Security Target for Dencrypt Server System

Version 1.1

Editor: Staffan Persson, atsec information security GmbH

Executive summary

This document is the Common Criteria Security Target for Dencrypt Server System. It is following the specification given in Part 1 appendix A of the Common Criteria version 3.1 release 4.

Document History

Version	Change Date	Author	Changes
0.1	2016-10-18	Staffan Persson	First draft version
0.2	2016-10-26	Staffan Persson	Updated after discussions with Dencrypt
0.3	2016-10-31	Staffan Persson	Further updates in section 5, 6 and 7
0.4	2016-11-11	Staffan Persson	Minor change to OSP.PROVISIONING
0.5	2016-11-23	Staffan Persson	Updated section 1 and 7 (adding 3rd party components to the TOE)
0.6	2016-11-29	Staffan Persson	Updated section 6
0.7	2016-12-09	Staffan Persson	Fixed some inconsistencies in encryption mode and hash length
0.8	2017-01-18	Søren Sennels	Product renaming
0.9	2017-02-07	Staffan Persson	Fixed evaluator comments
0.10	2017-02-16	Staffan Persson	Updated based on new input from Dencrypt and evaluator comments
0.11	2017-02-24	Staffan Persson	Clarified some crypto issues and updated the audit events in the SFRs and TSS.
0.12	2017-03-10	Staffan Persson	Evaluator comments. Minor fixes.
0.13	2017-05-09	Staffan Persson	Added the SSH service access.
0.14	2017-06-01	Staffan Persson	Added the SSH parameters and CRL check.
0.15	2017-06-06	Staffan Persson	Added the SSH to the DDB, and some fixes
0.16	2017-06-23	Staffan Persson	Fixed mapping to O.SERVICE, added guides
1.0	2017-09-05	Staffan Persson	Updated the TOE and guidance versions
1.1	2017-10-11	Søren Sennels	Updated description of A.NETWORK and OE.NETWORK.

Contents

1 Introduction

1.1 Security Target identification and organisation

Title:	Security Target for Dencrypt Server System version 2.0
ST Version:	1.1
Status:	Final version
Date:	2017-10-11
Sponsor:	Dencrypt A/S
Developer:	Dencrypt A/S
Keywords:	Mobile Application Management, VoIP, Voice and Data 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 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 2.0 consisting of the following components:

- Dencrypt Certificate Manager (DCM), version 1.0.90
- Dencrypt Provisioning Server (DPS), version 1.0.163
- Dencrypt Control Center (DCC), version 3.0.69
- Dencrypt Database (DDB), version 1.0.59
- Dencrypt Communication Server (DCS), version 1.0.434

1.3 TOE type

The TOE is software only and consists of the VoIP server and management components that are part of the server system of the Dencrypt Communication Solution to support mobile device clients for end-to-end voice encryption and encrypted live chat between iPhones.

1.4 TOE overview

The Dencrypt Server System consists of the server components of the Dencrypt Communication Solution that are necessary to support provisioning, management and call establishment of mobile device client for end-to-end voice encryption and encrypted live chat between iPhones. The server components include the SIP server for call establishment as well as components for the provisioning and management of the Dencrypt Talk app on the handset. The Dencrypt Talk, the MDM or any other apps on the handset are not included into the TOE.

Note: The Dencrypt Talk 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 Dencrypt Talk clients and TOE itself
  - Auditing and audit review
- Secure provisioning of Dencrypt Talk clients
- Trusted channel to clients
- Trusted channel to service access
- Provisioning to end users of new configurations and phone books
- Key generation and certificate issuing and a certificate authority

The Dencrypt Communication Solution consists of Dencrypt Server System (the TOE) and Dencrypt Talk on mobile devices. The Dencrypt Server System in turn consists of a Dencrypt Communication Server (a VoIP server), a Dencrypt Database (provides database services to DCS), a Dencrypt Certificate Manager (signs server and client certificates), a Dencrypt Provisioning Server (provisions clients) and a Dencrypt Control Center (provides administrator interface). 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 Talk is specified in another Security Target and subject to a separate evaluation and certification.

1.5 TOE description

1.5.1 Introduction and intended use

The key feature of the Dencrypt Server System and the Dencrypt Communication Solution is to provide mobile devices with a secure end-to-end voice and live chat within closed user groups that are centrally managed. Within the Dencrypt Communication Solution the Dencrypt Server System provides secure provisioning, Dencrypt Talk 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 red 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.

