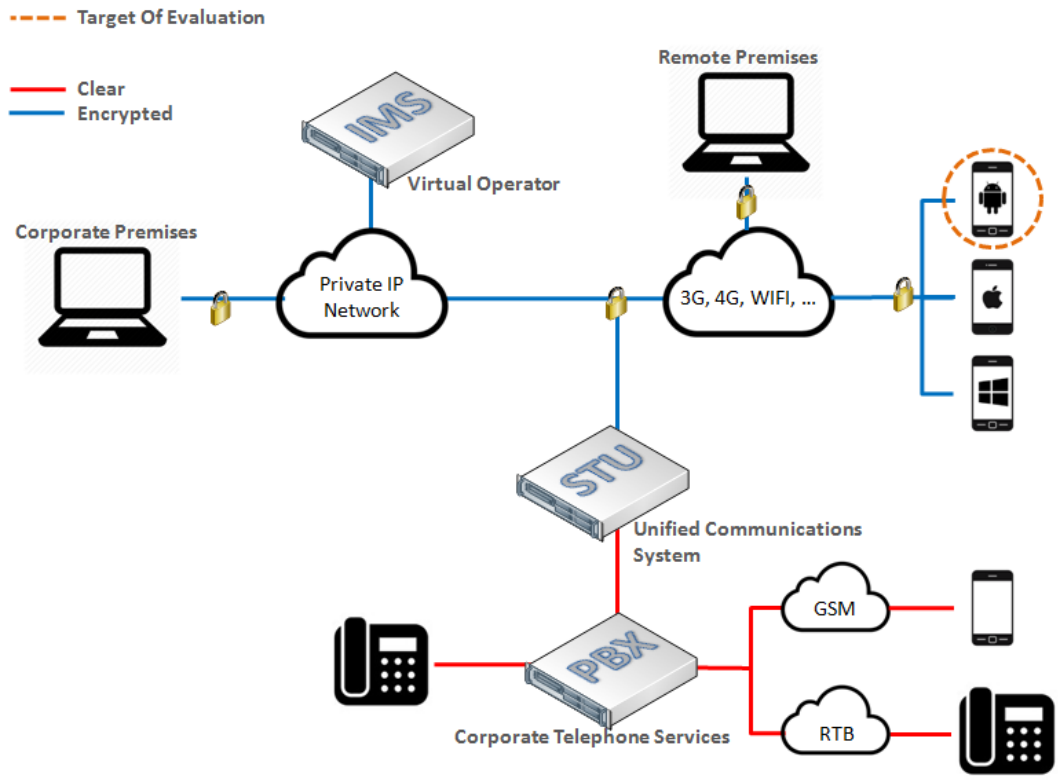

Figure 1. COMSec System Architecture

The TOE is a Secure Communications App for Android devices allowing the user to protect its real time communications (VoIP, Instant Messaging & Data) while using regular public networks (3G, 4G, WIFI …) provided by commercial wireless services company.

1.4 TOE ARCHITECTURE

1.4.1 Physical Boundaries

The TOE is the COMSec Android Application, consisting of an Android Application Package file, with filename extension “apk”. The TOE is delivered to the customer installed in the Smartphone.

Applicable manuals references are provided in section 1.4.5.

1.4.2 Hardware Requirements

The TOE shall be run in a Samsung Galaxy A3 2016 Smartphone.

The Android Application is a software client running on the host platform, and only communicating with the IMS Server, via 3G, 4G or WIFI connections. The IMS Server (secure VoIp/SIP server for

the system), the corporate communications systems, and other client platforms beyond the Android Client are not specified in this Security Target, and lay outside the scope of the TOE.

1.4.3 Software Requirements

The TOE shall be run in Android 5.1.1 Operating System (or higher)

1.4.4 TOE Security Functions

This section summarizes the security functions provided by the TOE:

- Cryptographic Support
- User Data Protection
- Security Management
- Privacy
- Protection of the TSF
- Trusted Path/Channel
- Optional Requirements
- Selection Based Requirements

1.4.4.1 Cryptographic Support

The evaluated platform runs Android 5.1.1 (or higher), and the TOE uses the secure functions provided by this OS in order to generate random numbers when needed (e.g.: during key negotiation protocols).

