Security Target for Dencrypt Server System version 5.0

ST Version 0.17

Executive summary

This document is the Common Criteria Security Target for Dencrypt Server System. It is following the specification given in Part 1 annex A of the Common Criteria version 3.1 Revision 5.

Contents

1 Introduction

1.1 Security Target identification and organisation

Title:	Security Target for Dencrypt Server System version 5.0
ST Version:	0.17
Status:	Released
Date:	2021-06-11
Sponsor:	Dencrypt A/S
Developer:	Dencrypt A/S
Keywords:	Mobile application management, VoIP, voice and message encryption

This Security Target (ST) has been structured in accordance with [CC] Part 1. The main sections of the ST are the introduction, security problem definition, security objectives, security requirements, TOE summary description and annexes.

The introduction provides general information about the TOE, serves as an aid to understand the nature of the TOE and its security functionality and provide context for the evaluation.

The security problem definition describes the security aspects of the environment in which the TOE is to be used and the manner in which it is to be employed. The TOE security environment includes:

a) assumptions regarding the TOE's intended usage and environment of use

b) threats relevant to secure TOE operation

c) organisational security policies with which the TOE must comply

The security objectives reflect the stated intent of the ST. They pertain to how the TOE will counter identified threats and how it will cover identified organisational security policies and assumptions. The security objectives are divided into security objectives for the TOE and for the environment. The security objectives rationale demonstrates that the stated security objectives are traceable to all of the aspects identified in the TOE security problem definition and that they are suitable to cover them.

The extended components section identifies any extended security requirements, i.e. requirements that in addition to requirements defined in CC Part 2 and 3 are used within this ST.

The security requirements section provides detailed requirements, in separate subsections, for the TOE and its environment. The security requirements are further divided into the TOE security functional requirements and the TOE security assurance requirements.

The TOE summary specification addresses the security functions that are represented by the TOE to answer the security requirements.

The annex contains a list of abbreviations and a glossary relevant for this ST.

1.2 TOE identification

The TOE is the Dencrypt Server System version 5.0. The complete build version is 5.0.0.122. The Dencrypt Server System is the server part of the Dencrypt Communication Solution and consists of the following Dencrypt software components:

- Dencrypt Certificate Manager (DCM)
- Dencrypt Provisioning Server (DPS)
- Dencrypt Control Center (DCC)
- Dencrypt Database (DDB)

- Dencrypt Communication Server (DCS)
- Dencrypt Server Bridge (DSB)

For more detailed description of the Dencrypt Server System TOE, please refer to 1.5.2.1. For a description of the TOE scope and 3rd party components of the TOE, please refer to 1.5.4.

1.3 TOE type

The TOE is software only and consists of a VoIP and messaging server together with management components that are part of the server system of the Dencrypt Communication Solution. The solution supports mobile device clients for end-to-end encrypted voice, video and messaging between iPhones.

1.4 TOE overview

The Dencrypt Communication Solution consists of Dencrypt Server System (the TOE) and Dencrypt app on mobile devices. The Dencrypt Server System supports management and end-to-end communication for mobile device clients. The server system consists of a Dencrypt Communication Server (a SIP and Lime server), a Dencrypt Database (provides database services to DCS), a Dencrypt Certificate Manager (signs server and client certificates), a Dencrypt Provisioning Server (provisions clients), a Dencrypt Control Center (provides administrator interface), and a Dencrypt Server Bridge (communicates with other Dencrypt Server Systems). Only the Dencrypt Server System is part of the TOE. The other parts are not within the scope of the TOE, but are considered as necessary parts of the TOE environment. The Dencrypt App, referred to as the Dencrypt Connex Application (DCA), or any other apps on the handset are not included into the TOE.

Note: The Dencrypt app is a critical and the most exposed component of the Dencrypt Communication Solution and is therefore subject to a separate EAL4+ evaluation.

The main security features of the TOE are:

- Administration:
  - Identification and authentication of administrators
  - Administrative roles and privileges associated with those role
  - Management functions, for managing the DCA clients and TOE itself
  - Auditing and audit review
- Secure provisioning of DCA clients
- Trusted channel to clients
- Trusted channel to service access
- Provisioning to end users of new configurations and central managed phone books
- Key generation and certificate issuing and a certificate authority
- Secure bridge to another DSS
- Encryption of push notifications
- TCP tunnelling for voice or video communication

1.5 TOE description

1.5.1 Introduction and intended use

The key feature of the Dencrypt Server System (DSS) and the Dencrypt Communication Solution is to provide mobile devices with secure end-to-end voice, video and message communication

within closed user groups that are centrally managed. Within the Dencrypt Communication Solution the DSS provides secure provisioning, DCA management and secure communication establishment.

1.5.2 The TOE architecture and key functions

1.5.2.1 Introduction

The TOE is part of the Dencrypt Communication solution and consists of the server system components. The whole Dencrypt Communication Solution is shown in the picture below. The components that are marked blue are those developed by Dencrypt. On the Server side, in addition to Dencrypt developed components, certain 3rd party components are required to provide security functionalities. These 3rd party components include Debian Linux operating system, Apache server, PHP, Laravel framework and MySQL database. The TOE consists of Dencrypt developed server system components and the before mentioned 3rd party components.

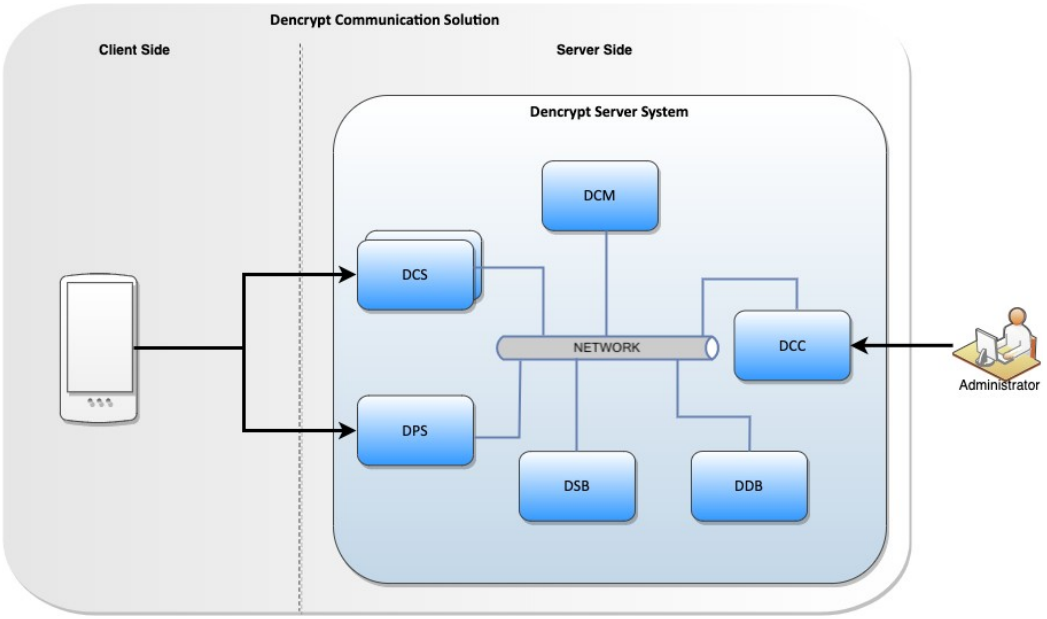


Illustration 1: Dencrypt Communication Solution overview

The functionality of the main components is described in more details below, also indicating which components are part of the TOE and which are not:

Dencrypt Connex Application (TOE environment)

The Dencrypt Connex Application (DCA) is a mobile SIP and messaging client that runs on a mobile device (e.g. an iPhone). The client is able to establish encrypted calls and messaging between clients on other mobile devices using the SIP Server and Lime server of the Decrypt Communication Server. The client is installed and updated using the Apple App Store. The client must be configured and initialised before being used. This is done using the provisioning service provided by the TOE.

Dencrypt Provisioning Server (part of TOE)

The Dencrypt Provisioning Server (DPS) is used to initialise clients with user credentials, DCS URL and respond to the Certificate Signing Request (CSR) sent from the client to set up its certificate. During provisioning, the client is provided with a HTTPS web link for the initialisation. The link can also be encoded into a QR code which the app can scan. The link is provided in a secure way as part of the TOE

environment. The HTTPS web link points to the web server of the DPS, which is authenticated by the client. The link contains a token which is validated by the server.

Dencrypt Communication Server (part of TOE)

The Dencrypt Communication Server (DCS) provides the connections and communication services to the DCA:

- SIP Server – necessary for the clients to establish voice or video communication between two or more clients. Also provides routing for messaging.
- Lime server – provides the key-exchange functionality to initiate secure message communication.
- Tunnel server - provides a server to tunnel voice or video communication over TCP.

Dencrypt Database (part of TOE)

The Dencrypt Database (DDB) provides the database services for the DCS. It keeps the user data and most meta data e.g. call statistics.

Dencrypt Control Center (part of TOE)

The user management is performed using the Dencrypt Control Center (DCC). The user management means creating/deleting/revoking users and groups, as well as adding and removing users from these groups. The DCC offers a web interface that is accessible using a web browser from the administrator's local machine.

Dencrypt Certificate Manager (part of TOE)

Dencrypt Certificate Manager (DCM) is the central point for TLS certificates in the system. Once provisioning has taken place, all connections between the DCA and DSS use mutually authenticated TLS connections. The required TLS certificates are issued by the DCM via the following procedure: The client or server generates the private/public key pair and creates a CSR. The CSR is sent to the DCM which signs the CSR if permitted. The DCM provides the certificate back to client/server for employment. All communication between the DCM and the DCA take place via the DCS, except during provisioning where it takes place via the DPS.

Dencrypt Server Bridge (part of TOE)

The Dencrypt Server Bridge (DSB) introduces functionality to make Dencrypt Calls between users on two different DSS. It enables Secure Phonebook synchronization between systems and routing of SIP or messaging data between systems. The connection between two DSS are protected by a TLS channel. The DSB ensure that this connection is authenticated.

All the above-mentioned backend servers (DCC, DCS, DPS, DCM, DDB and DSB) are installed with a turnkey Linux distribution which includes, among other things, Debian Linux operating system, Apache server and PHP. Debian Linux, Apache and PHP are part of the TOE. The DCC supports configuration of backups for all systems to a secure storage server.

The DCC has two additional components that are also part of the TOE: the Laravel framework and the MySQL database. The Laravel framework is a collection of libraries and services that handle common server side tasks such as encryption, database connection, CLI, emails, error handling, etc. The MySQL database contains information about administrators, server connections, preferences and permissions.

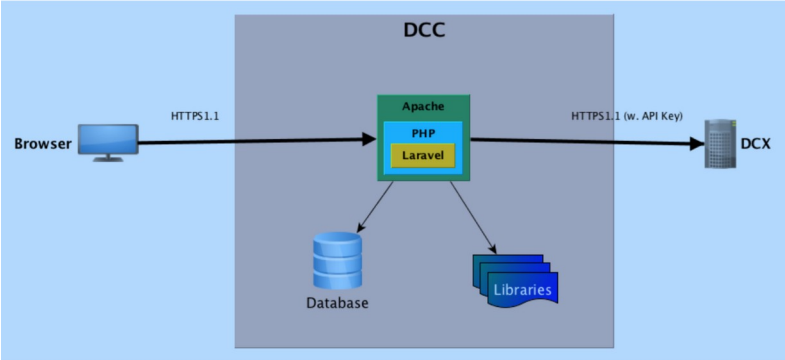


Illustration 2, The DCC Overview

1.5.2.2 Provisioning and user registration process

The provisioning process consists of two independent steps:

- Installation of DCA
- Provisioning of the data to the DCA where the data are the following:
  - DCS user credentials,
  - DSS domain,
  - Signed certificate from the CSR

The DCA is provided and installed using Apple's App Store. The App Store delivery mechanism operated by Apple is assumed to be trusted and secure.

Note that the DCA is delivered in an unprovisioned state. One possibility is to us MDM to configure apps apps, however, MDMs usually configure apps with constant values such as user names or server URLs. The provisioning uses one-time tokens, which are by default not constant. Additionally, there might be a separation of duties between MDM administrators and DCA administrators. Thus, DSS provides its own provisioning server to facilitate the initial configuration of DCA.

Provisioning is started by the Dencrypt administrator adding the user to the DSS system and directory. After that the administrator will send an invitation message e.g. by email to the user's handset. The invitation message has a link to the web server, the user shall tap the link which starts DCA. DCA parses the link, fetches the provisioning data from the DPS and installs the data. The link can also be encoded into a QR code, which the application can scan. The provisioning data are deleted on the DPS, i.e. the HTTPS link can be used only once. Additionally, the link is only valid for a limited time after the link has been provided. The URL, i.e. the token in the URL, is verified by the server when the client connects. The URL check is implemented in the TLS module and a token check failure results in a TLS connection termination. Thus, Dencrypt describes the DPS TLS connection as token-based client authenticated. This feature allows to connect securely to the DPS from any internet connection and securely provision a new device.

1.5.2.3 Managing settings and phone book

The DCA (TOE environment) only allows calls and messages to persons listed in the local phonebook as well as to predefined emergency contacts. The local phonebook is individual for each user and contains only the persons which a user is allowed to call. Thus, each user may have a different phonebook. Note you may be able to receive calls from users not in your phone book. The user administrator for the DSS (part of TOE) can change the groups of users to whom a specific user can call to or message at any time. The administrators also have the ability to send out user notifications to one or more DCAs regardless of the phonebook. Users of the DCA have

the ability to call users of a different DSS if that system is connected via the DSB. The administrator of a DSB can then push phonebooks to the remote system.

The DCS takes care of distributing the phonebook to the individual users. When a user starts the DCA, it establishes a TLS connection to the DCS and makes a SIP registration. When registration is successful, the client will send a web request to check if the phonebook or settings have changed. If the phonebook has changed, the DCS notifies the DCA about the current phonebook version. The client downloads the phonebook if its currently used phonebook version does not match the advertised phonebook version. Note, if the client has no phonebook, it is considered as phonebook version 0. The same method applies for settings distribution. The following figure displays the described process.

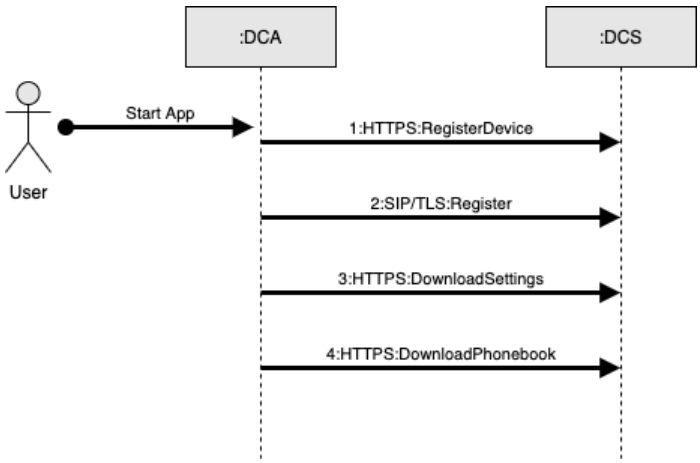


Illustration 3: Registration process

In addition to the phonebook, other settings are also controlled by the DSS. For instance, the ability to integrate with native iOS call history. The TOE also assists the DCA in protecting data-at-rest by storing the encrypted storage encryption key of the DCA. This key is submitted at the end of provisioning. The DCA must connect to the TOE, retrieve and decrypt this key to be able to decrypt its stored data.