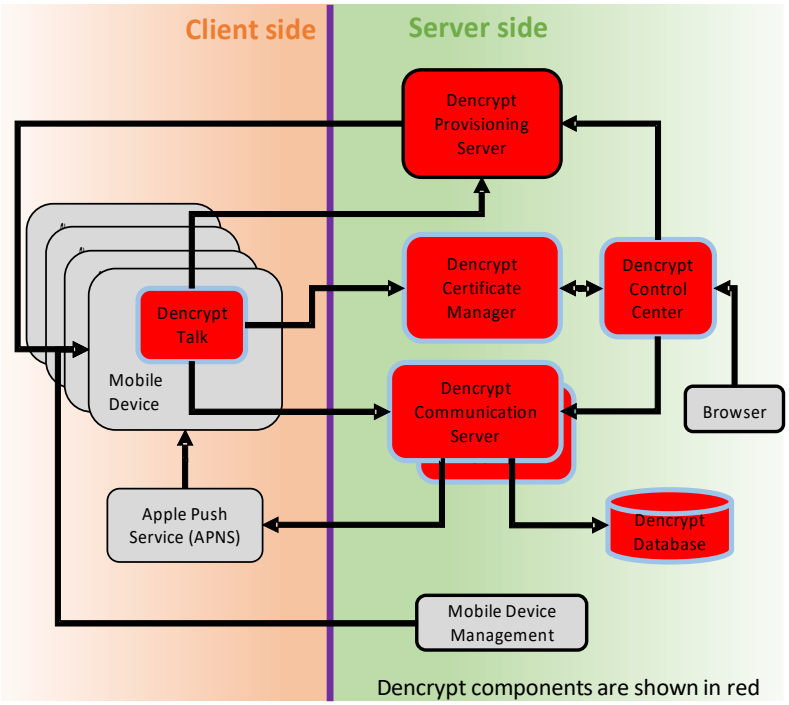

Illustration 1, The Dencrypt Communication Solution Overview

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 Talk (TOE environment)

The Dencrypt Talk is a mobile SIP client that runs on a mobile device (e.g. an iPhone). The client is able to establish encrypted calls and live chats with clients on other mobile devices using the SIP Server of the Decrypt Communication Server. The client is installed and updated using an MDM. The client must be configured and initialised before being used. This is done using the provisioning service.

Dencrypt Provisioning Server (part of TOE)

The Dencrypt Provisioning Server (DPS) is used to initialise clients with user credentials, DCS URL and temporary client key and certificate so they can communicate with the DCS and DCM. The client is provided with a HTTPS web link for the initialisation. 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 that is only reachable within a safe environment.

Dencrypt Communication Server (part of TOE)

The Dencrypt Communication Server (DCS) provides the SIP Services that are

necessary for the Clients to establish voice and live chat communication between two or more clients.

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 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 Dencrypt Talk and Dencrypt Server System use mutually authenticated TLS connections. The required TLS certificates are issued by the Dencrypt Certificate Manager by the following procedure: The client or server generates the private/public key pair and creates a certificate signing request (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.

Note: The provisioning process deviates from the above procedure because the DCM generates both the private key and the certificate (a certificate is the public key with meta data). However, this key pair is only temporary and will be replaced by a new key/certificate pair as soon as Dencrypt Talk has been successfully provisioned.

All the above-mentioned backend servers (DCC, DCS, DPS, DCM and DDB) 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 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.

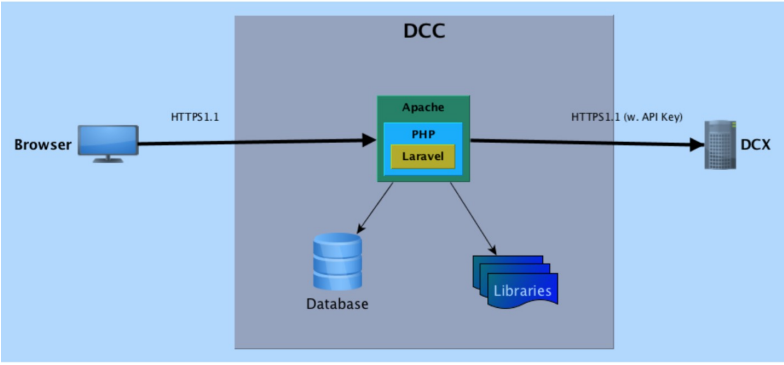

Illustration 2, The DCC Overview

1.5.2.2 Provisioning and user registration process

The provisioning process consists of two independent steps:

- Installation of Dencrypt Talk
- Provisioning of the data to the Dencrypt Talk where the data are the following:
    - DCS user credentials,
    - Dencrypt Server System domain,
    - temporary client key,
    - and temporary client certificate.