For cryptographic algorithms the TOE uses the functionalities provided by OpenSSL, which is used as an external library in the Client Application.

1.4.4.2 User Data Protection

The COMSec Android Client protects user data communicated over the network depending on the data type:

- Data Communications (including instant messaging, attached files, etc…) are protected by means of two layers of encryption:
  - TLS secure channel encrypting data using AES-256
  - COMSec specific protection, over the TLS channel, using AES- 256
- Voice Communications are protected using AES- 256
- Sensitive data stored in non-volatile memory is protected using AES-256 and HMAC.

1.4.4.3 Security Management

The TOE functionalities are not available to the user until, the COMSec Android Client has been initialized and provided with the corresponding secure credentials. Once it has been properly initialized the user must provide a password in order to start using the Application services (VoIP, Instant Messaging, etc…).

1.4.4.4 Privacy

The TOE is not requesting any Personally Identifiable Information from the user, and therefore not transmitting it through the network.

1.4.4.5 Protection of the TSF

The TOE performs a proper management of the device memory in, and for secure installation it takes advantage of the Android Package Management.

Regarding the used third-party libraries the TOE package includes only those strictly needed for its proper operation.

1.4.4.6 Trusted Path/Channel

For Data Communications with the IMS Server the TOE uses TLS 1.2 and a proprietary COMSec secure protocol over the previously established TLS channel.

For Voice Communications the TOE uses a proprietary COMSec secure protocol.

1.4.4.7 Optional Requirements

For Symmetric Key Generation the TOE relies on approved Java primitives in order to generate the required Random Numbers:

- During the establishment of the COMSec specific protection Key in the Data Communications channel
- During the establishment of Ephemeral Keys for the Voice Communications channel

Regarding the TLS 1.2 supporting the Data Communications channel it is based on X509.3 certificates.

1.4.4.8 Selection Based Requirements

These requirements are defining:

- How the TOE performs several Cryptographic Operations (Encryption/Decryption, Hashing, Signing, Keyed-Hash Message Authentication)
- How TOE implements the TLS Client Protocol
- How the TOE is dealing with the X.509 Certificate Validation & Authentication

1.4.5 TOE Documentation

AD-01	4941802000000DF01-A2 Descripción Técnica Sistema COMSec.pdf
AD-02	4941802000000MA02-A2 COMSec Admin+ Manual Android v.3.1.1x.pdf
AD-03	4941802000000MA01-A3 Carga de Credenciales COMSec Admin+.pdf
AD-04	4941802000000ES01-A3 Especificación Funcional COMSec Admin+.pdf

2 CONFORMANCE CLAIMS

2.1 CC CONFORMANCE

This TOE is conformant to:

- Common Criteria for Information Technology Security Evaluations Part 1
  - Version 3.1, Revision 4, September 2012
- Common Criteria for Information Technology Security Evaluations Part 2
  - Version 3.1, Revision 4, September 2012: Part 2 extended
- Common Criteria for Information Technology Security Evaluations Part 3
  - Version 3.1, Revision 4, September 2012: Part 3 extended

2.2 PACKAGE CONFORMANCE

This security target meets the security assurance package described in section 5.2 of the *Protection Profile for Application Software (Version 1.2, 22 April 2016)*.

2.3 PROTECTION PROFILE CONFORMANCE

The Security Target does not claim conformance to any protection profile.

2.4 CONFORMANCE RATIONALE

The Security Target takes the following items from the *Protection Profile for Application Software (Version 1.2, 22 April 2016)*:

- Security Problem Definition
- Security Objectives
- Security Requirements

3 SECURITY PROBLEM DEFINITION

The Security Problem Definition has been taken directly from the corresponding section included in the Protection Profile for Application Software v1.2.

3.1 THREATS

Threat Id.	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_EAVESDROP	An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may 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.

3.2 ASSUMPTIONS

Assumption Id.	Description
A.PLATFORM	The TOE relies upon a trustworthy computing platform 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 willfully 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, willfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy.

3.3 ORGANIZATIONAL SECURITY POLICIES

There are no Organizational Security Policies for the application.

4 SECURITY OBJECTIVES

The Security Objectives have been taken directly from the corresponding section included in the Protection Profile for Application Software v1.2.