1.5.2.4 Making secure calls

The uniqueness of the Dencrypt Communication Solution is that the end-to-end encrypted voice, video and messaging uses Dynamic Encryption, which ensures that each call session is encrypted using a randomly chosen algorithm and randomly chosen keys. The calls are made between two or more DCAs (TOE environment).

The following figure illustrates the steps for a secure call.

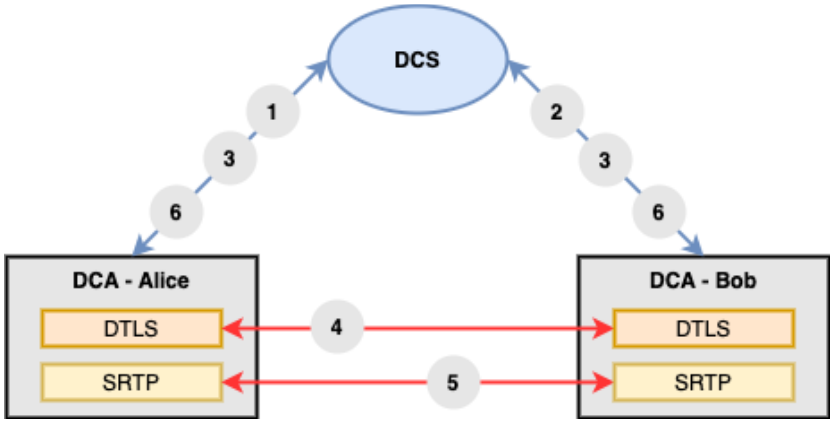


Illustration 4: Secure voice or video call description

1. Alice's DCA contacts the DCS she is registered to.
2. The SIP server resolves Bob's address and contacts Bob's DCA. This resolution is limited because Alice can only contact the DCS for users listed in her phonebook.
   Note: For calls to users of a different DSS, these are forwarded via the DCS. When a call is received to a user outside of the current system, the call establishment (SIP) is forwarded to the DCS of a remote system from the DCS of the TOE over a TLS connection.
3. SIP takes care of signalling, i.e. triggers Bob's DCA to start ringing. As soon as Bob accepts the call, both DCAs are signalled to start a media session for the real-time audio data stream.
4. Before the audio connection is encrypted, DTLS-SRTP takes over the data of media session. DTLS-SRTP exchanges certificates and fingerprints to authenticate the callers. A shared secret is then negotiated between Alice and Bob's DCA. Additionally, DTLS-SRTP has been modified to securely negotiate the ciphersuite to be used for Dynamic Encryption.
   Note: The TOE has the ability to provide a tunnelling service to tunnel all encrypted voice and video communication data over TCP and TLS. Note that this tunnel is not used for security purposes as the tunnelled call audio data is still end-to-end encrypted.
5. Once DTLS-SRTP has established the shared secret, it calculates different keys for the bi-directional audio data stream between Alice and Bob. The key and the Dynamic Encryption parameters derived using Key Boosting. These are required for the dynamic encryption of the audio data stream. The dynamically encrypted real-time data are transported over the IP network by the secure variant of the real-time protocol, so called SRTP.
6. When Bob ends the call, the DCS signals the call termination to Alice. All key material is erased.

The following list characterises the secure call in the Dencrypt Communication Solution:

- Dynamic encryption of voice or video data is implemented as multiple layers of encryption optimized for voice or video data over the SRTP protocol.
- Voice and video communication is bidirectional, i.e. each needs to encrypt and decrypt data and each bit stream uses different keys.
- The DCA complete a DTLS handshake over ECDHE to initiate the secure channel.
- DTLS-SRTP provides a 256bit key and 96bit IV to the SRTP-KDF initiating the end-to-end encrypted communication.

- To authenticate the callers after call setup each caller will provide the SIP ID and the certificate of the remote party to the server, and verify that the SIP ID is associated to that certificate.
- Dynamic Encryption for voice and video use the following keys, which are derived from the DTLS-SRTP shared secret using Key Boosting:
  - 256 bit key for the standard AES-256 encryption. The key is provided by DTLS-SRTP.
  - 128 bit Dynamic Encryption algorithm selection key that defines the S-box for an additional AES-round.
- Dynamic Encryption keys and algorithm are established at call setup and destroyed as soon as the call is terminated.
- Random number generation using RNG on Apple iOS (TOE environment).

1.5.2.5 Secure Messaging

DCA (TOE environment) provides secure end-to-end encrypted messaging between two or more end users. Secure messaging enables asynchronous text based communication between the parties, where attachments can also be sent. Messaging is implemented through the LIMEv2 protocol. The DCS (TOE) is acting as a trusted server providing identification of peer devices and routing of messages.

1. Key pairs are generated by the DCAs and the public keys are uploaded to the Lime server for the X3DH protocol.
2. The sender of a message downloads these keys, generates an ephemeral key pair, and performs the following operations: Diffie-Hellman computations generates shared keys that are fed into a Double Ratchet KDF, which in turn generates a chain key. The chain key is fed through the KDF to compute a message key and a new chain key.
3. The 256-bit message key uses Dencrypt Key Boosting to create a 384-bit key that is used for Dynamic Encryption.
4. The encrypted message is sent over SIPS together with the sender's public identity keys and DH public keys.
5. The receiver will use the keys to perform the same key derivation process and decrypt the message.
   Note: After having established a shared secret once, only Double Ratchet is used to derivate keys and not X3DH.

Secure messaging can also be used between users of different DSS, which is demonstrated in the following figure.

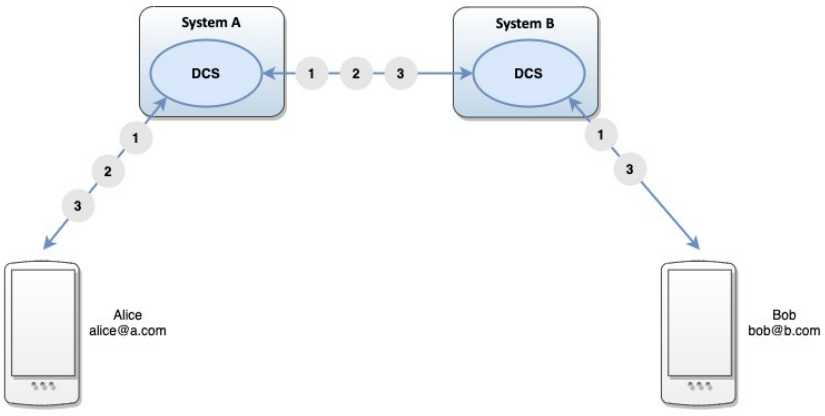


Illustration 5: Secure messaging over the DSB

1. Alice sends an invite to a chat room. The chat room is created on the DCS on system A, but the invitation is forwarded to the DCS on system B and to Bob.
2. Alice requests keys from Bob from the Lime server. The key request is forwarded to System B.
3. Alice and Bob can then exchange encrypted messages, which are forwarded via the DCS.

1.5.2.6 The TLS connection

All connections made between the TOE and any other external component, including the DCA mobile client, the administrator browser and other DSS, are established using a trusted channel implemented using TLS version 1.2.

Push notifications sent from the server to the DCA are not sent through this TLS channel, but instead transmitted via the mobile operating system vendor's (in this case Apple) systems. These are encrypted by the server using AES.

The TLS connection to from the DCA is mutually authenticated to ensure that only authorized clients can establish connections, i.e. to retrieve the phonebook information, and that the DCA is not connecting to a server system that may deceive the DCA user with false phonebooks or provisioning data. The TLS connection also ensures the confidentiality and integrity of any data transmitted. The TLS connection is always initiated by the DCA and never by the server system, such as the DCS or DPS. TLS is also used by the DSB to connect to other DSS for enabling calls between different setups of the DSS.

There is no client authentication of the TLS connection between the administrator browser and the DCC. However, the browser is assumed to perform server authentication. There is no TLS-based client authentication since administrators also have to authenticate themselves to the server anyhow.

Note that the TLS connection to the provisioning web server is not mutually authenticated because the DCA is not yet configured and has no signed client certificate. However, the client will perform server authentication.

Finally, a TLS connection can also be used to tunnel the encrypted voice or video communication over TCP. If SRTP communication is not possible over the networks connecting two DCA users, the TOE can provide a TCP tunnel for said communication. The tunnel does not provide the protection of the traffic, since it is still end-to-end encrypted via SRTP.

Details of the protocols and cipher suites used are provided in the TOE Summary Specification.

1.5.2.7 The SSH service access connection

Remote service access will have access to the TOE, using a SSH connections. Having service access means to have full access to install and configure the TOE. Once identified and authenticated, the service administrator will have a shell command access to the Debian Linux operating system. The SSH connection is implemented using the SSH-2 protocol that relies on the OpenSSL for the cryptographic primitives.

1.5.3 Security functions

This section provides a summary of the security functions implemented by the TOE. The TOE is only a portion of the Dencrypt Communication Solution, and the TOE mainly is there to support the core security functionality, i.e. to provide secure voice, video and message communication, it is necessary to describe the security functionality of the TOE separately.

Within the Dencrypt Communication Solution, the TOE provide the following functionality:

- Secured call setup via a dedicated SIP server
- Secure messaging using the Lime v2.0 protocol

- Provisioning to end users of configurations
- User and phonebook management

In doing this the TOE provides the following security functionality:

- Administration:
  - Identification and authentication of administrators
  - Administrative roles and privileges associated with those role
  - Management functions, for managing the DCA clients and TOE itself
  - Auditing and audit review
- Secure provisioning of DCA clients
- Trusted channel to clients
- Trusted channel to service access
- Provisioning to end users of new configurations and central managed phone books
- Key generation and certificate issuing and a certificate authority
- Secure bridge to another DSS
- Encryption of push notifications
- TCP tunnelling for voice or video communication

1.5.4 Physical scope of the TOE

The TOE is software only and limited to the Dencrypt Server System components as well as the user documentation. The following documentation is provided to the administrators:

- *Operational User Guide v. 5.0*
- *Preparative Guide v. 5.0*

The TOE is delivered as an ISO image containing the TOE (Dencrypt developed server system components, the Debian Linux operating system, the Apache server, PHP, the Laravel framework, the MySQL database, and the regular dependencies for these software components.)"

The delivery, installation and initial configuration is performed by Dencrypt employees or by personnel that have been trained to perform delivery and installation on behalf of Dencrypt.

1.5.4.1 IT environment

The IT environment must provide the following:

- Mobile devices (iPhones with iOS) where the DCA App is installed.
- The virtual or physical amd64 or x86_64 server systems to host and run the DSS components.
- Mail server to send new DCA user invitations.
- Push server used for sending notifications to DCA clients. For this evaluation of the DCA on iOS, Apple Push Notification (APN) is used.
- NTP Server to provide correct time to the DSS components.
- An administrative client and browser for the TOE administrator to manage the TOE.

2 Conformance claims

2.1 CC conformance claim

This ST is CC Part 2 extended and CC Part 3 conformant. This ST claims conformance to CC version 3.1 Revision 5, April 2017.

This ST claims conformance to the EAL2 package of security assurance requirements, augmented with ALC_FLR.2. This ST does not claim conformance to any Protection Profile (PP).

2.2 Conformance rationale

In general, assurance requirements must be commensurate with the exposure of systems to untrustworthy and unauthorized entities. The EAL2 level was also deemed sufficient because this will provide a necessary assurance for a product that is not directly exposed to external attackers, but still able to resist attacker with basic attack potential.

The assurance requirements of the EAL2 package provides a full Security Target and requires an analysis using a functional and interface specification and a basic description of the architecture of the TOE, which would give sufficient confidence in the design and architecture for a meaningful vulnerability analysis that is considered necessary and sufficient to support the use of the more exposed handsets.

3 Security problem definition

It is assumed that the TOE is under physical and logical control of the organization using it, so that it is operated in a data center by administrators who are trained and trusted to operate crypto systems for the organisation. That its connections to the outside is through firewalls, preventing any other access or protocols than the ones that are provided by the TOE. Although the users are assumed to be trustworthy and trained, we cannot exclude that mistakes are being made. For this reason is also assumed that attackers have an attack potential that is limited to basic.

3.1 Threats

This section of the security problem definition describes the threats that are countered by the TOE, its operational environment, or a combination of the two. Threat agents are typically characterized by a number of factors such as expertise, available resources, and motivation, with the motivation being linked directly to the value of the assets at stake.

For this TOE, the assets to be protected consists of:

- TSF data – software, configuration files, audit records, authentication information and cryptographic keys that are controlling the behaviour of the TSFs or is a result of a TSF, and relevant to determine if the TOE is in a secure state.
    - Examples of TSF data for TOE are certificates, audit records and provisioning links.
- User data – any other data of users that is stored on the TOE.
    - Examples of user data for the TOE are data related to the DCA such as phonebook data, SIP data and DCA settings.

There are two types of threat agents. The first type are external entities not authorized to access TSF services. Those may attempt to get access to TSF services either by masquerading as an authorized entity or by attempting to use TSF services without proper authorization. External threat agents may also passively capture data transmitted between the TOE and other trusted parties, or actively manipulate such data. Such a threat agent have a limited attack surface, due to the fact that external connections are limited to TLS authenticated connections and are protected with a firewall.

The second type consists of local users that unintentionally or out of curiosity tries to access information or use resources that they are not authorized for. Since the TOE is used in a controlled environment, such an attack will be limited to a basic attack potential.

The following threats are addressed by the TOE and the TOE environment.

Threat	Description
T.COMMUNICATION	An external attacker reads or manipulates information transmitted between the TOE and components that are outside of the trusted network. This affects both user and TSF data.
T.MASQUERADE	An external attacker gain read or write access to information or resources that are held by the TOE including user data, phone books, audit information or any other TSF data.
T.UNAUTH	An administrator may by accident access data or use management functions for which they have not been authorised to, to read, modify or destroy security critical TSF data or tamper with the TSFs.
T.UNDETECTED	An external attacker may attempt to compromise the assets without being detected. This threat includes a threat agent causing audit records to be lost, deleted or prevent future records from being

Threat	Description
	recorded by taking actions to exhaust audit storage capacity, thus masking an attacker's actions.
T.BRIDGE	An external attacker could intercept or manipulate user and TSF data sent between two DSS via the DSB.

3.2 Organisational security policies

The following organisational security policies are enforced by the TOE and the TOE environment.

OSP	Description
OSP.MANAGE	The TOE shall provide the authorized administrators with the means to manage the TSFs and the DCAs associated with the TOE installation.
OSP.SERVICE	The TOE shall provide the authorized secure service access to manage the TSFs and the TOE installation.
OSP.ACCOUNT	Administrators shall be accountable for the actions they conduct by generating and maintaining sufficient audit records for the actions.
OSP.PROVISIONING	The TOE must provide a secure provisioning process that can be used for any remote users without access to the secure local network.
OSP.CA	The TOE must be able to generate it's own private-public keys and generate its own certificates as well as sign certificates for DCAs.
OSP.CLIENTKEY	The TOE must be able to distribute previously encrypted keys to authenticated clients on request, to support data-at-rest protection for the DCA.
OSP.TUNNEL	The TOE must provide a tunnelling service to enable voice or video communication to the DCA over TCP by tunnelling the connection over TLS 1.2.

3.3 Assumptions

This section specifies the assumptions on the TOE environment that are necessary for the TOE to meet its security objectives.