The Dencrypt Talk is provided and installed using an MDM. The MDM is not part of the Dencrypt Server System but considered as part of the TOE environment. It is assumed that the MDM is under control of the user's organization.

Although an MDM might offer to configure apps, provisioning is a sensitive process and the security of an ordinary MDM system may not be considered secure enough for the provisioning of Dencrypt Talk. Additionally, there might be a separation of duties between MDM administrators and Dencrypt Talk administrators. Thus, Dencrypt Server System provides its own provisioning server to facilitate the initial configuration of Dencrypt Talk.

Provisioning is started by the Dencrypt administrator adding the user to the Dencrypt Server System 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 Dencrypt Talk. Dencrypt Talk parses the link, fetches the provisioning data from the DPS and installs the data. 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.

1.5.2.3 Managing settings and phone book

The Dencrypt Talk (TOE environment) only allows calls to persons listed in the local phonebook. 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. The user administrator for the Dencrypt Server System(part of TOE) can change the groups of users to whom a specific user can call to at any time.

The Dencrypt Communication Server takes care of distributing the phonebook to the individual users. When a user starts the Dencrypt Talk, it establishes a TLS connection to the DCS and makes a SIP registration. When registration is successful, the client will subscribe for phonebook changes. Right after subscription and if the phonebook has been changed, DCS notifies the Dencrypt Talk client 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.

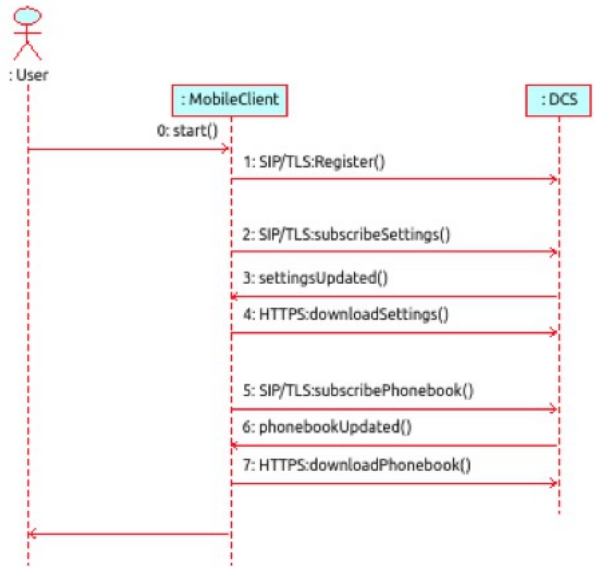

Illustration 3, The User registration process

1.5.2.4 Making secure calls

The uniqueness of the Dencrypt Communication Solution is that the end-to-end encrypted voice and live chat uses dynamic encryption, which ensures that each call session is encrypted using a randomly chosen algorithm and randomly chosen keys.

The following figure illustrates the steps for a secure call.