4.1 SECURITY OBJECTIVES FOR THE TOE

TOE Security Objective Id.	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. Addressed by: FDP_DEC_EXT.1, FMT_CFG_EXT.1, FPT_AEX_EXT.1, FPT_TUD_EXT.1
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 behavior relies upon using only documented and supported APIs. Addressed by: FMT_MEC_EXT.1, FPT_API_EXT.1, FPT_LIB_EXT.1
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. Addressed by: FMT_SMF.1, FPT_TUD_EXT.1.5, FPR_ANO_EXT.1
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. Addressed by: FDP_DAR_EXT.1, FCS_STO_EXT.1, FCS_RBG_EXT.1

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. Addressed by: FTP_DIT_EXT.1, FCS_TLSC_EXT.1, FCS_RBG_EXT.1

4.2 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT

OE Security Objective Id.	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 willfully 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, willfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy

4.3 SECURITY OBJECTIVES RATIONALE

Threat, Assumption, or OSP	Security Objectives	Rationale
T.NETWORK_ATTACK	O.PROTECTED_COMMS, O.INTEGRITY, O.MANAGEMENT	The threat T.NETWORK_ATTACK is countered by O.PROTECTED_COMMS as this provides for integrity of transmitted data. The threat T.NETWORK_ATTACK is countered by O.INTEGRITY as this provides for integrity of software that is installed onto the system from the network. The threat T.NETWORK_ATTACK is countered by O.MANAGEMENT as this provides for the ability to configure the application to defend against network attack.

T.NETWORK_EAVESDROP	O.PROTECTED_COMMS, O.QUALITY, O.MANAGEMENT	The threat T.NETWORK_EAVESDROP is countered by O.PROTECTED_COMMS as this provides for confidentiality of transmitted data. The objective O.QUALITY ensures use of mechanisms that provide protection against network-based attack. The threat T.NETWORK_EAVESDROP is countered by O.MANAGEMENT as this provides for the ability to configure the application to protect the confidentiality of its transmitted data.
T.LOCAL_ATTACK	O.QUALITY	The objective O.QUALITY protects against the use of mechanisms that weaken the TOE with regard to attack by other software on the platform.
T.PHYSICAL_ACCESS	O.PROTECTED_STORAGE	The objective O.PROTECTED_STORAGE protects against unauthorized attempts to access physical storage used by the TOE.
A.PLATFORM	OE.PLATFORM	The operational environment objective OE.PLATFORM is realized through A.PLATFORM.
A.PROPER_USER	OE.PROPER_USER	The operational environment objective OE.PROPER_USER is realized through A.PROPER_USER.
A.PROPER_ADMIN	OE.PROPER_ADMIN	The operational environment objective OE.PROPER_ADMIN is realized through A.PROPER_ADMIN.

5 EXTENDED COMPONENTS DEFINITION

The Extended Security Functional Requirements and Extended Security Assurance Requirements included in this ST are taken from the *Protection Profile for Application Software (Version 1.2, 22 April 2016)*.

All rationales applicable to extended components shall to be considered according to the described in the PP.

6 SECURITY REQUIREMENTS

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the Target of Evaluation (TOE) and to scope the evaluation effort.

All the requirements included in this section have been taken and refined from the Protection Profile for Application Software v1.2

6.1 CONVENTIONS

Common Criteria defines four functional operations allowed to be performed on Functional Requirements:

Operation	Description
Selection	Allows the specification of one or more elements from a list. Indicated in this document with *italicized text*.
Assignment	Allows the specification of an identified parameter. Indicated in this document with ***italicized & bold text***.
Refinement	Allows the addition of details. Indicated in this document with underlined text.
Iteration	Allows a component to be used more than once with varying operations. Indicated in this document by means a number surrounded by parenthesis and placed at the end of the component. For instance: FCS_COP.1(1).

6.2 SECURITY FUNCTIONAL REQUIREMENTS

6.2.1 Cryptographic Support (FCS)

6.2.1.1 FCS_RBG_EXT.1 Random Bit Generation Services

FCS_RBG_EXT.1.1

The application shall [*invoke platform-provided DRBG functionality*] for its cryptographic operations.