Assumption	Description
A.NETWORK	It is assumed that the underlying hardware of the TOE and local network is dedicated to the TOE usage and function.
A.NOEVIL	It is assumed that administrators are given privileges they are authorized for, and that they are competent, non-hostile and follow all their guidance; however, they are capable of error.
A.PHYSICAL	The TOE is physically protected, i.e. no unauthorised persons have physical access to the TOE and its underlying system. This includes the administrators that only can access the TOE via the local network or through a trusted VPN connection.
A.REVIEW	It is assumed that audit trails are regularly analysed for misuse and security incidents.
A.TIME	It is assumed that the IT environment will provide a reliable time

Assumption	Description
	source to the TOE and the TOE environment.
A.WORKSTATION	It is assumed that administrators are performing administration from computers that are well-configured, located in a secure environment and are not exposed to other users or potential attackers.
A.LINK	It is assumed the link, possibly encoded as a QR code, used for provisioning is provided to the correct DCA user and not being disclosed to anyone else.
A.TRUSTANCHOR	It is assumed that a trust anchor is provided and will be used for TLS connections by the DCAs, administrator browsers and other DSS for validation of TOE certificates when connecting to the TOE.
A.USER	It is assumed that the DCA users are trustworthy and trained to perform their actions in accordance with their instructions and security policies.
A.FIREWALL	It is assumed that the IT environment provides a firewall or other suitable means to protect the TOE from untrusted networks.
A.CROSS-SYSTEM	It is assumed that separate DSS which the TOE connects to via the DSB to enable cross-system functionality are trusted and operated in a secure manner.

4 Security objectives

The security objectives provide a concise statement of the intended response to the security problem. It will describe which security needs will be addressed by the TOE and which will be addressed by the TOE environment, in the form of a statement of security objectives.

4.1 Security objectives for the TOE

The following are the security objectives to be met by the TOE.

Security Objective	Description
O.ACCESS	The TOE must ensure that administrators only can access information and functions that they are explicitly authorized for.
O.AUDIT	The TOE must be able to provide audit evidence of security relevant events as well as authorised use of security management functions to allow identification of security violations attempts as well as maintain accountability of administrators.
O.CA	The TOE must be able to generate it's own private-public keys and generate its own certificates as well as sign certificates for DCA clients.
O.CHANNEL	The TOE must provide mutually authenticated and trusted channels to any outside components to protect information transmitted to and received from such components against unauthorised disclosure and to detect any modification of incoming information transmitted from such components, and to provide the means for such components to verify the integrity of information transmitted out of the TOE.
O.MANAGE	The TOE shall provide the authorized administrators with the means to manage the TSF and the DCA applications associated with the TOE installation.
O.PROVISIONING	The TOE must provide an unpredictable link for one-time registration, ensuring that such a link is only available for a very limited time to limit the window of opportunity in case of no or late use of activation.
O.REMOTE	The TOE must uniquely identify and authenticate administrators and provide them with a secure communication channel before allowing administrators any access to the TOE.
O.REVIEW	The TOE must provide an authorised administrator and only the authorised administrator with ability to read the audit trail.
O.SERVICE	The TOE must provide the authorized secure service access to manage the TSFs and the TOE installation.
O.CLIENTKEY	The TOE must be able to distribute previously encrypted keys to authenticated clients on request, to support data-at-rest protection for the DCA.
O.PUSH	The TOE must be able to encrypt Push Notifications to be sent via the mobile operating system vendor's systems, so that only the recipient DCA can decrypt them.
O.BRIDGE	The TOE must provide a mutually authenticated and secure channel between the TOE and another DSS which are connected via the DSB.
O.TUNNEL	The TOE must provide a tunnelling service to enable DCA voice or video communication over TCP by tunnelling the connection over TLS

Security Objective	Description
	1.2.

4.2 Security objectives for the TOE environment

The following are the security objectives to be met by the TOE environment.

Security Objective	Description
OE.LINK	The TOE environment must ensure that the link, possibly encoded into a QR code, used for provisioning is provided to the correct DCA user and not being disclosed to anyone else.
OE.NETWORK	The TOE environment must ensure that the underlying hardware of the TOE and local network is dedicated to the TOE usage, functions and physically protected.
OE.NOEVIL	The TOE environment must ensure administrators are given privileges they are authorized for, and that they are competent, non-hostile and follow all their guidance; however, they are capable of error.
OE.PHYSICAL	The TOE environment must ensure that the TOE is physically protected, i.e. no unauthorised persons have physical access to the TOE and its underlying system. This includes the administrators that only can access the TOE via the local network or through a trusted VPN connection.
OE.REVIEW	The TOE environment must ensure that audit trails are regularly analysed for misuse and security incidents.
OE.TIME	The TOE environment must ensure that the IT environment will provide a reliable time source to the TOE and the TOE environment.
OE.WORKSTATION	The TOE environment must ensure that administrators are performing administration from computers that are well-configured, located in a secure environment and are not exposed to other users or potential attackers.
OE.TRUSTANCHOR	The TOE environment must ensure that a trust anchor is provided and will be used for TLS connections by DCA clients, administrator browsers and other DSS for validation of TOE certificates when connecting to the TOE.
OE.USER	The operational environment shall ensure that DCA users are trustworthy and trained to perform their actions in accordance with their instructions and security policies.
OE.FIREWALL	The TOE environment shall provide a firewall or other suitable means to protect the TOE from untrusted networks.
OE.CROSS-SYSTEM	The TOE environment shall ensure that separate DSS which the TOE connects to via the DSB to enable cross-system functionality are trusted and operated in a secure manner.

4.3 Security objectives rationale

4.3.1 Security objectives completeness

The following tables provide a mapping of security objectives both for the TOE and the TOE environment to the environment defined by the threats, policies and assumptions, illustrating that each security objective for the TOE covers at least one threat or policy, and that each security objective for the TOE environment covers at least one policy, threat or assumption.

	T.COMMUNICATION	T.MASQUERADE	T.UNAUTH	T.UNDETECTED	T.BRIDGE	OSP.MANAGE	OSP.ACCOUNT	OSP.PROVISIONING	OSP.SERVICE	OSP.CA	OSP.CLIENTKEY	OSP.TUNNEL	A.NETWORK	A.NOEVIL	A.PHYSICAL	A.REVIEW	A.TIME	A.WORKSTATION	A.LINK	A.TRUSTANCHOR	A.USER	A.FIREWALL	A.CROSS-SYSTEM
O.ACCESS			X																				
O.AUDIT				X			X																
O.CA										X													
O.CHANNEL	X	X																					
O.MANAGE						X																	
O.PROVISIONING		X						X															
O.REMOTE			X																				
O.REVIEW				X			X																
O.SERVICE									X														
O.CLIENTKEY											X												
O.TUNNEL												X											
O.PUSH	X																						
O.BRIDGE					X																		
OE.LINK		X						X											X				
OE.NETWORK													X										
OE.NOEVIL														X									
OE.PHYSICAL															X								
OE.REVIEW				X			X									X							
OE.TIME				X			X										X						
OE.WORKSTATION																		X					
OE.TRUSTANCHOR																				X			
OE.USER																					X		
OE.FIREWALL																						X	
OE.CROSS-SYSTEM																							X

4.3.2 Security objectives sufficiency

The following rationale provides justification that the security objectives are suitable to counter each individual threat and that each security objective tracing back to a threat actually contributes to the mitigation of that threat.

Threat	Rationale for the security objectives
T.COMMUNICATION	This threat is addressed by O.CHANNEL that ensures that there is a trusted path between the TOE and external components ensuring authenticity, confidentiality and integrity of any TSF or user data transmitted between the TOE and external components, such as phonebook updates and SIP connection data. O.PUSH ensures that Push Notifications containing user data sent from the TOE to the DCA are encrypted. These are transmitted via the mobile device operating system vendor.
T.MASQUERADE	This threat is address by O.CHANNEL that ensures that external components must authenticate before any information is passed or given access to, such as phonebook updates and SIP connection data. For the provisioning, there is no identification of the client side to the TOE. O.PROVISIONING ensures that the link is unpredictable, is available for a limited time and can only be used once, also under assumption that the link is provided in a secure way to the end user for which the provisioning applies (OE.LINK).
T.UNAUTH	This threat is address by O.REMOTE that ensure that all administrators will have to identify and authenticate before gaining administrator access to the TOE, and by O.ACCESS that ensures that administrators only can access information and functions that they are explicitly authorized for.
T.UNDETECTED	This threat is address by O.AUDIT and O.REVIEW that ensure that security relevant events are being audited and that they can be reviewed by an authorized administrator, and only by an authorised administrator. The O.AUDIT is supported by OE.TIME providing a secure time stamp, while OE.REVIEW ensures that audit logs are regularly reviewed.
T.BRIDGE	This threat is addressed by O.BRDIGE that ensures there is a trusted path between the TOE and a different DSS, ensuring authenticity, confidentiality and integrity of all user and TSF data transmitted between the systems.

The following rationale provides justification that the security objectives of the TOE and the TOE environment are suitable to address each individual OSP and that each security objective tracing back to a OSP actually contributes in addressing the OSP.

OSP	Rationale for the security objectives
OSP.ACCOUNT	This OSP is addressed by O.AUDIT and O.REVIEW that ensure that any administrator actions are being audited and that they can be reviewed by an authorized administrator, and only by an authorised administrator. The O.AUDIT is supported by OE.TIME providing a secure time stamp, while OE.REVIEW ensures that audit logs are regularly reviewed.
OSP.MANAGE	This OSP is addressed by O.MANAGE that ensures that the TOE provides the necessary management functions for managing the TSFs and the DCA associated with the TOE installation.
OSP.SERVICE	This OSP is addressed by O.SERVICE that ensures that the TOE provides a secure channel to the service functions of the TOE to manage the TSFs and the TOE installation.
OSP.PROVISIONING	This OSP is addressed by O.PROVISIONING by providing an unpredictable link for one-time registration and ensuring that such a link is only available

OSP	Rationale for the security objectives
	for a very limited time. The link is then provided to the DCA user in a secure way as part of the TOE environment (OE.LINK).
OSP.CA	This OSP is addressed by O.CA that ensure that the TOE is able to generate it's own private-public keys and generate its own certificates as well as sign certificates for DCA clients.
OSP.CLIENTKEY	This OSP is addressed by O.CLIENTKEY that ensures the TOE can correctly distribute the DCA keys stored by the TOE.
OSP.TUNNEL	This OSP is addressed by O.TUNNEL that ensures the TOE can provide a TLS 1.2 tunnel server.

The following rationale provides justification that the security objectives of the TOE environment are suitable to address each individual assumption and that each security objective tracing back to an assumption actually contributes in addressing the assumption.

Assumption	Rationale for the security objectives
A.NETWORK	Addressed by OE.NETWORK, which is identical to the assumption
A.NOEVIL	Addressed by OE.NOEVIL, which is identical to the assumption
A.PHYSICAL	Addressed by OE.PHYSICAL, which is identical to the assumption
A.REVIEW	Addressed by OE.REVIEW, which is identical to the assumption
A.TIME	Addressed by OE.TIME, which is identical to the assumption
A.WORKSTATION	Addressed by OE.WORKSTATION, which is identical to the assumption
A.LINK	Addressed by OE.LINK, which is identical to the assumption
A.TRUSTANCHOR	Addressed by OE.TRUSTANCHOR, which is identical to the assumption
A.USER	Addressed by OE.USER, which is identical to the assumption
A.FIREWALL	Addressed by OE.FIREWALL, which is identical to the assumption
A.CROSS-SYSTEM	Addressed by OE.CROSS-SYSTEM, which is identical to the assumption.

5 Extended components definition

The extended requirements are used to specify TLS for clients and servers. A TOE that implements TLS must in addition to FTP_ITC.1 or FTP_TRP.1 also specify the TLS protocol that is implemented. This is done in FCS_TLSS_EXT.1, FCS_TLSS_EXT.2 and FCS_TLSC_EXT.2 (for cryptography) and FCS_RNG.1 (for the random number generation). The HTTPS protocol itself is specified in FCS_HTTPS_EXT.1.

The extended components defined in this Security Target have been copied from [cPPND] and [RNGfc]. They are re-iterated herein exactly as stated in the reference with no changes made.

5.1 Cryptographic Support (FCS)

5.1.1 Cryptographic Protocols (FCS_TLSS_EXT, FCS_TLSC_EXT, FCS_SSHS_EXT, FCS_HTTPS_EXT)

5.1.1.1 FCS_HTTPS_EXT – HTTPS Protocol

Family Behaviour

Components in this family define the requirements for protecting remote management sessions between the TOE and a Security Administrator. This family describes how HTTPS will be implemented. This is a new family defined for the FCS Class.

Component Levelling

FCS_HTTPS_EXT.1 HTTPS requires that HTTPS be implemented according to RFC 2818 and supports TLS.

Management: FCS_HTTPS_EXT.1

The following actions could be considered for the management functions in FMT: a) There are no management activities foreseen.

Audit: FCS_HTTPS_EXT.1

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

a) There are no auditable events foreseen.

5.1.1.1.1 FCS_HTTPS_EXT.1 – HTTPS Protocol

Hierarchical to: No other components

Dependencies: [FCS_TLSC_EXT.1 TLS Client Protocol, or
FCS_TLSS_EXT.1 TLS Server Protocol]

FCS_HTTPS_EXT.1.1 The TSF shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2 The TSF shall implement the HTTPS protocol using TLS.

FCS_HTTPS_EXT.1.3 If a peer certificate is presented, the TSF shall [selection: *not establish the connection*, *request authorization to establish the connection*, *[assignment: other action]*] if the peer certificate is deemed invalid.

5.1.1.2 FCS_SSHS_EXT – SSH Server Protocol

Family Behaviour

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

Component levelling

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

Management: FCS_SSHS_EXT.1

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

a) There are no management activities foreseen.

Audit: FCS_SSHS_EXT.1

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

a) Failure of SSH session establishment

b) SSH session establishment

c) SSH session termination

5.1.1.2.1 FCS_SSHS_EXT.1 – SSH Server Protocol

Hierarchical to: No other components

Dependencies: FCS_CKM.1Cryptographic Key Generation

FCS_CKM.2 Cryptographic Key Establishment

FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption)

FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)

FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)

FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)

FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption)

FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)

FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)

FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)

FCS_RBG_EXT.1 Random Bit Generation

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFC(s) [selection: *4251*, *4252*, *4253*, *4254*, *5647*, *5656*, *6187*, *6668*, *8332]*

Application Note

The ST author selects which of the RFCs to which conformance is being claimed. Note that these need to be consistent with selections in later elements of this component (e.g., cryptographic algorithms permitted). RFC 4253 indicates that certain cryptographic algorithms are "REQUIRED". This means that the implementation must include support, not that the algorithms must be enabled for use. Ensuring that algorithms indicated as "REQUIRED" but not listed in the later elements of this component are implemented is out of scope of the evaluation activity for this requirement. RFC 5647 only applies to the RFC compliant implementation of GCM; a TOE that only implements the "@openssh.com" variant of GCM should not select 5647. aes-gcm@openssh.com is specified in Section 1.6 of the OpenSSH Protocol Specification (https://cvsweb.openbsd.org/cgi-bin/cvsweb/src/usr.bin/ssh/PROTOCOL?rev=1.31).*

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

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

Application Note

RFC 4253 provides for the acceptance of "large packets" with the caveat that the packets should be of "reasonable length" or dropped. The assignment should be filled in by the ST author with the maximum packet size accepted, thus defining "reasonable length" for the TOE.

FCS_SSHS_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [assignment: *encryption algorithms*].