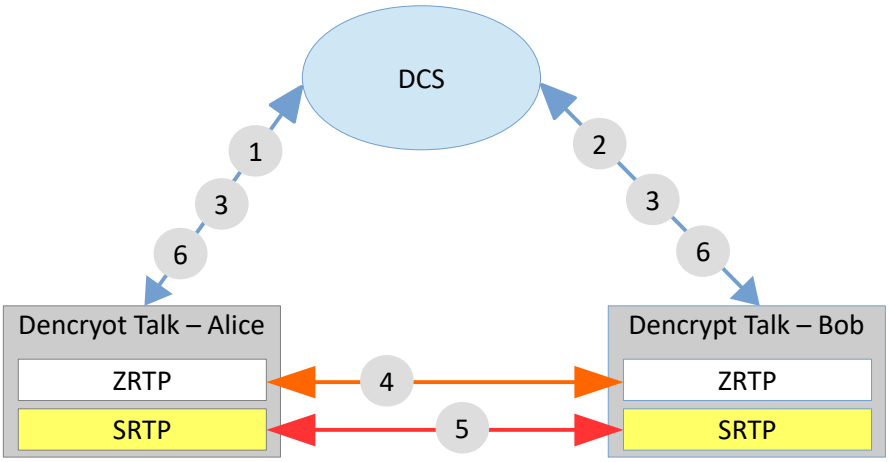

Illustration 4: Secure Call Life-Cycle

1. Alice's Dencrypt Talk client contacts the DCS she is registered to.
2. The SIP server resolves Bob's address and contacts Bob's Dencrypt Talk client. This resolution is limited because Alice can only contact the DCS for users listed in her phonebook.
3. SIP takes care of signalling, i.e. triggers Bob's Dencrypt Talk client to start ringing. As soon as Bob accepts the call, both Dencrypt Talk clients are signalled to start a media session for the real-time audio data stream.
4. Before the audio connection is encrypted, ZRTP takes over the data of media session. ZRTP negotiates a shared secret between Alice and Bob's Dencrypt Talk client. Additionally, ZRTP has been modified to securely and confidentially transport the parameters used for dynamic encryption.
5. Once ZRTP has established the shared secret, it calculates different keys for the bi-directional audio data stream between Alice and Bob. Both, these keys and the dynamic encryption parameters 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 realtime 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 data is implemented as multiple layers of encryption optimized for voice data over the SRTP protocol.
- Dynamic encryption of live chat data is implemented as multiple layers of encryption optimized for text over SIP.

- Both voice and live chat are bidirectional, i.e. each needs to encrypt and decrypt data and each bit stream uses different keys.
- The Dencrypt Talk uses 3072 bit Diffie-Hellman with 256 hash function for key negotiation over the ZRTP protocol.
- ZRTP key negotiation results in a common secret that is hashed into 4-letter phrase. If this 4-letter phrase is the same for both sides, the equality of the negotiated secret is confirmed, thus authenticating that the connection is not intercepted. This 4-letter readout hash-based key authentication is called SAS.
- Dynamic encryption for voice and live chat share the following keys:
  - 256 bit key for the standard AES-256 encryption. The key is provided by ZRTP.
  - 2 x 128-bit whitening keys as an additional encryption layer. The whitening keys are randomly generated by the encrypting entity and transmitted during ZRTP negotiation to the decrypting entity.
  - 128-bit dynamic encryption algorithm selection key that defines the S-Box for an additional AES-round. The algorithm selection key is randomly generated by the encrypting entity and transmitted during ZRTP negotiation to the decrypting entity.
- 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 iOS (TOE environment).

1.5.2.5 The TLS connection

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

The TLS connection to from the Dencrypt Talk client is mutually authenticated to ensure that only authorized clients can establish connections, i.e. to retrieve the phonebook information, and that the Dencrypt Talk client is not connecting to a server system that may deceive the Dencrypt Talk 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 Dencrypt Talk client and never by the server system, such as the DCS, DCM or DPS.

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 client authentication since administrators also have to authenticate themselves to the server anyhow.

Please note that the TLS connection to the provisioning web server is not mutually authenticated because the Dencrypt Talk client is not yet configured and has no client key and certificate. However, the client will perform server authentication.

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

1.5.2.6 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 and live chat 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
- 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 Dencrypt Talk clients and TOE itself
  - Auditing and audit review
- Secure provisioning of Dencrypt Talk clients
- Trusted channel to clients
- Trusted channel to service access
- Provisioning to end users of new configurations and phonebooks
- Key generation and certificate issuing and a certificate authority

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 Dencrypt Server System v. 2.0*
- *Maintenance Guide Dencrypt Server System v. 2.0*
- *Preparative Guide & Hosting Requirements Dencrypt Server System v. 2.0*
- *Acceptance Test & Handover Dencrypt Communication Solution Dencrypt Server System v. 2.0 Dencrypt Talk v. 4.2*

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, and the MySQL database) and any other software components that are necessary for the TOE.

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 Dencrypt Talk App is installed.
- An MDM provides a local application store server for offering the Dencrypt Talk and other approved signed applications.
- The hardware and software components (excluding the TOE components) that are delivered as part of the ISO image where the TOE is installed. These components are necessary for running the TOE components of the Dencrypt Server System with DPS, DCC, DCM, DDB and DCS.