6.2.1.2 FCS_STO_EXT.1 Storage of Credentials

FCS_STO_EXT.1.1

The application shall *[implement functionality to securely store:*

- ***User PIN***
- ***TLS Certificates***

] to non-volatile memory.

6.2.2 User Data Protection (FDP)

6.2.2.1 FDP_DEC_EXT.1 Access to Platform Resources

FDP_DEC_EXT.1.1

The application shall restrict its access to [

- *network connectivity*
- *camera*
- *microphone*
- *location services*
- *Bluetooth*
- ***Speaker***
- ***SD external storage***
- ***Battery***
- ***Device Administrator***
- ***Telephony***
- ***Disable Keyguard***
- ***Receive Boot Completed***

] .

FDP_DEC_EXT.1.2

The application shall restrict its access to [*Address book*].

6.2.2.2 FDP_NET_EXT.1 Network Communications

FDP_NET_EXT.1.1

The application shall restrict network communication to [

- *user-initiated communication for [*
  - ***TCP connection to IMS***
  - ***UDP for voice/video calls**]*
- *respond to [**UDP for voice/video calls**]*

] .

6.2.2.3 FDP_DAR_EXT.1 Encryption of Sensitive Application Data

FDP_DAR_EXT.1.1

The application shall [*leverage platform-provided functionality to encrypt sensitive data*] in non-volatile memory.

6.2.3 Security Management (FMT)

6.2.3.1 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.

6.2.3.2 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 it and its data from unauthorized access.

6.2.3.3 FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1

The TSF shall be capable of performing the following management functions [

- *Delete application data*
- *Device factory reset*
- *Device internal storage encryption*

].

6.2.4 Privacy

6.2.4.1 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*].

6.2.5 Protection of the TSF (FPT)

6.2.5.1 FPT_API_EXT.1 Use of Supported Services and APIs

FPT_API_EXT.1.1

The application shall use only documented platform APIs.

6.2.5.2 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 [***none***].

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 compiled with stack-based buffer overflow protection enabled.

6.2.5.3 FPT_TUD_EXT.1 Integrity for Installation and Update

FPT_TUD_EXT.1.1

The application shall [*leverage the platform*] to check for updates and patches to the application software.

FPT_TUD_EXT.1.2

The application shall be distributed using the format of the platform-supported package manager.

FPT_TUD_EXT.1.3

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.1.4

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

FPT_TUD_EXT.1.5

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

FPT_TUD_EXT.1.6

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

6.2.5.4 FPT_LIB_EXT.1 Use of Third Party Libraries

FPT_LIB_EXT.1.1

The application shall be packaged with the following main Third Party Libraries:[

- ***openssl-1.0.2n***
- ***opencore-amr-0.1.5***
-

].

6.2.6 Trusted Path/Channel (FTP)

6.2.6.1 FTP_DIT_EXT.1 Protection of Data in Transit

FTP_DIT_EXT.1.1

The application shall [*encrypt all transmitted data with [TLS]*] between itself and another trusted IT product.

6.3 SECURITY ASSURANCE REQUIREMENTS

6.3.1 Class ASE: Security Target

N/A

6.3.2 Class ADV: Development

6.3.2.1 ADV_FSP.1 Basic Functional Specification (ADV_FSP.1)

Developer action elements:

ADV_FSP.1.1D

The developer shall provide a functional specification.

ADV_FSP.1.2D

The developer shall provide a tracing from the functional specification to the SFRs.

Content and presentation elements:

ADV_FSP.1.1C

The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.2C

The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.3C

The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.

ADV_FSP.1.4C

The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

Evaluator action elements:

ADV_FSP.1.1E

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

ADV_FSP.1.2E

The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

6.3.3 Class AGD: Guidance Documentation

6.3.3.1 AGD_OPE.1 Operational User Guidance (AGD_OPE.1)

Developer action elements:

AGD_OPE.1.1D

The developer shall provide operational user guidance.

Content and presentation elements:

AGD_OPE.1.1C

The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C

The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C

The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C

The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C

The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences, and implications for maintaining secure operation.

AGD_OPE.1.6C

The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C

The operational user guidance shall be clear and reasonable.

Evaluator action elements:

AGD_OPE.1.1E

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

6.3.3.2 AGD_PRE.1 Preparative Procedures (AGD_PRE.1)

Developer action elements:

AGD_PRE.1.1D

The developer shall provide the TOE, including its preparative procedures.