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [selection: *ssh-rsa*, *rsa-sha2-256*, *rsa-sha2-512*, *ecdsa-sha2-nistp256*, *x509v3-ssh-rsa*, *ecdsa-sha2-nistp384*, *ecdsa-sha2-nistp521*, *x509v3-ecdsa-sha2-nistp256*, *x509v3-ecdsa-sha2-nistp384*, *x509v3-ecdsa-sha2-nistp521*, *x509v3-rsa2048-sha256*] as its public key algorithm(s) and rejects all other public key algorithms.

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

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

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

Application Note

This SFR defines two thresholds - one for the maximum time span the same session keys can be used and the other one for the maximum amount of data that can be transmitted using the same session keys. Both thresholds need to be implemented and a rekey needs to be performed on whichever threshold is reached first. For the maximum transmitted data threshold, the total incoming and outgoing data needs to be counted. The rekey applies to all session keys (encryption, integrity protection) for incoming and outgoing traffic.

It is acceptable for a TOE to implement lower thresholds than the maximum values defined in the SFR.

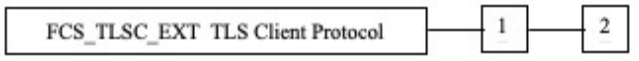
For any configurable threshold related to this requirement the guidance documentation needs to specify how the threshold can be configured. The allowed values must either be specified in the guidance documentation and must be lower or equal to the thresholds specified in this SFR or the TOE must not accept values beyond the thresholds specified in this SFR.

5.1.1.3 FCS_TLSC_EXT – TLS Client Protocol

Family Behaviour

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

Component levelling



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

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

Management: FCS_TLSC_EXT.1, FCS_TLSC_EXT.2

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

a) There are no management activities foreseen.

Audit: FCS_TLSC_EXT.1, FCS_TLSC_EXT.2

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

a) Failure of TLS session establishment

b) TLS session establishment

c) TLS session termination

5.1.1.3.1 FCS_TLSC_EXT.1 – TLS Client Protocol

Hierarchical to: No other components

Dependencies: FCS_CKM. 1 Cryptographic Key Generation

FCS_CKM.2 Cryptographic Key Establishment

FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption)

FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)

FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)

FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)

FCS_TLSC_EXT.1.1 The TSF shall implement [selection: *TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)*] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites:

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

Application Note

The ciphersuites to be tested in the evaluated configuration are limited by this requirement.

FCS_TLSC_EXT.1.2 The TSF shall verify that the presented identifier matches the reference identifier per RFC 6125 section 6.

Application Note

The rules for verification of identify are described in Section 6 of RFC 6125. The reference identifier is established by the user (e.g. entering a URL into a web browser or clicking a link), by configuration (e.g. configuring the name of a mail server or authentication server), or by an application (e.g. a parameter of an API) depending on the application service. Based on a singular reference identifier's source domain and application service type (e.g. HTTP, SIP, LDAP), the client establishes all reference identifiers which are acceptable, such as a Common Name for the Subject Name field of the certificate and a (case-insensitive) DNS name, URI name, and Service Name for the Subject Alternative Name field. The client then compares this list of all acceptable reference identifiers to the presented identifiers in the TLS server's certificate.

The preferred method for verification is the Subject Alternative Name using DNS names, URI names, or Service Names. Verification using the Common Name is required for the purposes of backwards compatibility. Additionally, support for use of IP addresses in the Subject Name or Subject Alternative name is discouraged as against best practices but may be implemented.

Finally, the client should avoid constructing reference identifiers using wildcards. However, if the presented identifiers include wildcards, the client must follow the best practices regarding matching; these best practices are captured in the evaluation activity.

FCS_TLSC_EXT.1.3 When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the server certificate is invalid. The TSF shall also [selection:

- *Not implement any administrator override mechanism*
- *require administrator authorization to establish the connection if the TSF fails to [selection: match the reference identifier, validate certificate path, validate expiration date, determine the revocation status] of the presented server certificate*

].

FCS_TLSC_EXT.1.4 The TSF shall [selection: *not present the Supported Elliptic Curves Extension, present the Supported Elliptic Curves Extension with the following NIST curves: [selection: secp256r1, secp384r1, secp521r1] and no other curves*] in the Client Hello.

Application Note

If ciphersuites with elliptic curves were selected in FCS_TLSC_EXT.1.1, a selection of one or more curves is required. If no ciphersuites with elliptic curves were selected in FCS_TLS_EXT.1.1, then "not present the Supported Elliptic Curves Extension" should be selected.

This requirement limits the elliptic curves allowed for authentication and key agreement to the NIST curves from FCS_COP.1/SigGen and FCS_CKM.1 and FCS_CKM.2. This extension is required for clients supporting Elliptic Curve ciphersuites.

5.1.1.3.2 FCS_TLSC_EXT.2 – TLS Client Protocol with Authentication

Hierarchical to: FCS_TLSC_EXT.1 TLS Client Protocol

Dependencies: FCS_CKM.1Cryptographic Key Generation

FCS_CKM.2 Cryptographic Key Establishment

FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption)

FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)

FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)

FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)

FCS_RBG_EXT.1 Random Bit Generation

FCS_TLSC_EXT.2.1 The TSF shall implement [selection: *TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)*] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites:

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

Application Note

The ciphersuites to be tested in the evaluated configuration are limited by this requirement.

FCS_TLSC_EXT.2.2 The TSF shall verify that the presented identifier matches the reference identifier per RFC 6125 section 6.

Application Note

The rules for verification of identify are described in Section 6 of RFC 6125. The reference identifier is established by the user (e.g. entering a URL into a web browser or clicking a link), by configuration (e.g. configuring the name of a mail server or authentication server), or by an

application (e.g. a parameter of an API) depending on the application service. Based on a singular reference identifier's source domain and application service type (e.g. HTTP, SIP, LDAP), the client establishes all reference identifiers which are acceptable, such as a Common Name for the Subject Name field of the certificate and a (case-insensitive) DNS name, URI name, and Service Name for the Subject Alternative Name field. The client then compares this list of all acceptable reference identifiers to the presented identifiers in the TLS server's certificate.

The preferred method for verification is the Subject Alternative Name using DNS names, URI names, or Service Names. Verification using the Common Name is required for the purposes of backwards compatibility. Additionally, support for use of IP addresses in the Subject Name or Subject Alternative name is discouraged as against best practices but may be implemented. Finally, the client should avoid constructing reference identifiers using wildcards. However, if the presented identifiers include wildcards, the client must follow the best practices regarding matching; these best practices are captured in the evaluation activity.

FCS_TLSC_EXT.2.3 When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the server certificate is invalid. The TSF shall also [selection:

- *Not implement any administrator override mechanism*
- *require administrator authorization to establish the connection if the TSF fails to [selection: match the reference identifier, validate certificate path, validate expiration date, determine the revocation status] of the presented server certificate*

].

FCS_TLSC_EXT.2.4 The TSF shall [selection: *not present the Supported Elliptic Curves Extension, present the Supported Elliptic Curves Extension with the following NIST curves: [selection: secp256r1, secp384r1, secp521r1] and no other curves*] in the Client Hello.

Application Note

If ciphersuites with elliptic curves were selected in FCS_TLSC_EXT.1.1, a selection of one or more curves is required. If no ciphersuites with elliptic curves were selected in

FCS_TLS_EXT.1.1, then "not present the Supported Elliptic Curves Extension" should be selected.

This requirement limits the elliptic curves allowed for authentication and key agreement to the NIST curves from FCS_COP.1/SigGen and FCS_CKM.1 and FCS_CKM.2. This extension is required for clients supporting Elliptic Curve ciphersuites.

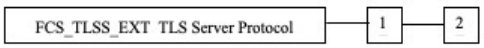
FCS_TLSC_EXT.2.5 The TSF shall support mutual authentication using X.509v3 certificates.

5.1.1.4 FCS_TLSS_EXT – TLS Server Protocol

Family Behaviour

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

Component levelling



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

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

Management: FCS_TLSS_EXT.1, FCS_TLSS_EXT.2

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

a) There are no management activities foreseen.

Audit: FCS_TLSS_EXT.1, FCS_TLSS_EXT.2

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

a) Failure of TLS session establishment

b) TLS session establishment

c) TLS session termination

5.1.1.4.1 FCS_TLSS_EXT.1 – TLS Server Protocol

Hierarchical to: No other components

Dependencies: FCS_CKM.1 Cryptographic Key Generation

FCS_CKM.2 Cryptographic Key Establishment

FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption)

FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)

FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)

FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)

FCS_RBG_EXT.1 Random Bit Generation

FCS_TLSS_EXT.1.1 The TSF shall implement [selection: *TLS 1.2 (RFC 5246)*, *TLS 1.1 (RFC 4346)*] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites:

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

Application Note

The ciphersuites to be tested in the evaluated configuration are limited by this requirement.

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

Application Note

All SSL versions and TLS v1.0 are denied. Any TLS versions not selected in FCS_TLSS_EXT.1.1 should be selected here. (If "none" is the selection for this element then the ST author may omit the words "and none".)

FCS_TLSS_EXT.1.3 The TSF shall [selection: p*erform RSA key establishment with key size [selection: 2048 bits*, *3072 bits*, *4096 bits]; generate EC Diffie-Hellman parameters over NIST curves [selection: secp256r1*, *secp384r1*, *secp521r1] and no other curves; generate Diffie-Hellman parameters of size [selection: 2048 bits*, *3072 bits]*].

Application Note

The assignments will be filled in based on the assignments performed in FCS_TLSS_EXT.1.1.

5.1.1.4.2 FCS_TLSS_EXT.2 – TLS Server Protocol with mutual authentication

Hierarchical to: FCS_TLSS_EXT.1

Dependencies: FCS_CKM.1 Cryptographic Key Generation

FCS_CKM.2 Cryptographic Key Establishment

FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption)

FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)

FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)

FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)

FCS_RBG_EXT.1 Random Bit Generation

FCS_TLSS_EXT.2.1 The TSF shall implement [selection: *TLS 1.2 (RFC 5246)*, *TLS 1.1 (RFC 4346)*] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites:

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

Application Note

The ciphersuites to be tested in the evaluated configuration are limited by this requirement.

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

Application Note

All SSL versions and TLS v1.0 are denied. Any TLS versions not selected in FCS_TLSS_EXT.2.1 should be selected here. (If "none" is the selection for this element then the ST author may omit the words "and none".)

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

Application Note

The assignments will be filled in based on the assignments performed in FCS_TLSS_EXT.2.1.

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

Application Note

The use of X.509v3 certificates for TLS is addressed in FIA_X509_EXT.2.1. This requirement adds that this use must include support for client-side certificates for TLS mutual authentication.

FCS_TLSS_EXT.2.5 When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the client certificate is invalid. The TSF shall also [selection:

- *Not implement any administrator override mechanism*
- *require administrator authorization to establish the connection if the TSF fails to [selection: match the reference identifier, validate certificate path, validate expiration date, determine the revocation status] of the presented client certificate*

].

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

Application Note

This requirement only applies to those TOEs performing mutually-authenticated TLS (FCS_TLSS_EXT.2.4). The peer identifier may be in the Subject field or the Subject Alternative Name extension of the certificate. The expected identifier may either be configured, may be compared to the Domain Name, IP address, username, or email address used by the peer, or may be passed to a directory server for comparison.

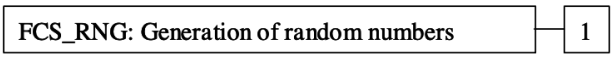
5.1.2 Generation of random numbers (FCS_RNG)

5.1.2.1 FCS_RNG – Generation of random numbers

Family Behaviour

This family defines quality requirements for the generation of random numbers that are intended to be used for cryptographic purposes.

Component levelling:



Management: FCS_RNG.1

a) There are no management activities foreseen.

Audit: FCS_RNG.1

a) There are no actions defined to be auditable.

5.1.2.1.1 FCS_RNG.1 – Random number generation

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1 The TSF shall provide a [selection: *physical*, *non-physical true*, *deterministic*, *hybrid physical*, *hybrid deterministic*] random number generator that implements: [assignment: *list of security capabilities*].

FCS_RNG.1.2 - The TSF shall provide random numbers that meet [assignment: *a defined quality metric*].

6 Security requirements

6.1 Security functional policy

6.1.1 Client SFP

The TOE must implement a functional policy for keys imported from the DCA over the encrypted TLS channel. The TOE must be able to receive these keys, after which they must be associated with the DCA that they were received from and stored in the database. These keys are received through a mutually authenticated TLS channel, where the DCA is clearly identified. This does not involve any information flow or access control policy, and the keys are received securely from a trusted source.

6.2 Security functional requirements

The following convention is used for operations applied to the Security Functional Requirements: Assignment and selection are indicated by **bold**. Refinements are indicated by **bold underscore** for additions and by ~~**bold strike through**~~ for deletions. Iterations are indicated by appending a letter to the requirement, e.g. FCS_COP.1a.

6.2.1 FAU_GEN.1 – Audit data generation

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

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

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

c)

- **DCC:**
  - Install root certificate on a DCM
  - Initialize a DCM by requesting a CSR
  - Install intermediate certificate on a DCM
  - Install external certificate on any server
  - Adding Apple push certificates
  - Removing Apple push certificates
  - Updating Apple push certificates
  - Login attempt
  - Set SMS and email credentials for DPS
  - Set configuration on a server
  - Remove server from DCC
  - Add server to DCC
  - Changing IP and apikey
  - Changing IP
  - Changing apikey
  - Editing scheduled configuration in system settings
  - Editing features in system settings
  - Importing Excel file for user management
  - Adding user to a group
  - Remove user from a group
  - Send email invitation
  - Send sms invitation
  - Create company
  - Edit a company's logo
  - Edit a company's name

- Delete a company
- Allow a company to add users to a group
- Remove permission for a company to add users to a group
- Create group
- Edit a group's name
- Delete group
- Create department
- Edit a department's name
- Delete department
- Delete user
- Edit user (excluding image)
- Edit user's image
- Create user
- TLS connection attempt
- SSH session attempts
- SSH session termination
- Add or remove emergency contacts list
- Modify emergency contacts list
- configure emergency contacts list for user
- Send user push notification
- Change link between groups
- Provision a user with a QR code
- Add or remove a new bridge connection
- Change the maintenance mode of a server
- Create or delete standard messages
- Modify standard messages

- **DCS:**
  - TLS connection attempt
  - SSH session attempt
  - SSH session termination
- **DPS:**
  - TLS connection attempt
  - Use of provisioning one-time link
  - SSH session attempt
  - SSH session termination
- **DCM:**
  - Issue certificates
  - SSH session attempt
  - SSH session termination
- **DDB:**
  - SSH session attempt
  - SSH session termination
- **DSB:**
  - SSH session attempt
  - SSH session termination
  - TLS connection attempt

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

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the ~~**PP/**~~ST, **no other information**.

6.2.2 FAU_SAR.1 – Audit review

FAU_SAR.1.1 The TSF shall provide **Company Admin, System Admin and Service Access** with the capability to read **all information** from the audit records.

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

6.2.3 FAU_SAR.2 – Restricted audit review

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

Application note: Read access to the audit trail is limited to the Company Admin, System Admin and Service Access roles (the System Admin role is a subset of the Service Access role).

6.2.4 FAU_STG.2 – Guarantees of audit data availability

FAU_STG.2.1 The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

FAU_STG.2.2 The TSF shall be able to **prevent** unauthorised modifications to the stored audit records in the audit trail.

FAU_STG.2.3 The TSF shall ensure that **all events for the last period of** stored audit records will be maintained when the following conditions occur: **audit storage exhaustion**.