- 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 claim conformance to CC version 3.1 Revision 4.

This ST claims no conformance to any Protection Profiles. This ST claims conformance to the EAL2 package of security assurance requirements, augmented with ALC_FLR.2.

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 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.

Threat agents could be 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.

Threat agents could also be 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 recorded by taking actions to exhaust audit storage capacity, thus masking an attacker's actions.

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 Dencrypt Talk applications associated with

OSP	Description
	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 Dencrypt Talk clients.

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 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 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 used for provisioning is provided to the correct Dencrypt Talk 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 Dencrypt Talk clients and administrator browsers for validation of TOE certificates when connecting to the TOE.
A.USER	It is assumed that the Dencrypt Talk 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.

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 Dencrypt Talk 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 Dencrypt Talk 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.

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 used for provisioning is provided to the correctDencrypt Talk user and not being disclosed to anyone else.
OE.NETWORK	The TOE environment must ensure that the underlying hardware of

Security Objective	Description
	the TOE and local network is dedicated to the TOE usage, functions and physically protected.
OE.NOEVIL	It is assumed that administrators 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 Dencrypt Talk clients and administrator browsers for validation of TOE certificates when connecting to the TOE.
OE.USER	The operational environment shall ensure that Dencrypt Talk 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.

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	OSP.MANAGE	OSP.ACCOUNT	OSP.PROVISIONING	OSP.SERVICE	OSP.CA	A.NETWORK	A.NOEVIL	A.PHYSICAL	A.REVIEW	A.TIME	A.WORKSTATION	A.LINK	A.TRUSTANCHOR	A.USER	A.FIREWALL
O.ACCESS			X																
O.AUDIT				X		X													

	T.COMMUNICATION	T.MASQUERADE	T.UNAUTH	T.UNDETECTED	OSP.MANAGE	OSP.ACCOUNT	OSP.PROVISIONING	OSP.SERVICE	OSP.CA	A.NETWORK	A.NOEVIL	A.PHYSICAL	A.REVIEW	A.TIME	A.WORKSTATION	A.LINK	A.TRUSTANCHOR	A.USER	A.FIREWALL
O.CA									X										
O.CHANNEL	X	X																	
O.MANAGE					X														
O.PROVISIONING		X					X												
O.REMOTE			X																
O.REVIEW				X		X													
O.SERVICE								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

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.
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 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).

Threat	Rationale for the security objectives
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.

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 Dencrypt Talk Application 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 for a very limited time. The link is then provided to the Dencrypt Talk 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 Dencrypt Talk clients.

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

Assumption	Rationale for the security objectives
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

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 and FCS_TLSS_EXT.2 (both for cryptography) and FCS_RNG.1 (for the random number generation).

This Security Target does not define its own extended components. The requirements FCS_TLSS_EXT.1, FCS_TLSS_EXT.2 and FCS_SSHS_EXT.1 have been taken directly from the extended components defined in [cPPND], while the requirement FCS_RNG.1 uses the extended component defined in [RNGfc].

6 Security requirements

6.1 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.

6.1.1 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/dial group
  - Remove user from a group/dial 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

- **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**
  - TLS connection attempt
  - Issue certificates
  - SSH session attempt
  - SSH session termination
- **DDB**
  - SSH session attempt
  - SSH session termination

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**.

6.1.2 FAU_SAR.1 – Audit review

FAU_SAR.1.1 The TSF shall provide **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.

6.1.3 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 System Admin and Service Access roles (the System Admin role is a subset of the Service Access role).

6.1.4 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.

6.1.5 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~~3072 and 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 DCM) and for the DCC for the administrator access. This is also the generation of the private-public key pair for the initial certificate for which the Dencrypt Talk client is provided during provisioning. This requirement is a refinement of FCS_CKM.1 defined in [cPPND].

6.1.6 FCS_CKM.1b – Cryptographic key generation (AES key for TLS)

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **as defined in the TLS v1.2 standard [RFC5246] for 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.

6.1.7 FCS_CKM.2a – Cryptographic key distribution (Provisioning of client key)

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **one-time use URL** that meets the following: **vendor specific key distribution**.

Application note: By accessing the HTTPS link the TOE will distribute a temporary 3072 bit private key and certificate to the Dencrypt Talk client.

6.1.8 FCS_CKM.2b – Cryptographic key distribution (Server 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 Dencrypt Talk Client or the administrator browser during the TLS handshake.