Content and presentation elements:

AGD_PRE.1.1C

The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2C

The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

Evaluator action elements:

AGD_PRE.1.1E

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

AGD_PRE.1.2E

The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

6.3.4 Class ALC: Life-cycle Support

6.3.4.1 ALC_CMC.1 Labeling of the TOE (ALC_CMC.1)

Developer action elements:

ALC_CMC.1.1D

The developer shall provide the TOE and a reference for the TOE.

Content and presentation elements:

ALC_CMC.1.1C

The application shall be labeled with a unique reference.

Evaluator action elements:

ALC_CMC.1.1E

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

6.3.4.2 ALC_CMS.1 TOE CM Coverage (ALC_CMS.1)

Developer action elements:

ALC_CMS.1.1D

The developer shall provide a configuration list for the TOE.

Content and presentation elements:

ALC_CMS.1.1C

The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

ALC_CMS.1.2C

The configuration list shall uniquely identify the configuration items.

Evaluator action elements:

ALC_CMS.1.1E

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

6.3.4.3 ALC_TSU_EXT.1 Timely Security Updates

Developer action elements:

ALC_TSU_EXT.1.1D

The developer shall provide a description in the TSS of how timely security updates are made to the TOE.

ALC_TSU_EXT.1.2D

The developer shall provide a description in the TSS of how users are notified when updates change security properties or the configuration of the product.

Content and presentation elements:

ALC_TSU_EXT.1.1C

The description shall include the process for creating and deploying security updates for the TOE software.

ALC_TSU_EXT.1.2C

The description shall express the time window as the length of time, in days, between public disclosure of a vulnerability and the public availability of security updates to the TOE.

ALC_TSU_EXT.1.3C

The description shall include the mechanisms publicly available for reporting security issues pertaining to the TOE.

Evaluator action elements:

ALC_TSU_EXT.1.1E

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

6.3.5 Class ATE: Tests

6.3.5.1 ATE_IND.1 Independent Testing – Conformance (ATE_IND.1)

Developer action elements:

ATE_IND.1.1D

The developer shall provide the TOE for testing.

Content and presentation elements:

ATE_IND.1.1C

The TOE shall be suitable for testing.

Evaluator action elements:

ATE_IND.1.1E

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

ATE_IND.1.2E

The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

6.3.6 Class AVA: Vulnerability Assessments

6.3.6.1 AVA_VAN.1 Vulnerability Survey (AVA_VAN.1)

Developer action elements:

AVA_VAN.1.1D

The developer shall provide the TOE for testing.

Content and presentation elements:

AVA_VAN.1.1C

The application shall be suitable for testing.

Evaluator action elements:

AVA_VAN.1.1E

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

AVA_VAN.1.2E

The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.1.3E

The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

6.4 OPTIONAL REQUIREMENTS

6.4.1.1 FCS_CKM.1(2) Cryptographic Symmetric Key Generation

FCS_CKM.1.1(2)

The application shall generate symmetric cryptographic keys using a Random Bit Generator as specified in FCS_RBG_EXT.1 and specified cryptographic key sizes [*256 bit*]

6.4.1.2 FCS_TLSC_EXT.2 TLS Client Protocol

FCS_TLSC_EXT.2.1

The application shall support mutual authentication using X.509v3 certificates.

6.5 SELECTION BASED REQUIREMENTS

6.5.1.1 FCS_CKM_EXT.1 Cryptographic Key Generation Services

FCS_CKM_EXT.1.1

The application shall [*implement asymmetric key generation*].

6.5.1.2 FCS_CKM.1(1) Cryptographic Asymmetric Key Generation

FCS_CKM.1.1(1)

The application shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm [*ECC schemes*].

6.5.1.3 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, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"*].

6.5.1.4 FCS_COP.1(1) Cryptographic Operation - Encryption/Decryption

FCS_COP.1.1(1)

The application shall perform encryption/decryption in accordance with a specified cryptographic algorithm

> AES (as defined in NIST SP 800-38 series)

and cryptographic key sizes 256-bit and [*no other key sizes*].

6.5.1.5 FCS_COP.1(2) Cryptographic Operation - Hashing

FCS_COP.1.1(2)

The application shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm [*SHA*] and message digest that meet the following: FIPS Pub 180-4.