Application note: The TOE prevents any deletion of audit records. Audit records can only be deleted outside of the TOE control by the system administrators of the TOE environment. All audit events will be kept for a specified time, but older events will be overwritten to ensure that audit is never exhausted. The time is configurable during installation of the TOE and is typical set between three months and one year.

6.2.5 FCS_CKM.1a – Cryptographic key generation (RSA keypair)

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

- **RSA schemes using cryptographic key sizes of ~~2048-bit or greater~~ 4096 bit that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3.**

Application note: This SFR is there because of the RSA kay pair generation for the certificates for the servers that are facing the outside TLS connections (DPS, DCS and DSB) and for the DCC for the administrator access. This requirement is a refinement of FCS_CKM.1 defined in [cPPND].

6.2.6 FCS_CKM.1b – Cryptographic key generation (AES)

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **AES-256 in the Galois/Counter Mode (GCM)** and specified cryptographic key sizes **256 bit (AES-256)** that meet the following: **[FIPS197] and [NIST SP 800-38D]**.

Application note: This is the generation of AES keys (session keys) for the TLS connection.

6.2.7 FCS_CKM.2b – Cryptographic key distribution (TLS public key)

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **TLS 1.2** that meets the following: **RFC5246**.

Application note: The server system will send its certificate to the DCA or the administrator browser during the TLS handshake. This aims to cover the key establishment of a TLS session.

6.2.8 FCS_CKM.2c – Cryptographic key distribution (DCA Storage encryption key)

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method:

- **The TOE distributes an encrypted Storage Encryption Key for the DCA. The encrypted Key is transmitted via the trusted TLS channel to its associated DCA recipient after the DCA has successfully authenticated.**

that meets the following: **no standard**.

Application note: The server system will send the encrypted key to the DCA via the trusted TLS channel as described by FTP_ITC.1. The DCA only stores the Key Encryption Key (KEK) used to encrypt the Storage Encryption Key. The Storage Encryption Key is originally generated and transmitted to the TOE by the DCA.

6.2.9 FCS_CKM.2d – Cryptographic key distribution (SSH public key)

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **SSH** that meets the following: **RFC4253**.

Application note: The server system will send its SSH keys to the service administrator during the SSH handshake. This aims to cover the key establishment of an SSH session.

6.2.10 FCS_CKM.4 – Cryptographic key destruction

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **zeroization** that meets the following: **no standard**.

Application note: Key destruction is performed of all symmetric keys that are generated by the TOE and used for data encryption and decryption.

6.2.11 FCS_COP.1a – Cryptographic Operation (AES)

FCS_COP.1.1 The TSF shall perform **decryption and encryption** in accordance with a specified cryptographic algorithm **AES-256 in GCM mode** and cryptographic key sizes **256 bit** that meet the following: **[FIPS197] and [NIST SP 800-38D]**.

Application note: This requirement addresses the data stream encryption and decryption of the TLS connections between the TOE and other parties.

6.2.12 FCS_COP.1b – Cryptographic Operation (Signature Verification and Generation)

FCS_COP.1.1 The TSF shall perform **digital signature verification and generation** in accordance with a specified cryptographic algorithm **RSA Digital Signature Algorithm** and cryptographic key sizes **3072 bit and 4096 bit** that meet the following: **FIPS PUB 186-4 "Digital Signature Standard (DSS)" Section 5.5 using PKCS #1 v2.1 Signature Scheme RSASSA-PKCS1-v1.5 [FIPS186-4][PKCS1v2.1]**.

Application note: This requirement addresses the RSA digital signature verification and generation performed as part of the TLS client and server authentication performed by the TOE. This also addresses the TOE verification of the signature on the certificate request (CSR) from the client and server components. The DCA generated certificate are 3072 bits. The server certificates verified for the DSB are 4096 bits.

6.2.13 FCS_COP.1c – Cryptographic Operation (Hashing)

FCS_COP.1.1 The TSF shall perform **secure hash** in accordance with a specified cryptographic algorithm **SHA384 and SHA512** ~~**and cryptographic key sizes**~~ that meet the following: **ISO/IEC 10118-3:2018**.

Application note: The secure hash is used by FCS_TLSS_EXT.2, FCS_TLSS_EXT.1, FCS_TLSC_EXT.2 and FCS_SSHS_EXT.1 to ensure the integrity of the connections.

6.2.14 FCS_COP.1d – Cryptographic Operation (Certificate signing)

FCS_COP.1.1 The TSF shall perform **digital signature signing** in accordance with a specified cryptographic algorithm **RSA [RSASSA-PKCS1-v1_5]** and cryptographic key sizes **4096 bit** that meet the following: **[PKCS1v2.1]**.

Application note: This requirement addresses the RSA digital signing of certificates as part of the certificate generation both for the DCA and for the TOE components (server system) itself. The client generates the private/public key pair, creates a CSR with the public key and sends the CSR to the DCM via the DCS. The DCM signs the CSR if permitted and returns an X.509v3 client certificate to the DCA. It uses Python.

6.2.15 FCS_COP.1e – Cryptographic Operation (Push encryption)

FCS_COP.1.1 The TSF shall perform **encryption** in accordance with a specified cryptographic algorithm **AES-256 in CFB mode** and cryptographic key sizes **256 bit** that meet the following: **[FIPS197] and [NIST SP 800-38A]**.

Application note: This requirement addresses the encryption of Push Notifications to be sent to DCA clients through the mobile OS vendor's systems. The key used is generated by the DCA and uploaded to the server at the end of provisioning.

6.2.16 FCS_COP.1f – Cryptographic Operation (Keyed-hash)

FCS_COP.1.1 The TSF shall perform **keyed-hash message authentication** in accordance with a specified cryptographic algorithm **HMAC-SHA-384 and HMAC-SHA-512** and cryptographic key sizes **384 and 512 bit** that meet the following: **[ISO9797]**

Application note: The keyed-hash message authentication is used to ensure the message integrity of the trusted channels.

6.2.17 FCS_RNG.1 – Random Number Generation

FCS_RNG.1.1 The TSF shall provide a **deterministic** random number generator that implements:

a) DRG2.1: If initialized with a random seed using high-resolution time stamps of block device access events, human interface device events and interrupt events as seed source, the internal state of the RNG shall have a minimum entropy of 48 bits.

b) DRG2.2: The DRNG provides forward secrecy.

c) DRG2.3: The DRNG provides backward secrecy.

FCS_RNG.1.2 The TSF shall provide random numbers that meet:

a) DRG.2.4: The RNG initialized with a random seed every time a random number is obtained that is equal in size as the generated random number generates output for which 219 strings of bit length 128 are mutually different with probability of greater than 1-2**-10.**

b) DRG.2.5: Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A.

Application note: This requirement addresses the RNG for key generation for the public-private key pair, the symmetric AES key (TLS session key) and the one-time key (web link) used for provisioning.

6.2.18 FCS_SSHS_EXT.1 – SSH Server Protocol

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs **4251**, **4252**, **4253**, **4254**, **5647, 5656, 6187, 6668**.

Application note: The SSH implemented is SSH-2 only. SSH-1 has inherent design flaws which makes it vulnerable, it is now generally considered obsolete is therefore avoided also disabling fallback from SSH-2 to SSH-1.

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

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

Application note: The RFC 4253 provides for the acceptance of "large packets" with the caveat that the packets should be of "reasonable length" or dropped. The assignment defines the maximum packet size accepted, thus defining "reasonable length" for the TOE.

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

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses **ssh-rsa** as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses **hmac-sha2-512** as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7 The TSF shall ensure that **ecdh-sha2-nistp384** and **no other methods** are the only allowed key exchange methods used for the SSH protocol.

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

Application note: The SSH connection is only used for the service administrator access to the TOE.

6.2.19 FCS_TLSS_EXT.1 – TLS Server Protocol (Unauthenticated)

FCS_TLSS_EXT.1.1 The TSF shall implement **TLS 1.2 (RFC 5246)** and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites:

- **TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289**

FCS_TLSS_EXT.1.2 The TSF shall deny connection from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and **TLS 1.1**.

FCS_TLSS_EXT.1.3 The TSF shall **generate EC Diffie-Hellman parameters over NIST curves secp384r1 and no other curves.**

Application note: The unauthenticated TLS connection is only used for the provision connection of the TOE. This is also used for the browser connection by the administrator. This would be sufficient since there is also a user name/password authentication done.

6.2.20 FCS_TLSS_EXT.2 – TLS Server Protocol (Authenticated)

FCS_TLSS_EXT.2.1 The TSF shall implement **TLS 1.2 (RFC 5246)** and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites:

- **TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289**

Application note: The ciphersuites used for the DPS webAPI connection and the DCS webAPI connection are the same. The same ciphersuite is used for DSB connections to a different DSS.

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

FCS_TLSS_EXT.2.3 The TSF shall **generate EC Diffie-Hellman parameters over NIST curves secp384r1 and no other curves.**

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

FCS_TLSS_EXT.2.5 When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the client certificate is invalid. The TSF shall also **not implement any administrator overrider mechanism.**

Application note: Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with RFC 5280.

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

Application note: The authenticated TLS connection is for all TLS connections with exception of the provisioning TLS connection ad for the administrator connections. The refinements have been performed only to remove weak cipher suites and key lengths.

6.2.21 FCS_TLSC_EXT.2 – TLS Client Protocol

FCS_TLSC_EXT.2.1 The TSF shall implement **TLS 1.2 (RFC 5246)** and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites:

- **TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289**

Application note: The encryption mechanism and cipher suites used to connect from the DSB of the TOE to the DSB of a different trusted DSS.

FCS_TLSC_EXT.2.2 The TSF shall verify that the presented identifier matches the reference identifier per RFC 6125 section 6.

FCS_TLSC_EXT.2.3 When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the server certificate is invalid. The TSF shall also **not implement any administrator override mechanism.**

FCS_TLSC_EXT.2.4 The TSF shall **present the Supported Elliptic Curves Extension with the following NIST curves: secp384r1 and no other curves** in the Client Hello.

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

6.2.22 FCS_HTTPS_EXT.1 – HTTPS Protocol

FCS_HTTPS_EXT.1.1 The TSF shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS.

FCS_HTTPS_EXT.1.3 If a peer certificate is presented, the TSF shall **not establish the connection** if the peer certificate is deemed invalid.

6.2.23 FDP_ITC.2 – Import of user data with security attributes

FDP_ITC.2.1 The TSF shall enforce the **client SFP** when importing user data, controlled under the SFP, from outside of the TOE.

Application note: The TOE imports encryption keys from the DCA for later use in either FCS_COP.1e or FCS_CKM.2c. The client SFP implies that the TOE shall receive and associate keys with DCA user from which the key was received. The keys are received over a mutually

authenticated channel, as specified in FCS_TLSS_EXT.2. There is no additional flow or access control policy imposed on the object.

FDP_ITC.2.2 The TSF shall use the security attributes associated with the imported user data.

Application note: The keys are associated with the DCA user and session (TLS connection) through which the keys are imported. These attributes will be contained in the client certificate submitted during the TLS handshake.

FDP_ITC.2.3 The TSF shall endure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

FDP_ITC.2.4 The TSF shall ensure that the interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: **associate the keys with the DCA from whom the keys were received.**

Application note: These keys will later be used for operations supporting the DCA as demonstrated by other requirements. When importing, the TOE will store these keys and associate them with the DCA for later use.

6.2.24 FIA_UAU.2 – User authentication before any action

FIA_UAU.2.1 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Application note: This SFR is both for the user name / password mechanism used by the administrators as well for the service access. The administrators are authenticated using a web browser and a HTTPS connection to the web server running on the DCC. The service access is identified and authenticated using a password based SSH.

6.2.25 FIA_UAU.4 – Single-use authentication mechanism

FIA_UAU.4.1 The TSF shall prevent reuse of authentication data related to **one-time link**.

Application note: This is for the one-time random link that is provided to the DCA user for the registration during provisioning.

6.2.26 FIA_UID.2 – User identification before any action

FIA_UID.2.1 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

Application note: This requirements applies to the authentication of administrators as well as to the service access.

6.2.27 FMT_MTD.1 – Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to **modify** the **Systems and Company information** to **the individual users of administrative roles that have been explicitly assigned that right**.

Application note: Administrative users may either have the right to manage *all systems* and *all companies* or may have restrictions on which system or company they are allowed to manage. Note that User Admin may be limited to one or more companies, while other roles may have access to all companies. The FMT_MTD is used rather than FDP_ACC and FDP_ACF since there is management function only associated with the right of management roles and not an access control policy relying on security attributes.

6.2.28 FMT_SMF.1 – Specification of Management Functions

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

- **Edit users, groups, departments and emergency contacts [User Admin]**
- **Statistics [User Admin]**
- **Browser Verification [User Admin]**
- **Company Administration [Company Admin]**
- **Administrators Roles & Permissions [Company Admin]**
- **View Log [System Admin]**
- **Servers, Certificates & Systems (Read) [System Admin]**
- **Servers, Certificates & Systems (Modify) [Service Access]**
- **System Management [Service Access]**
- **DSB Management [Service Access]**

Application note: The management functions listed above cover all management functions of the TOE. The role shown in brackets indicates the privilege necessary for performing the task. Since the roles are hierarchical privileges in the following increasing order User Admin, Company Admin, System Admin and Service Access, a user with the System Admin role has the privileges of the User Admin and Company Admin and therefore can perform all tasks of these roles.

Note that User Admins can only modify the contents within the companies they have access to, and administrators can only modify lower level administrators.

6.2.29 FMT_SMR.1 – Security roles

FMT_SMR.1.1 The TSF shall maintain the roles: **User Admin, Company Admin, System Admin and Service Access**

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

Application note: The roles are limited to administrative users since the users, i.e. the DCA users, are not directly users of the TOE and are accessing the TOE indirectly through the DCA only. Each administrative user has one role only (which makes sense since they are hierarchical anyhow). The roles are hierarchical in the order listed above, so that the privileges available to users with User Admin role is a subset of the Company Admin, etc.

6.2.30 FTP_ITC.1 – Inter-TSF Trusted Channel (TLS and SSH)

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2 The TSF shall permit **the TSF or another Trusted IT product** to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for **cross-system communication via the Dencrypt Server Bridge.**

Application note: There are three different trusted channels:

- This first one is the trusted channel (TLS) between the TOE (the DCS, DPS and DCC) and external components (i.e. DCA and browser on administration workstations). The cryptography for TLS is described in FCS_TLSS_EXT.1, FCS_TLSS_EXT.2 and FCS_HTTPS_EXT.1.
- Then there is the second trusted channel (SSH) between the client for service access and the TOE. The cryptography for SSH is described FCS_SSHS_EXT.1. Communications between the TOE and external components through SSH is always initiated by the external components and never by the TOE.

- The final trusted channel is initiated between the DSB of the TOE and another trusted DSS. The cryptography for TLS is described in FCS_TLSS_EXT.2 and FCS_TLSC_EXT.2.

6.3 Security functional requirements rationale

6.3.1 Coverage

The following table provides a mapping of SFR to the security objectives, showing that each security functional requirement addresses at least one security objective.