6.1.9 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 **zerorization** 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.

6.1.10 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.

6.1.11 FCS_COP.1b – Cryptographic Operation (Signature Verification)

FCS_COP.1.1 The TSF shall perform **digital signature verification** in accordance with a specified cryptographic algorithm **RSA Digital Signature Algorithm** and cryptographic key sizes **3072 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 verification performed as part of the TLS client authentication performed by the TOE. This also addresses the TOE verification of the signature on the certificate request (CSR) from the client. Both the temporary Dencrypt Talk certificate generated by the DCM and the Dencrypt Talk generated certificate are 3072 bits.

6.1.12 FCS_COP.1c – Cryptographic Operation (Hashing)

FCS_COP.1.1 The TSF shall perform **secure hash** in accordance with a specified cryptographic algorithm **SHA-384 ~~and cryptographic key sizes~~** that meet the following: **ISO/IEC 10118-3:2004**.

Application note: The secure hash is used by both FCS_TLSS_EXT.2 and FCS_TLSS_EXT.1 to ensure the integrity of the TLS connection.

6.1.13 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 Dencrypt Talk clients 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. The DCM signs the CSR if permitted and returns an X.509v3 client certificate to the Dencrypt Talk Client. It uses Python.

6.1.14 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.

6.1.15 FCS_SSHS_EXT.1 – SSH Server Protocol

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254, and **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,**~~ 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: **aes-256-gcm**.

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

FCS_SSHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses **hmac-sha2-512** and **no other MAC algorithms** as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7 The TSF shall ensure that **ecdh-sha2-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 the SSH connection be rekeyed after no more than 2^28 packets have been transmitted using that key.

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

6.1.16 FCS_TLSS_EXT.1 – TLS Server Protocol (Unauthenticated)

FCS_TLSS_EXT.1.1 The TSF shall implement **TLS 1.2 (RFC 5246)** supporting the following ciphersuites:

- ~~**Mandatory Ciphersuites:**~~
  - ~~**TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268**~~
- ~~**Optional Ciphersuites:**~~
  - **TLS_ECDHE_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 1.0, SSL 2.0, SSL 3.0, TLS 1.0 and **TLS 1.1**.

FCS_TLSS_EXT.1.3 The TSF shall generate key establishment parameters using RSA with key size ~~**2048 bits and 4096**~~ **3072 bits** and **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. The refinements have been performed only to remove weak cipher suites and key lengths.

6.1.17 FCS_TLSS_EXT.2 – TLS Server Protocol (Authenticated)

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

- ~~**Mandatory Ciphersuites:**~~
  - ~~**TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268**~~
- ~~**Optional Ciphersuites:**~~
  - **TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289**

Application note: The ciphersuites used for the DPS webAPI connection, the DCS webAPI connection and the DCM webAPI connection are the same. The refinements have been performed only to remove weak cipher suites and key lengths.

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

FCS_TLSS_EXT.2.3 The TSF shall generate key establishment parameters using RSA with key size ~~**2048**~~ **3072 bits and 4096 bits** and **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 The TSF shall not establish a trusted channel if the peer certificate is invalid.

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 peer.

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.

6.1.18 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.

6.1.19 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 Dencrypt Talk user for the registration during provisioning.

6.1.20 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.

6.1.21 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: User administrator may be limited one or more companies, while an organizer (next level) has access to all companies. System administrator may be limited to one or more systems, while a moderator (next level) has access to all systems. 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.

6.1.22 FMT_SMF.1 – Specification of Management Functions

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

- **Edit Profile [User Admin]**
- **Dashboard [User Admin]**
- **Call Statistic [User Admin]**
- **User Administration [User Admin]**
- **Cross-Company Administration [Company Admin]**
- **Administrator Roles & Permissions [Company Admin]**
- **Servers, Certificates & Settings (Read) [System Admin]**
- **View Log [System Admin]**
- **Servers, Certificates & Settings (Modify) [Service Access]**
- **System 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.

6.1.23 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 Dencrypt Talk users, are not directly users of the TOE and are accessing the TOE indirectly through the Dencrypt Talk client 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.

6.1.24 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 **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 **no functions**.