6.5.1.6 FCS_COP.1(3) Cryptographic Operation - Signing

FCS_COP.1.1(3)

The application shall perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm [*ECDSA schemes*].

6.5.1.7 FCS_COP.1(4) Cryptographic Operation - Keyed-Hash Message Authentication

FCS_COP.1.1(4)

The application shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm

> none

and

> [*SHA*]

with key sizes and message digest that meet the following: FIPS Pub 198-1 The Keyed-Hash Message Authentication Code and FIPS Pub 180-4 Secure Hash Standard.

6.5.1.8 FCS_TLSC_EXT.1 TLS Client Protocol

FCS_TLSC_EXT.1.1

The application shall [*implement TLS 1.2 (RFC 5246)*] supporting the following cipher suites:

Mandatory Cipher Suites:

None

Optional Cipher Suites:

[*TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*].

FCS_TLSC_EXT.1.2

The application shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.1.3

The application shall establish a trusted channel only if the peer certificate is valid.

6.5.1.9 FCS_TLSC_EXT.4 TLS Client Protocol

FCS_TLSC_EXT.4.1

The application shall present the supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [*secp384r1*] and no other curves.

6.5.1.10 FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1.1

The application shall [*invoked platform-provided 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.
- The application shall validate the status of the certificate when connecting to server. The server may remove user certificate avoiding the TLS channel establishment.
- The application shall validate the extendedKeyUsage field according to the following rules:
    - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
    - 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 extendedKeyUsage field.
    - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.
    - 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 extendedKeyUsage 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.

6.5.1.11 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*].

6.6 SECURITY REQUIREMENTS AND DEPENDENCY RATIONALE

The Security Functional Requirements and Security Assurance Requirements included in this ST are taken from the *Protection Profile for Application Software (Version 1.2, 22 April 2016)*.

All hierarchical relationships and dependencies in this ST are considered to be identical to those that are defined in the PP.

7 TOE SUMMARY SPECIFICATION

This chapter describes how the TOE meets the Security Functional Requirements presented in 6.2, 6.4 and 6.5.

7.1 CRYPTOGRAPHIC SUPPORT (FCS)

SFR	Rationale
FCS_RBG_EXT.1.1	The TOE uses the `java.security.SecureRandom` class for DRGB functionality
FCS_STO_EXT.1.1	• The TOE store credentials encrypted in application internal storage space.

7.2 USER DATA PROTECTION (FDP)

SFR	Rationale
FDP_DEC_EXT.1.1	The TOE sets the permissions by means of the corresponding Android configuration file. These permissions are needed in order to: • *network connectivity* : Communication with server and with endpoint users. • *camera, microphone, bluetooth, speaker* : Manage voice and video calls • *location services* : Send device location to server if needed. • *SD external storage* : Manage files of application • *Battery* : Control and log battery status • *Device Administrator*: Encrypt internal storage and perform a device factory reset on demand. • *Telephony:* Detect GSM calls to manage OnHold state • *Disable Keyguard:* Allow the user to answer calls when the terminal is locked without having to unlock it. • *Receive Boot Completed:* Start application at boot.
FDP_DEC_EXT.1.2	The TOE sets the permissions by means of the corresponding Android configuration file. Address book access is needed in order to perform calls to personal contacts.
FDP_NET_EXT.1.1	The TOE is only using network communications with IMS server for signalling, voice and instant messaging.
FDP_DAR_EXT.1.1	The TOE is storing sensitive data in the application private directory.

7.3 SECURITY MANAGEMENT (FMT)

SFR	Rationale
FMT_MEC_EXT.1.1	• The TOE uses SharedPreferences functionality for storing configuration options.

FMT_CFG_EXT.1.1	On fresh installations the TOE only permits credentials provisioning.
FMT_CFG_EXT.1.2	The TOE files have appropriate access permissions
FMT_SMF.1.1	The TOE is not transmitting data related with the management functions: • No hardware, software or configuration data from the TOE is transmitted • No PII data is transmitted • No application state data is transmitted • No network backup functionality is available in the TOE

7.4 PRIVACY

SFR	Rationale
FPR_ANO_EXT.1.1	The TOE is not transmitting PII related data.