	O.ACCESS	O.AUDIT	O.CA	O.CHANNEL	O.MANAGE	O.PROVISIONING	O.REMOTE	O.REVIEW	O.SERVICE	O.CLIENTKEY	O.PUSH	O.BRIDGE	O.TUNNEL
FAU_GEN.1		X											
FAU_SAR.1					X			X					
FAU_SAR.2	X							X					
FAU_STG.2	X	X											
FCS_CKM.1a (RSA key pair)			X	X					X			X	
FCS_CKM.1b (AES key)				X					X			X	
FCS_CKM.2b (Server public key)				X								X	
FCS_CKM.2c (DCA storage encryption key)										X			
FCS_CKM.2d (SSH Public key)									X				
FCS_CKM.4				X					X		X	X	
FCS_COP.1a (AES)				X					X			X	
FCS_COP.1b (Signature verification)				X					X			X	
FCS_COP.1c (Hashing)				X					X			X	
FCS_COP.1d (Certificate signing)			X										
FCS_COP.1e (Push encryption)											X		
FCS_COP.1f				X					X				
FCS_RNG.1			X	X		X			X			X	
FCS_SSHS_EXT.1									X				
FCS_TLSS_EXT.1 (Unauthenticated)						X							
FCS_TLSS_EXT.2 (Authenticated)				X								X	X
FCS_TLSC_EXT.2												X	
FCS_HTTPS_EXT.1							X						

	O.ACCESS	O.AUDIT	O.CA	O.CHANNEL	O.MANAGE	O.PROVISIONING	O.REMOTE	O.REVIEW	O.SERVICE	O.CLIENTKEY	O.PUSH	O.BRIDGE	O.TUNNEL
FDP_ITC.2										X	X		
FIA_UAU.2							X		X				
FIA_UAU.4						X							
FIA_UID.2							X		X				
FMT_MTD.1	X												
FMT_SMF.1	X				X								
FMT_SMR.1					X								
FTP_ITC.1 (TLS and SSH)				X					X	X	X	X	X

6.3.2 Sufficiency

The following rationale provides justification for each security objective for the TOE, showing that the security functional requirements are suitable to meet and achieve the security objectives.

Security Objectives	Security objectives
O.ACCESS	The objective • To ensure that administrators only can access information and functions that they are explicitly authorized for. is met by • FMT_SMF.1 limits the management functions for each role • FMT_MTD.1 restricts administrative users access to the management of certain systems and companies • FAU_STG.2 prevents any administrator from deleting any audit records and ensure that the audit space is not exhausted • FAU_SAR.2 ensures that read access to audit records is restricted only to authorized administrators
O.AUDIT	The objective • To provide audit evidence of security relevant events as well as authorised use of security management functions to allow identification of security violations attempts as well as maintain accountability of administrators. is met by • FAU_GEN.1 ensures that audit events are generated for each secure relevant event and each user management functions. FAU_STG.2 ensures that the audit space is not exhausted and prevents unauthorized modification of the audit trail.
O.CA	The objective • To generate it's own private-public keys and its own certificate as well as sign certificates for DCA. is met by • FCS_CKM.1a ensures that RSA key-pairs are generated for the TOE

Security Objectives	Security objectives
	• FCS_RNG.1 ensures that the RSA key-pairs are generated using standard conformant random number generator • FCS_COP.1d ensures that certificates are signed, both for the TOE and for the DCA.
O.CHANNEL	The objective: • To provide mutually authenticated and trusted channels to any outside components to protect information transmitted to and received from such components against unauthorised disclosure and to detect any modification of incoming information transmitted from such components, and to provide the means for such components to verify the integrity of information transmitted out of the TOE. is met by: • FTP_ITC.1 ensures that there is a trusted path between the TOE and the DCA, or between the TOE and a different DSS. • Key generation is ensured by FCS_CKM.1a, FCS_CKM.1b and FCS_RNG.1. Key distribution is done by FCS_CKM.2b which is part of the TLS 1.2 protocol as specified by FCS_TLSS_EXT.2. • The cryptographic functionality to achieve confidentiality, integrity and authenticity is ensured by FCS_COP.1a, FCS_COP.1b, FCS_COP.1c and FCS_COP.1f • Key destruction is ensured by FCS_CKM.4
O.MANAGE	The objective • To provide the authorized administrators with the means to manage the TSF and the DCAs associated with the TOE installation is met by • FAU_SAR.1 ensures that the audit read capability is provided to authorized administrators • FMT_SMF.1 specifies the management functions of the TOE • FMT_SMR.1 limits the management functions for each role
O.PROVISIONING	The objective • To provide an unpredictable link for one-time registration, ensuring that such a link is only available for a very limited time to limit the window of opportunity in case of no or late use of activation is met by • FIA_UAU.4 ensure that the one-time link can only be used for a limited time and only once. • FCS_RNG.1 ensures that the one-time link is unpredictable and cannot be guessed by an attacker. • FTP_ITC.1 ensures there is a trusted path between the TOE and the DCA.
O.REMOTE	The objective • To uniquely identify and authenticate administrators and provide them with a secure communication channel before allowing administrators any access to the TOE. is met by • FIA_UID.2 ensures that each administrator is successfully identified before being allowed to access the TOE.

Security Objectives	Security objectives
	• FIA_UAU.2 ensures that each administrator is successfully authenticated before being allowed to access the TOE. • FTP_ITC.1 ensure there is a trusted path between the TOE and the admin browser. • FCS_TLSS_EXT.1 ensures the TLS 1.2 protocol to cryptographically secure the connection. • FCS_HTTPS_EXT.1 ensures that the HTTPS protocol is implemented according to standard.
O.REVIEW	The objective • To provide an authorised administrator and only the authorised administrator with ability to read the audit trail. is met by • FAU_SAR.1 ensures that the audit read capability is provided to authorized administrators. • FAU_SAR.2 ensure that read access is restricted only to authorized administrators.
O.SERVICE	The objective • To provide the authorized secure service access to manage the TSFs and the TOE installation. is met by • FIA_UID.2 ensures that each administrator is successfully identified before being allowed to access the TOE. • FIA_UAU.2 ensures that each administrator is successfully authenticated before being allowed to access the TOE. • FTP_ITC.1 ensures that a trusted SSH channel is provided to authorized service access. • Key generation is ensured by FCS_CKM.1a, FCS_CKM.1b and FCS_RNG.1. Key distribution is done by FCS_CKM.2d which is part of the SSH protocol as specified by FCS_SSHS_EXT.1. • The cryptographic functionality to achieve confidentiality, integrity and authenticity is ensured by FCS_COP.1a, FCS_COP.1b, FCS_COP.1c and FCS_COP.1f. • Key destruction is ensured by FCS_CKM.4
O.CLIENTKEY	The objective • To distribute encrypted storage keys to support data-at-rest security for the DCA. is met by • FCS_CKM.2c ensures that the keys can be distributed to their respective DCA user. • FDP_ITC.2 ensures that the keys can be imported from a trusted DCA user and be associated with that DCA user. • FTP_ITC.1 ensures that the keys can be imported securely via the TLS channel where the DCA user is authenticated and trusted. • FCS_TLSS_EXT.2 ensures the TLS 1.2 protocol to cryptographically protect the connection where the key is transmitted.
O.PUSH	The objective • To encrypt the contents of a push notification before such notifications are sent to the DCA. is met by • FCS_COP.1e ensures that the contents are encrypted using 256

Security Objectives	Security objectives
	bit AES encryption. • FDP_ITC.2 ensures that the key can be imported from a trusted DCA user and be associated with that DCA user. • FTP_ITC.1 ensures that the keys can be imported securely via the TLS channel where the DCA user is authenticated and trusted. • FCS_TLSS_EXT.2 ensures the TLS 1.2 protocol to cryptographically protect the connection where the key is received. • FCS_CKM.4 ensures key destruction.
O.BRIDGE	The objective • To provide a trusted connection to a different DSS via the DSB. is met by • FTP_ITC.1 ensures that there is a trusted path between the TOE and the DSB of a different DSS. • Key generation is ensured by FCS_CKM.1a, FCS_CKM.1b and FCS_RNG.1. Key distribution is done by FCS_CKM.2b which is part of the TLS 1.2 protocol as specified by FCS_TLSS_EXT.2 or FCS_TLSC_EXT.2. • The cryptographic functionality to achieve confidentiality, integrity and authenticity is ensured by FCS_COP.1a, FCS_COP.1b, FCS_COP.1c and FCS_COP.1f • Key destruction is ensured by FCS_CKM.4
O.TUNNEL	The objectives • To provide a TCP tunnel between DCAs and the TOE. is met by • FTP_ITC.1 ensures that that the tunnel can be established. • FCS_TLSS_EXT.2 enables the TCP tunnel via TLS 1.2.

6.3.3 Dependency analysis between security functional components

The following table shows the dependencies of the SFRs and how these dependencies have been resolved.

SFR	Dependencies	Resolved?
FAU_GEN.1	FPT_STM.1	No, satisfied by OE.TIME
FAU_SAR.1	FAU_GEN.1	Yes, by FAU_GEN.1
FAU_SAR.2	FAU_SAR.1	Yes, by FAU_SAR.1
FAU_STG.2	FAU_GEN.1	Yes, by FAU_GEN.1
FCS_CKM.1a (RSA key pair)	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	Yes, by FCS_CKM.2b No, no key destruction is needed because the server private key is kept in the TOE for use as long as the corresponding certificate is valid
FCS_CKM.1b (AES)	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	Yes, by FCS_COP.1a Yes, by FCS_CKM.4
FCS_CKM.2b (Server public key)	[FDP_ITC.1 or FDP_ITC.2 or	Yes, by FCS_CKM.1a

SFR	Dependencies	Resolved?
	FCS_CKM.1] FCS_CKM.4	No, no key destruction is needed because the public key does not contain any secret information.
FCS_CKM.2c	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	Yes, by FDP_ITC.2 No, no key destruction is needed since this key is encrypted by the client, and can only be decrypted by the client.
FCS_CKM.2d	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	Yes, by FCS_CKM.1a Yes, by FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	Yes, FCS_CKM.1b
FCS_COP.1a (AES)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	Yes, by FCS_CKM.1b Yes, by FCS_CKM.4
FCS_COP.1b (Signature verification and generation)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	Yes, by FCS_CKM.1a No, no key destruction is needed. The client public key is used for signature verification and the public key does not contain any secret information. The server private key is kept in the TOE for use as long as the corresponding certificate is valid
FCS_COP.1c (Hashing)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	No, since no key is needed for the hash operation No, no key destruction is needed since there is no key associated with the hash operation
FCS_COP.1d (Certificate signing)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	Yes, by FCS_CKM.1a No, no key destruction is needed because the CA private key is kept in the TOE for use as long as the corresponding certificate is valid.
FCS_COP.1e (Push encryption)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	Yes, by FDP_ITC.2 Yes, by FCS_CKM.4
FCS_COP.1f (Keyed-hash)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	Yes, by FCS_CKM.1a Yes, by FCS_CKM.4
FCS_RNG.1	No dependencies	–

SFR	Dependencies	Resolved?
FCS_SSHS_EXT.1	FCS_CKM.1 FCS_CKM.2 FCS_COP.1/ DataEncryption FCS_COP.1/SigGen FCS_COP.1/Hash FCS_COP.1/ KeyedHash FCS_RBG_EXT.1	Yes, by FCS_CKM.1a Yes, by FCS_CKM.2d Yes, by FCS_COP.1a Yes, by FCS_COP.1b Yes, by FCS_COP.1c Yes, byFCS_COP.1f No, instead of using FCS_RBG_EXT.1 this ST is using FCS_RNG.1 defined in [RNGfc]
FCS_TLSS_EXT.1 (Unauthenticated)	FCS_CKM.1 FCS_CKM.2 FCS_COP.1/ DataEncryption FCS_COP.1/SigGen FCS_COP.1/Hash FCS_COP.1/KeyedHash FCS_RBG_EXT.1	Yes, by FCS_CKM.1b Yes, by FCS_CKM.2b Yes, by FCS_COP.1a Yes, by FCS_COP.1b Yes, by FCS_COP.1c Yes, byFCS_COP.1f No, instead of using FCS_RBG_EXT.1 this ST is using FCS_RNG.1 defined in [RNGfc]
FCS_TLSS_EXT.2 (Authenticated)	FCS_CKM.1 FCS_CKM.2 FCS_COP.1/ DataEncryption FCS_COP.1/SigGen FCS_COP.1/Hash FCS_COP.1/KeyedHash FCS_RBG_EXT.1	Yes, by FCS_CKM.1b Yes, by FCS_CKM.2b Yes, by FCS_COP.1a Yes, by FCS_COP.1b Yes, by FCS_COP.1c Yes, byFCS_COP.1f No, instead of using FCS_RBG_EXT.1 this ST is using FCS_RNG.1 defined in [RNGfc]
TCS_TLSC_EXT.2	FCS_CKM.1 FCS_CKM.2 FCS_COP.1/ DataEncryption FCS_COP.1/SigGen FCS_COP.1/Hash FCS_COP.1/KeyedHash FCS_RBG_EXT.1	Yes, by FCS_CKM.1b Yes, by FCS_CKM.2b Yes, by FCS_COP.1a Yes, by FCS_COP.1b Yes, by FCS_COP.1c Yes, byFCS_COP.1f No, instead of using FCS_RBG_EXT.1 this ST is using FCS_RNG.1 defined in [RNGfc]
FCS_HTTPS_EXT.1	[FCS_TLSC_EXT.1 or FCS_TLSS_EXT.1]	Yes, by FCS_TLSS_EXT.2, which is hierarchical to FCS_TLSS_EXT.1
FDP_ITC.2	[FDP_ACC.1 or FDP_IFC.1] [FTP_ITC.1 or FTP_TRP.1]	No, no dedicated Access Control or Information Flow Control policy is used since this import takes place over a mutually identified and authenticated TLS session initiated by the trusted DCA. Only the associated DCA data is available within this session and no additional controls are imposed. Yes, by FTP_ITC.1

SFR	Dependencies	Resolved?
	FPT_TDC.1	No, the TOE solely received the keys from the DCA. There is no need for consistency interpretation since the TOE is dependent on the trusted DCA for supplying the expected keys. No inter-TSF data consistency is necessary.
FIA_UAU.2	FIA_UID.1	Yes, by FIA_UID.2
FIA_UAU.4	No dependencies	–
FIA_UID.2	No dependencies	–
FMT_MTD.1	FMT_SMR.1 FMT_SMF.1	Yes, by FMT_SMR.1 Yes, by FMT_SMF.1
FMT_SMF.1	No dependencies	–
FMT_SMR.1	FIA_UID.1	Yes, by FIA_UID.2
FTP_ITC.1 (TLS and SSH)	No dependencies	–

6.4 Security assurance requirements

The security assurance requirements of this Security are those defined for the assurance level EAL2 augmented with ALC_FLR.2.

Assurance class	Assurance components
ADV: Development	ADV_ARC.1 Security architecture description
	ADV_FSP.2 Security-enforcing functional specification
	ADV_TDS.1 Basic design
AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
ALC: Life-cycle support	ALC_CMC.2 Use of a CM system
	ALC_CMS.2 Parts of the TOE CM coverage
	ALC_DEL.1 Delivery procedures
	ALC_FLR.2 Flaw reporting procedures (augmentation)
ASE: Security Target evaluation	ASE_CCL.1 Conformance claims
	ASE_ECD.1 Extended components definition
	ASE_INT.1 ST introduction
	ASE_OBJ.2 Security objectives
	ASE_REQ.2 Derived security requirements
	ASE_SPD.1 Security problem definition
	ASE_TSS.1 TOE summary specification
ATE: Tests	ATE_COV.1 Evidence of coverage
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing – sample

Assurance class	Assurance components
AVA: Vulnerability assessment	AVA_VAN.2 Vulnerability analysis

6.5 Security assurance requirements rationale

Dependencies within the EAL package selected (EAL2) for the security assurance requirements have been considered by the authors of CC Part 3 and are not analysed here again. The augmentation by flaw remediation, ALC_FLR.2, has no dependencies on other requirements. The security functional requirements in this Security Target do not introduce dependencies on any security assurance requirement; neither do the security assurance requirements in this Security Target introduce dependencies on any security functional requirement.