Application note: There are two different trusted channels. This first one is the trusted channel (TLS) between the TOE (the DCS, DCM, DPS and DCC) and external components (i.e. Dencrypt Talk clients and browser on administration workstations). The cryptography for TLS is described in FCS_TLSS_EXT.1 and in FCS_TLSS_EXT.2. 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 Any communications between the TOE and external components through TLS or SSH is always initiated by the external components and never by the TOE.

6.2 Security functional requirements rationale

6.2.1 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
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			
FCS_CKM.1b(AES key)				X					
FCS_CKM.2a (Provisioning of client key)						X			
FCS_CKM.2b (Server public key)				X					
FCS_CKM.4(				X					
FCS_COP.1a (AES)				X					
FCS_COP.1b (Signature verification)				X					
FCS_COP.1c (Hashing)				X					
FCS_COP.1d (Certificate signing)			X						
FCS_RNG.1			X	X		X			
FCS_SSHS_EXT.1									X
FCS_TLSS_EXT.1 (Unauthenticated)						X			
FCS_TLSS_EXT.2 (Authenticated)				X					
FIA_UAU.2							X		X
FIA_UAU.4						X			

	O.ACCESS	O.AUDIT	O.CA	O.CHANNEL	O.MANAGE	O.PROVISIONING	O.REMOTE	O.REVIEW	O.SERVICE
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

6.2.2 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.
O.CA	The objective • To generate it's own private-public keys and its own certificate as well as sign certificates for Dencrypt Talk clients. is met by • FCS_CKM.1a ensures that RSA key-pairs are generated for the TOE • 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 Dencrypt Talk clients.

Security Objectives	Security objectives
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.1ensures that there is a trusted path between the TOE and the Dencrypt Talk client. • Key generation is ensured by FCS_CKM.1b and 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 symmetric key generation for the AES key is done using the FCS_RNG.1. • Encryption is ensured by FCS_COP.1a. • Authentication is ensured by FCS_COP.1b • Integrity is ensured by FCS_COP.1c • 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 Dencrypt Talk applications 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. • FCS_TLSS_EXT.1 provides an unauthenticated TLS connection for the provisioning. Key generation is ensured by FCS_CKM.1a and key distribution is done by FCS_CKM.2a.
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

Security Objectives	Security objectives
	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.
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 • FCS_SSHS_EXT.1 ensures that a secure SSH channel is provided to authorized service access. • 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.

6.2.3 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 key for TLS)	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	Yes, by FCS_COP.1a Yes, by FCS_CKM.4
FCS_CKM.2a (Provisioning of client key)	[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 temporary key pair will be replaced by the client-generated key pair

SFR	Dependencies	Resolved?
		once the client has been successfully provisioned.
FCS_CKM.2b (Server public key)	[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 public key does not contain any secret information.
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)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	No, instead of using import of user data (e.g. FDP_ITC.1) this ST is using FMT_MTD.1 No, no key destruction is needed because client public key is used for signature verification and the public key does not contain any secret information.
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_RNG.1	No dependencies	–
FCS_SSHS_EXT.1	FCS_COP.1(1) FCS_COP.1(2) FCS_COP.1(3)	Yes, by FCS_COP.1a Yes, by FCS_COP.1b Yes, by FCS_COP.1c
FCS_TLSS_EXT.1 (Unauthenticated)	FCS_CKM.1 FCS_COP.1(1) FCS_COP.1(2) FCS_COP.1(3) FCS_RBG_EXT.1	Yes, by FCS_CKM.1b Yes, by FCS_COP.1a Yes, by FCS_COP.1b Yes, by FCS_COP.1c 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_COP.1(1) FCS_COP.1(2) FCS_COP.1(3) FCS_RBG_EXT.1	Yes, by FCS_CKM.1b Yes, by FCS_COP.1a Yes, by FCS_COP.1b Yes, by FCS_COP.1c No, instead of using FCS_RBG_EXT.1 this ST is using FCS_RNG.1 defined in [RNGfc]

SFR	Dependencies	Resolved?
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)	No dependencies	–

6.3 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 Basic functional specification
	ADV_TDS.1 Basic design
AGD: Guidance documents	AGD_OPE.1 Operational user guidance OBJ
	AGD_PRE.1 Preparative procedures
ALC: Life-cycle support	ALC_CMC.2 Use of a CM system
	ALC_CMS.2 Use of a CM system
	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 OBJ Evidence of coverage
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing – sample
AVA: Vulnerability assessment	AVA_VAN.2 Vulnerability analysis