7.5 PROTECTION OF THE TSF (FPT)

SFR	Rationale
FPT_API_EXT.1.1	The TOE leverages the following platform provided Application Programming Interfaces: • com.android.support:support-v4 • com.android.support:appcompat-v7
FPT_AEX_EXT.1.1	The TOE is not invoking mmap functionalities. The application does not use any explicit flag for enabling ASLR when compiling. The platform is responsible of enabling ASLR.
FPT_AEX_EXT.1.2	The TOE is not invoking mmap nor mprotect functionalities.
FPT_AEX_EXT.1.3	The TOE can be executed on the latest version of Android.
FPT_AEX_EXT.1.4	The TOE is not storing executable files in the `internal application` directory
FPT_AEX_EXT.1.5	The TOE uses –fstack-protector-all for compiling native libraries
FPT_TUD_EXT.1.1	The TOE APK can only be obtained by means of the customer provided MDM.
FPT_TUD_EXT.1.2	The TOE is packaged using the Android APK format,
FPT_TUD_EXT.1.3	After the TOE is removed from the host no files are left on the file system beyond the exceptions set in the requirement (e.g.: log files)
FPT_TUD_EXT.1.4	The TOE does not make any modification to its own binary code.
FPT_TUD_EXT.1.5	The user can access the TOE software version by means of a specific functionality provided in its GUI.
FPT_TUD_EXT.1.6	The TOE installation/update packages are digitally signed. Android applications are distributed under the APK file format. The developer is the only authorized source for signing the distribution package.

	The platform verifies the package at installation time. TOE updates must be signed with the same key of the original APK or the platform will not allow the update.
FPT_LIB_EXT.1.1	The TOE is including, among others, third party libraries listed in the SFR.

7.6 TRUSTED PATH/CHANNEL (FTP)

SFR	Rationale
FTP_DIT_EXT.1.1	The TOE ensures that every data exchanged through the TCP channel with the IMS Server is encrypted with TLS (beyond this, there is an extra layer of proprietary COMSec encryption).

7.7 OPTIONAL REQUIREMENTS

OR	Rationale
FCS_CKM.1.1(2)	The TOE uses the `java.security.SecureRandom` class, which is one of the methods proposed in the PP in order to be compliant with FCS_RBG_EXT.1
FCS_TLSC_EXT.2.1	The TOE supports mutual authentication using X.509v3 certificates

7.8 SELECTION BASED REQUIREMENTS

SBR	Rationale
FCS_CKM_EXT.1.1	The TOE implements asymmetric key generation.
FCS_CKM.1.1(1)	The TOE uses asymmetric key generation for key establishment purposes.
FCS_CKM.2.1	For key establishment the TOE supports Elliptic-Curve based mechanisms based in the required NIST SP publication 800-56A.
FCS_COP.1.1(1)	The TLS channel established between the TOE and the IMS Server supports the AES encryption algorithm included as optional in the requirement.
FCS_COP.1.1(2)	The TOE uses SHA-384 during the negotiation of the TLS cannel.
FCS_COP.1.1(3)	The TOE uses elliptic curves for signatures services.
FCS_COP.1.1(4)	The TOE supports HMAC-SHA-512 during the negotiation phase while establishing the COMSec channel over the TLS connection.
FCS_TLSC_EXT.1.1	The TLS channel established between the TOE and the IMS Server supports the TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 suite included as optional in the requirement.
FCS_TLSC_EXT.1.2	The TOE identifier is compliant with the reference identifier specified in RFC 6125.

FCS_TLSC_EXT.1.3	The TOE verifies that the IMS certificate is valid, and only in that case proceeds to establish the TLS channel.
FCS_TLSC_EXT.4.1	The TOE presents the secp384r1 Elliptic Curve Extension in the Client Hello during the TLS channel establishment. This curve is used by default and may not be configured.
FIA_X509_EXT.1.1	Certificate path validation algorithm is done according to RFC5280, using OpenSSL provided functionality.
FIA_X509_EXT.1.2	The TOE checks that CA certificates include the Basic Constraints extension and that their CA flag is set to TRUE.
FIA_X509_EXT.2.1	The TOE uses X.509v3 certificates during the TLS authentication.