The EAL2 level was also deemed sufficient because this will provide a necessary assurance for a product that is not directly exposed to external attackers, but still able to resist attacker with basic attack potential.

The assurance requirements of the EAL2 package provides a full Security Target and requires an analysis using a functional and interface specification and a basic description of the architecture of the TOE, which would give sufficient confidence in the design and architecture and for the evaluator to perform an analysis of the design and architecture for the vulnerability analysis.

7 TOE Summary Specification

The TOE summary specification identifies the security functions that the TOE implements to meet the requirements defined in chapter 6 to the security target.

The table below shows which SFRs are satisfied by each of the TSFs.

TSF	SFRs met by the TSF
SF.ROLES	FAU_SAR.1 FIA_UAU.2 FIA_UID.2 FMT_SMR.1
SF.AUDIT	FAU_GEN.1 FAU_SAR.2 FAU_STG.2
SF.PROVISIONING	FCS_TLSS_EXT.1 (Unauthenticated) FTP_ITC.1 (TLS) FIA_UAU.4 FCS_RNG.1
SF.MANAGEMENT	FMT_SMF.1 FAU_SAR.1 FAU_SAR.2 FMT_MTD.1
SF.CHANNEL	FCS_COP.1a (AES) FCS_COP.1b (Signature verification and generation) FCS_COP.1c (Hashing) FCS_COP.1f (Keyed-hash) FCS_CKM.1a (RSA keypair) FCS_CKM.1b (AES) FCS_RNG.1 FCS_CKM.2b (Server public key) FCS_TLSS_EXT.1 (Unauthenticated) FCS_TLSS_EXT.2 (Authenticated) FCS_TLSC_EXT.2 FTP_ITC.1 (TLS) FCS_CKM.4 (Key destruction) FCS_HTTPS_EXT.1
SF.PUSH	FCS_COP.1e FDP_ITC.2 FTP_ITC.1 (TLS)
SF.SERVICE	FCS_SSHS_EXT.1 (SSH server protocol) FIA_UAU.2 FIA_UID.2 FTP_ITC.1 (SSH) FCS_CKM.1a (RSA) FCS_CKM.1b (AES) FCS_CKM.2d FCS_COP.1a (AES) FCS_COP.1b (Signature verification and generation) FCS_COP.1c (Hashing)

TSF	SFRs met by the TSF
	FCS_COP.1f (Keyed-hash) FCS_CKM.4
SF.UPDATE	FCS_TLSS_EXT.2 (Authenticated) FTP_ITC.1 (TLS)
SF.CERTIFICATE	FCS_CKM.1a (RSA key pair) FCS_RNG.1 FCS_COP.1b (Signature verification) FCS_COP.1d (Certificate signing)
SF.CLIENTKEY	FCS_CKM.2c (DCA storage encryption key) FDP_ITC.2 FTP_ITC.1 (TLS)

7.1 Administration

7.1.1 SF.ROLES – I&A, administrative roles and access control

Administrators can access the TOE using a web interface. The administrators will establish an HTTPS connection from the web browser (TOE environment) to the Apache web server of the DCC (part of the TOE). Administrators will have to identify and authenticate themselves before administrative access is given.

The apache service is installed with PHP that handles the https requests. Apache and PHP are part of the turnkey Linux distribution which all servers including DCC are installed with. The central framework used in the DCC is Laravel (https://laravel.com) which is a collection of libraries and services. These libraries and services provide an easy way to handle common server side tasks such as encryption, database connection, CLI, emails, error handling etc. Apache, PHP and Laravel are all part of the TOE.

The DCC also has a MySQL database (part of the TOE) containing DCC admin user information, server's connection information, preferences and permissions.

Amongst other it holds the following records for each administrator:

- UID of the administrator
- Username in cleartext
- Password, which is hashed with PHP Bcrypt
- Type of user, i.e. the role of the administrator that can either be User Admin, Company Admin, System Admin or Service Access

This means that each user will be assigned one role only. But since the roles are hierarchical there is no need to have more than one role. The privileges associated with each role are shown in the figure below.

User Admin The User Admins can perform actions on DCA users such as adding new users, editing existing users, inviting users to the system, removing users, adding or removing users to groups, administer group and administer departments. This role must be explicitly granted access to each company. This makes the User Admin role suitable if access restriction on different companies are of concern.

Company Admin The Company Admin can in addition create, edit and delete companies. The role can be given to users where companies access restriction is not a concern. They can also create links between groups belonging to different companies, providing the opportunity for DCA users to communicate cross-company. This role

that can edit permissions for other administrators. Company Admin can also analyze logs for system events.

System Admin The System Admin role is used for daily system operation and monitoring. In additions to the functionalities of the User Admin and Company Admin, the System Admin has access to monitor technical status of the server system.

Service Access The Service Access role is intended for system maintenance and updates. The role is restricted to Dencrypt technical support and service partners. The Service Access role has full access to the entire DCC. The Service Access can also create, edit and remove server components. This means that the Service Access is having full shell command access and can perform any system management tasks.

Permissions

Feature	User Admin	Company Admin	System Admin	Service Access
ADMINISTRATION				
Users	Limited Access*	Access	Access	Access
Import	Limited Access*	Access	Access	Access
Groups	Limited Access*	Access	Access	Access
Emergency Contacts	Limited Access*	Access	Access	Access
Teams	Limited Access*	Access	Access	Access
Departments	Limited Access*	Access	Access	Access
Notifications	Limited Access*	Access	Access	Access
Companies	No Access	Access	Access	Access
Administrators	No Access	Limited Access*	Limited Access*	Access
Statistics				
Calls	Limited Access*	Access	Access	Access
Messages	Limited Access*	Access	Access	Access
User Statistics	Limited Access*	Access	Access	Access
System				
Browser Verification	Access	Access	Access	Access
Password Policies	Read Only	Read Only	Access	Access
Licenses	No Access	Access	Access	Access
Standard Messages	No Access	Access	Access	Access
Logs	No Access	Access	Access	Access
Certificates	No Access	Read Only	Access	Access
Servers	No Access	No Access	Read Only	Access
Bridges	No Access	No Access	Access	Access
Features	No Access	No Access	Read Only	Access
Alerts	No Access	No Access	Access	Access
Backup	No Access	No Access	Limited Access*	Access
Maintenance	No Access	No Access	Read Only	Access

* User Admins can only see and modify content they have explicit company access to
* Company Admins can only create, modify and delete administrators with *lower* priviledges
* System Admins can create, modify and delete administrators with *lower or equal* privileges
* System Admins cannot generate or retrieve backup key

Table 1: Roles and permissions

The MySQL database also holds permission tables that explicitly specify which users can access which companies (relevant only for users that have the User Admin role). If a user has permissions to access more than one company, this user will have multiple records in the permission tables, one for each company.

For every request from the administrator's browser, the DCC performs a permission evaluation before serving the request. More specifically, the DCC first checks whether the role of the logged-in user is equal or above the role required for the requested operation. It then consults the permission tables to check if the user has permission to access the company the request is targeting to. How many checks the evaluation executes depends on the request's permission requirements and the user's role. If any check fails, the request is immediately rejected. The only exception is the login HTML request and the actual login request which any one can access.

Please, note that the management functions are described in SF.MANAGEMENT below.

7.1.2 SF.AUDIT – Audit generation and protection

The DCC generates a log event for all events that change the state of the DCC or other server system components such as the DCS, DPS, DCM, DDB and DSB. In addition to server state change, the log also audits DCC login attempts (both successful and unsuccessful) and error responses from HTTPS requests to server system components. This log is stored in the MySQL database with the setting that no queries can delete or modify entries in that database table. This means that no administrative user of the TOE can delete or modify the audit events. To ensure that audit is always active, there is a log cycle, where logs are overwritten after a specified period of time.

The audit generates for each audit event: The date and time of the event; type of event; subject identity (if applicable); and the outcome (success or failure) of the event; and for each audit event type the additional information listed in the tables below.

The following events are generated by the DCC.

Event generated by the DCC	Additional log data
Install root certificate on a DCM	Installed root certificate for serverid: {dcm}
Initialize a DCM by requesting a CSR	Initialized DCM with CSR request from serverid: {dcm}
Install intermediate certificate on a DCM	Installed intermediate certificate for serverid: {dcm}
Install external certificate on any server	Certificates installed for serverid: {serverid}
Enable or disable client authentication in the provisioning process	Provisioning with certificates set to: {state}
Adding Apple push certificates	APNS id removed: {APNS_id}
Removing Apple push certificates	APNS added: {APNS_name}
Updating Apple push certificates	APNS updated: {APNS_name}
Login attempt	For successful login: {Username} logged in from {IP} For failed login: Unauthorized login attempt from {IP} as {username}
Set SMS and email credentials for DPS	Credentials updated for serverid: {serverid} Note: This is not a feature in the evaluated configuration.
Set configuration on a server	Configuration scheme modified for {serverip} / Service access
Remove server from DCC	A {servertype} at {serverip} has been removed / Service access
Add server to DCC	New {servertype} at {serverip} has been added / Service access
Changing IP and apikey	{old_serverip} changed address to {new_serverip} and changed API Key / Service access
Changing IP	{old_serverip} changed address to {new_serverip} / Service access
Changing apikey	{serverip} changed API Key / Service access
Editing scheduled configuration in system	Scheduled configuration modified / Service access

Event generated by the DCC	Additional log data
settings	
Editing features in system settings	Features updated / Service access
Importing Excel file for user management	An excel file has been imported
Adding user to a group	User with userid {userid} has been added to {type} with groupid {groupid}
Remove user from a group	User with userid {userid} has been removed from {type} with groupid {groupid}
Send email invitation	Invitation (email) has been send to {userid}
Send sms invitation	Invitation (sms) has been send to {userid} Note: This is not a feature in the evaluated configuration.
Create company	New Company {companyname} has been created
Edit a company's logo	Companyid {companyid} updated
Edit a company's name	Companyid {companyid} has changed name
Delete a company	Companyid {comapnyid} has been deleted
Allow a company to add users to a group	Groupid {groupid} added to companyid {companyid}
Remove permission for a company to add users to a group	Groupid {groupid} removed from companyid {companyid}
Create group	New Group: {groupname} has been created
Edit a group's name	Groupid {groupid} has changed name
Delete group	Groupid {groupid} has been deleted
Create department	New Department {departmentname} has been created
Edit a department's name	Departmentid {departmentid} has changed name
Delete department	Departmentid {departmentid} has been deleted
Delete user	{userid} has been deleted
Edit user (excluding image)	{userid} has been updated
Edit user's image	{userid} has new image
Create user	New User: {userid} has been created
TLS connection attempt	Success or fail connection from {IP}
SSH connection attempt and termination	Success or fail connection from {IP}
Emergency contacts list create	Emergency list {NAME} created
Emergency contacts list update	Emergency list {ID} has changed name to {NAME}
Emergency contacts list add contact	Userid {ID} has been added to emergency list {ID}
Emergency contacts list remove contact	Userid {ID} has been removed from emergency list {ID}
Emergency contacts list share	Emergency list {ID} has been made available to companyid {ID}
Emergency contacts list unshare	Emergency list {ID} has been made unavailable to companyid {ID}
Emergency contacts list delete	Emergency list {ID} has been deleted
Send user notification	Notification sent to user {ID}
Change group link	Group id {ID}'s link to group id {ID} is set to {LINKTYPE}
Provision user with QR code	Invitation (manual) has been generated for user with userid {ID}
Add new bridge connection	Bridge connection for system {ID} created
Added remote bridge connection request	Bridge connection request imported for system {ID}
Remove bridge connection	Bridge connection for system {ID} has been deleted

Event generated by the DCC	Additional log data
Change maintenance mode of a server	Maintenance for server id {ID} set to {MODE}
Create standard message	Standard message {ID} has been created
Update standard message	Standard message with id {ID} has been updated to: {ID}
Delete standard message	Standard message with id {ID} has been deleted
Change priority of a standard message	Standard message with id {ID} has been moved up/down in order

The following events are generated by the DCS.

Event generated by the DCS	Additional log data
TLS connection attempt	Success or fail connection from {IP}
SSH connection attempt and termination	Success or fail connection from {IP}

The following events are generated by the DPS.

Event generated by the DPS	Additional log data
TLS connection attempt	Success or fail connection from {IP}
Use of provisioning one-time link	Connection from {IP}
SSH connection attempt and termination	Success or fail connection from {IP}

The following events are generated by the DCM.

Event generated by the DCM	Additional log data
TLS connection attempt	Success or fail connection from {IP}
Issue certificates	Issue of certificate {CN}
SSH connection attempt and termination	Success or fail connection from {IP}

The following events are generated by the DDB.

Event generated by the DDB	Additional log data
SSH connection attempt and termination	Success or fail connection from {IP}

The following events are generated by the DSB.

Event generated by the DSB	Additional log data
TLS connection attempt	Success or fail connection from {IP}
SSH connection attempts and termination	Success or fail connection from {IP}

The audit review is described under SF.MANAGEMENT below.

7.1.3 SF.MANAGEMENT – Management functions

The TOE management is performed by an identified and authenticated administrator that has been assigned a specific role (SF.ROLES) and using the browser connected to the DCC over an HTTPS connection (SF.CHANNEL).

The management functions available to the administrator depend on the role assigned, as described in SF.ROLE. The TOE provides the following management functions:

Edit Users, groups, departments and emergency contacts:

- Add, edit and remove following: users, groups, departments and emergency contacts.
- User Admin role must have been given explicit access to the company.

Statistics:

- Visualization of how many calls have been initiated on a system. These are grouped by total amount of calls per day.
- Visualization of message statistics.
- Visualization of user statistics and their status.

Browser Verification:

- Download the DSS root certificate for installation in the administrator's browser.

Company Administration:

- Add, edit and remove companies
- Linking existing groups to groups in other companies (to allow cross-company calls)
- Add, edit and remove standard messages

Administrator Roles & Permissions

- Add and remove DCC administrators
- Set permissions for DCC administrators
- Can not be performed for higher privilege administrators

View Log:

- View DCC audit log. Logs are saved in the DCC database and consist of the logged in user, the time of the event and custom text of the event.

Servers, Certificates & Systems (Read)

- Detailed view of a server including connection URL+port, status, certificate expiration, CPU load, memory usage and disk space used.

Servers, Certificates & Systems (Modify)

- Add, edit and remove servers from a system
- Configurations:
  - DCS configuration: SIP settings, database connection and common name
  - DPS configuration: Email/SMS configuration (saved on DPS), credentials to send Email/SMS (saved on DCC) and common name
  - DCM configuration: Common name
  - DDB configuration: such as MySQL username and password
- Setup an uninitialized DCM. This includes providing it with a root certificate (Step 1), retrieving its CSR (Step 2) and providing it with an intermediate certificate (Step 3).
- Installing certificates on DCM, DPS, DSB and DCS. This is done through three steps where the DCC handles all communication and file transfers:
  - Step 1: Request CSR from targeted server
  - Step 2: Provide the DCM with the CSR. This returns three certificates (root, intermediate, leaf).
  - Step 3: Provide the targeted server with the three certificates. The server will now install these.

DSB management:

- Set up, configure and remove connections to additional DSS via the DSB.

All the above mentioned management functions are provided via the DCC web interface towards administrator browsers. After an administrator has been successfully authenticated by username

and password, the DCC returns a session id to the browser which saves it in a cookie. This cookie is then included in subsequent requests from the browser to identify the authenticated user session. To protect against potential Cross-Site Request Forgery (CSRF) attacks, the DCC sends back a random token in each HTTPS response. The browser shall include this CSRF token in the next HTTPS POST request. By verifying the CSRF token the DCC can detect forged requests from the browser. The Laravel framework (part of the TOE) used by the DCC implements and enforces the CSRF tokens.

When users are removed by administrator via the DCC web interface, the DCM updates the CRL and stores it in the DDB.

7.2 Security functions provided to clients

7.2.1 SF.PROVISIONING – Secure provisioning of DCA

Provisioning starts by the Dencrypt administrator by adding the user to the DCC. This will then trigger the creation of an invitation link to the user. The invitation link points to the web server of the DPS. The link can be encoded into a QR code which a user can scan into the DCA using the iPhone native camera QR scan functionality.

As part of TOE environment, this invitation link must be provided in a secure way to the user, i.e. the link is not disclosed during transmission to anyone else than the intended user. The invitation link might be mailed to the user if the mail transmission between mail server and handset's mail client is encrypted and the mail server is controlled by the user's organisation.

Note: SMS does not meet the requirement of non-disclosure because the mobile operator that transmits the SMS has access to the its content, the invitation link.

The link contains a random string of 30 characters, which is generated by the Debian RNG (part of the TOE). When accessing the link a trusted channel is established using a server-authenticated TLS connection. This is to ensure that any data downloaded is protected against disclosure and modification. The TLS connection is using TLS 1.2 with 4096 bits RSA and NIST curve secp384r1. The server also validates the URL to ensure that the client provided a correct provisioning token.

When accessing that link the DCA user will receive provisioning data consisting of configuration settings for connecting to the DCS SIP server. The client will generate a certificate and submit a CSR to be signed by the DCM. The link will only be available for a limited time and once accessed the link and the provisioning data will be removed from the DPS. This time limit is set by administrator and changes does not affect already created invitations.

At the end of provisioning, the DCA will upload encryption keys for Push Notifications (cleartext) and its storage encryption key (encrypted) over the secure TLS channel.

7.2.2 SF.UPDATE – Update of configuration

The TOE updates DCA with new phonebooks whenever there is a change of users or groups in the DDB. If the configuration of the SIP Server has changed, the clients will also be updated with new settings for the DCA app. This is achieved by the DCA polling the TOE for updates.

After a DCA has successfully registered to the DCS, it will regularly poll the DCS for phonebook and setting updates. When a new version of phonebook/setting becomes available the DCA will download the updated version. All these steps are carried out through the TLS channel between the DCA and the DCS (SF.CHANNEL).

7.2.3 SF.CHANNEL – Secure communication channel (TLS)

The TOE does not initiate the outside connection with the DCA, but it can accept and establish a secure channel coming from the DCA or from the web browser of the administrator. The TOE can initiate and accept connections via the DSB to a different DSS.

Communication Server (DCS) accepts of the following connections:

- Secure SIP connection between DCA and the SIP server on DCS (TOE), which is a mutually authenticated TLS connection.
- Secure HTTPS connection between DCA and web server (webAPI) on DCS (TOE), which is a mutually authenticated TLS connection.
    - The functionality of the DCM is also reachable via this connection. The DCM does not have a separate interface to the client.
- TLS tunnel channel between DCA and web server, which provides tunnelling of voice or video communication over TCP and TLS 1.2

Communication Server (DCS) also accepts and initiates the following connections:

- Secure TLS connection between the TOE and another trusted DSS, to enable sharing of phonebook data and forwarding of calls

Provisioning Server (DPS) accepts of the following connections:

- Secure HTTPS connection between DCA and web server (webAPI) on DPS (TOE). This connection is not authenticated.

Control Center (DCC) accepts of the following connections:

- Secure HTTPS connection between the web browser of the administrator and the DCC (TOE). This connection is not authenticated.

When DCS (not DPS) receives a TLS connection requests from a client (see SF.CHANNEL), they fetch the latest CRL from the DDB and use it to check the validity of the client certificate. If the client certificate is listed as revoked in the CRL, the TLS connection request is rejected. The validity of a client certificate is determined by the certificate path, the expiration date, and the revocation status in accordance with RFC 5280.

The connection to the provisioning web server is only used once during the provisioning of new DCA and the link is only active within a limited time after the link has been provided.

For the connection between the administrator and the Apache web server of the DCC there is no client authentication since the administrator will authenticate using user name and password. It is assumed that the browser (TOE environment) performs server authentication and it is covered by OE.TRUSTANCHOR.

For the DSB, a TLS connection can be established to a separate DSS to enable calls between different systems. Initially, this connection is used to exchange the metadata and phonebook data to make the calls. SIP call initialisation and messages to users belong to a different domain are forwarded by the TOE DCS to the DCS on the other system. In this scenario the DCS can also act as a TLS client. In addition to providing the secure channel it also verifies the identity of the remote system via certificate validation.

All connections are using TLS v1.2. and are implemented using the OpenSSL library provided by the Debian Linux operating system (part of the TOE). The OpenSSL library both generates sessions keys using its own RNG as well as destroying keys after they are no longer needed.

7.2.4 SF.PUSH

In addition to SF.CHANNEL, the TOE can also communicate to the DCA via encrypted push notifications. The push notifications will be transferred via the systems of the DCA platform vendor. Since these notifications are not sent over the TLS encrypted SF.CHANNEL there is a need for separate protection measures. TOE will import a key from the DCA and use that for encryption

of any messages sent as push notifications. The importing of this key is done at the end of provisioning, where the key is transferred securely over the TLS channel between the TOE and an authenticated DCA. The DCA will then decrypt the message within the application once it has been received. Note that the transfer from the TOE to the vendor's systems are protected by TLS.

7.2.5 SF.SERVICE – Service access channel

The TOE provides a cryptographically secured network channels to allow remote service access to interact with the TOE. The OpenSSH application provides secure service access to the command line interface of the TOE. The console provided via OpenSSH provides the same environment as a local console. OpenSSH implements the SSHv2 protocol. The cryptographic primitives are provided by OpenSSL.

The TOE supports the following security functions of the SSH v2.0 protocol:

- Establishing a secure communication channel using the following cryptographic functions provided by the SSH v2.0 protocol:
  - Encryption as defined in section 4.3 of [RFC4253] – the keys are generated using the random number generator of the underlying cryptographic library
  - Diffie-Hellman key exchange as defined in section 6.1 of [RFC4253]
  - The keyed hash function for integrity protection as defined in section 4.4 of [RFC4253]

  Note: The protocol supports more cryptographic algorithms than the ones listed in the FCS_SSHS_EXT.1 and referenced below. Those other algorithms are not covered by this evaluation and should be disabled or not used when running the evaluated configuration.
- Performing user authentication requests as defined in chapter 5 of [RFC4252].
- Performing user authentication using passwords as defined in chapter 8 of [RFC4252].
- Checking the integrity of the messages exchanged and close down the connection in case an integrity error is detected.

The following table documents implementation details concerning the OpenSSH implementation's compliance to the relevant standards. It addresses areas where the standards permit different implementation choices such as optional features.

The security functional requirements are suitable to meet and achieve the security objectives.

Reference	Description	Implementation Details
RFC4253, chapter 5	Compatibility with old SSH versions	The OpenSSH implementation is capable of interoperating with clients and servers using the old 1.x protocol. That functionality is explicitly disabled in the evaluated configuration, it permits protocol version 2.0 exclusively.
RFC4253, section 6.2	Compression	OpenSSH supports the OPTIONAL "zlib" compression method.
RFC4253, section 6.3	Encryption	The ciphers supported in the evaluated configuration are listed in FCS_SSHS_EXT.1 for the SSH protocol.
RFC4252, chapter 8	Password Authentication Method: "password"	This authentication method is supported by OpenSSH but can be disabled by the administrator of the OpenSSH daemon.
RFC4252, chapter 8	Password change request and setting new password	The OpenSSH implementation supports the optional password change mechanism in the evaluated configuration.

Reference	Description	Implementation Details
RFC4252, chapter 9	Host-Based Authentication: "hostbased"	This authentication method is disabled in the evaluated configuration.

The OpenSSH applications of sshd, ssh and ssh-keygen use the OpenSSL random number generator seeded by pulling data from /dev/random or /dev/urandom to generate cryptographic keys. OpenSSL provides different DRNGs depending whether the FIPS 140-2 mode is enabled in the system.

7.3 Other security functions

7.3.1 SF.CERTIFICATE – Key generation and certificate management

As part of the installation of the DSS, the DCM generates a 4096-bit RSA key pair and obtains a certificate signed by the root CA. This certificate (called intermediate or system certificate) and the root certificate are installed on the DCM. The DCM is thus made ready to issue certificates to both DCAs and system servers. DCA users can access the functionality of the DCM via the DCS or DPS.

When a new DCA user is created and accesses the provisioning link, the DCA will generate a local 3072-bit RSA certificate and submit a CSR to the DCM via the DPS. The DCM will then sign the certificate with its own RSA private key if the supplied invite ID is valid and distribute the client cert and provisioning data to the DCA.

If the DCA detects that its certificate is about to expire. It generates by itself a 3072 bit RSA key pair, creates a CSR and sends the CSR to the DCM. The DCM signs the certificate with its own RSA private key and returns the signed certificate to the client.

For server certificates, each server generates by itself a 4096-bit RSA key pair, creates a CSR and sends it to the DCM. The DCM signs the certificate with its own RSA private key and returns the signed certificate to the server. Also in this case the RSA key pairs are generated using OpenSSL that is part of Debian Linux. It relies on the RNG which is part of OpenSSL.

The DCM stores all certificates it has issued in the DDB MySQL database.

Administrators (Service Access) can renew the certificates for all externally visible servers, i.e. the DPS, DSB, DCS and the DCC itself. The management function for this is described in SF.MANAGEMENT and this is part of the DCC.

The actual generation of RSA key pairs and the certificate signing are performed as part of the SF.CERTIFICATE.

7.3.2 SF.CLIENTKEY – Key distribution to support client data-at-rest security

The TOE provides support for data-at-rest functionality of the DCA by storing and providing an encrypted cryptographic AES key to the DCA. The DCA generates a key to encrypt its stored files. As a measure to protect it's data-at-rest in the event of compromise of its underlying hardware, it encrypts this key and submits it to the DSS (TOE). Only the Key Encryption Key is stored locally by the DCA, ensuring that its storage can only be decrypted after a valid connection and authentication to the TOE. The TOE must as such import, store and distribute this key to the DCA client. These actions are performed via the secure and mutually authenticated TLS channel between the DCA and the TOE.

7.4 Cryptographic functions and parameters

This section summarizes the cryptographic mechanisms and primitives and parameters used by the TSFs previously described.

TLS	Used by **SF.CHANNEL**

	TLS 1.2 Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 Elliptic curves: secp384r1 • Used for the DPS webAPI connection • Used for the DCS webAPI connection • Used for the DCC Web Interface • Used for the SIP Server DCS connection • Used for the DSB connection • Used by the TCP tunnel
SSH	Used by **SF.SERVICE** SSH Cipher options: • aes256-gcm • sha-rsa • ecdh-sha2-nistp384 • hmac-sha2-512
RSA key generation and signing	Used by **SF.CERTIFICATE** for generating keys and signing certificates • RSA 4096 bits
X509 Certificates	Used by **SF.CHANNEL** for the TLS authentication • RSA 4096 bits • SHA512
RNG	Random number generation uses the OpenSSL RNG from Debian Linux.
Push encryption AES	Used by **SF.PUSH** for the encryption of push notifications. • AES-256 CFB

8 Abbreviations, terminology and references

8.1 Abbreviations

AES	Advanced Encryption Standard
AES-CM	AES – Counter Mode
CC	Common Criteria
CN	Common name in a certificate
CSR	Certificate Signing Request
CSRF	Cross-Site Request Forgery
DCM	Dencrypt Certificate Manager
DCC	Dencrypt Control Center
DCS	Dencrypt Communication Server
DDB	Dencrypt Database
DH	Diffie-Hellman key exchange
DPS	Dencrypt Provisioning Server
DTLS	Datagram Transport Layer Security
EAL	Evaluation Assurance Level
GCM	Galois/Counter Mode
HMAC	Keyed-Hash Message Authentication Code
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP over TLS
IEC	International Electrotechnical commission
ISO	International Organization for Standardization
MDM	Mobile Device Management
NIST	National Institute of Standards and Technology
OSP	Organisational Security Policy
PP	Protection Profile
RNG	Random Number Generation
RSA	Acronym for Rivest, Shamir, Adleman, the creators of the RSA algorithm
SAR	Security Assurance Requirement
SAS	Short Authentication String
SFR	Security Functional Requirement
SIP	Session Initiation Protocol
SIPS	SIP over TLS
SMS	Short Message Service
SRTP	Secure Real-time Transport Protocol
ST	Security Target

TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality
VoIP	Voice over IP
ZRTP	Zimmermann Real-time Transport Protocol

8.2 References

[CC] Common Criteria for Information Technology Security Evaluation. Part 1: Introduction and general model, April 2017, Version 3.1 Revision 5, CCMB-2017-04-001; Part 2: Security functional Components, April 2017, Version 3.1 Revision 5, CCMB-2017-04-002; Part 3: Security assurance components, April 2017, Version 3.1 Revision 5, CCMB-2017-04-003.

[CEM] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, April 2017 Version 3.1 Revision 5, CCMB-2017-04-004.

[cPPND] Collaborative Protection Profile for Network Devices, Version 2.1, 24-Sept-2018. https://www.commoncriteriaportal.org/files/ppfiles/CPP_ND_V2.1.pdf

[ISO15446] Technical Report ISO/IEC TR 15446, Information technology – Security techniques – Guide for the production of Protection Profiles and Security Targets, Second edition 2009-03-01.

[ISO10118] ISO/IEC 10118-3:2018, October 2018, Information technology – Security Techniques – Hash-functions – Part 3: Dedicated hash-functions

[PPST-Guide] The PP/ST Guide, August 2010, Version 2, Revision 0, Bundesamt für Sicherheit in der Informationstechnik.

[FIPS186-4] Federal Information Processing Standards Publication 186-4, Digital Signature Standard (DSS), July 2013.

[FIPS197] Federal Information Processing Standards Publication 197, Advanced Encryption Standard (AES), November 26, 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[NIST SP 800-38A] NIST Special Publication 800-38A 2001 Edition, NIST Special Publication 800-38A 2001 Edition, Recommendation for Block Cipher Modes of Operation. http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf

[NIST SP 800-38D] NIST Special Publication 800-38D, November 2007, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf

[NIST SP 800-56A] NIST Special Publication 800-56A Rev. 3, April 2018, Revision 2, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”;

[PKCS1v2.1] PKCS #1 v2.1: RSA Cryptography Standard, RSA Laboratories June 14, 2002 https://www.teletrust.de/fileadmin/files/oid/oid_pkcs-1v2-1.pdf

[RFC3261] SIP: Session Initiation Protocol, June 2002

[RFC3711] The Secure Real-time Transport Protocol (SRTP), March 2004

[RFC5246] The Transport Layer Security (TLS) Protocol, Version 1, August 2008

[RFC5289] TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM), August 2008.

[RFC4253] The Secure Shell (SSH) Transport Layer Protocol, January 2006

[ISO9797] ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”

[RNGfc] A proposal for: Functionality classes for random number generators, Bundesamt für Sicherheit in der Informationstechnik, Version 2.0, 18 September 2011.