6.4 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_CKM.1a (RSA key generation) FCS_RNG.1 FCS_CKM.2a (Provisioning of client key)
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) FCS_COP.1c (Hashing) FCS_CKM.1b (AES key for TLS) FCS_RNG.1 FCS_CKM.2b (Server public key) FCS_TLSS_EXT.1 (Unauthenticated) FCS_TLSS_EXT.2 (Authenticated) FTP_ITC.1 (TLS) FCS_CKM.4 (Key destruction)
SF.SERVICE	FCS_SSHS_EXT.1 (SSH server protocol) FIA_UAU.2 FIA_UID.2 FTP_ITC.1 (SSH)
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)

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.

Feature	User Admin	Company Admin	System Admin	Service Access
Edit Profile	✔	✔	✔	✔
Dashboard	✔	✔	✔	✔
Call Statistics	✔	✔	✔	✔
User Administration	✔	✔	✔	✔
Cross-Company Administration		✔	✔	✔
Administrator Roles & Permissions		✔	✔	✔
Servers & Certificates (Read)			✔	✔
View Log			✔	✔
Servers & Certificates (Modify)				✔
System Management				✔

User Admin The User Admins can perform actions on Dencrypt Talk 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 link groups to multiple companies, providing the opportunity

for Dencrypt Talk users to communicate cross-company. This role that can edit permissions for other administrators.

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
- Analyze logs for system events

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. Unlike all other roles, this role does not require permissions set for every system. The Service Access can also create, edit and remove systems. This means that the Service Access is having full shell command access and can perform any system management tasks.

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) and which users can access which systems (relevant for users that have the User Admin, Company Admin or System Admin role). If a user has permissions to access more than one company/system, this user will have multiple records in the permission tables, one for each company/system.

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/system 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. 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}

Event generated by the DCC	Additional log data
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 settings	Scheduled configuration modified / Service access
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/dial group	User with userid {userid} has been added to {type} with groupid {groupid}
Remove user from a group/dial 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}

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}

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 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 Profile:

- Change own password (requires typing current password)
- Set default system for the user when logging in

Dashboard:

- Overview status for all the servers the user (i.e. administrator) has access to

Call Statistics:

- Visualization of how many calls have been initiated on a system. These are grouped by total amount of calls per day.

User Administration:

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

Cross-Company Administration:

- Add, edit and remove companies

- Add existing groups to other companies (to allow cross-company calls)

Administrator Roles & Permissions

- Add and remove DCC administrators
- Set permissions for DCC administrators

Servers, Certificates & Systems (Read)

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

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 (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: None
- 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 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.

System Management:

- Add, rename and remove systems from the DCC

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 Dencrypt Talk clients

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.

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 list a trusted channel is established using an unauthenticated 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 3072 bits RSA and NIST curve secp384r1.

When accessing that link the Dencrypt Talk user will receive provisioning data consisting of a temporary certificate and configuration settings for connecting to the DCS SIP server. 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 at installation time and the administrators are not able to change it.

7.2.2 SF.UPDATE – Update of configuration

The TOE updates Dencrypt Talk clients with new phonebooks whenever there is a change of users or groups in the Dencrypt Database (DDB). If the configuration of the SIP Server has changed, the clients will also be updated with new settings for the Dencrypt Talk app. These are achieved by using a subscription based notification mechanism.

After a Dencrypt Talk client has successfully registered to the DCS, it subscribes for phonebook and setting updates. When a new version of phonebook/setting becomes available the DCS sends notification to the client which then downloads the new version. All these steps are carried out through the TLS channel between the Dencrypt Talk client and the DCS (SF.CHANNEL).

7.2.3 SF.CHANNEL – Secure communication channel (TLS)

The TOE updates Dencrypt Talk clients with new phonebooks whenever there is a change of users or group.

The TOE does not initiate any outside connection, but it can can accept and establish a secure channel coming from the Dencrypt Talk client or from the web browser of the administrator.

Communication Server (DCS) accepts of the following connections:

- Secure SIP connection between Dencrypt Talk and the SIP server on DCS (TOE), which is a mutually authenticated TLS connection.
- Secure HTTPS connection between Dencrypt Talk and web server (webAPI) on DCS (TOE), which is a mutually authenticated TLS connection.

Certificate Manager (DCM) accepts of the following connections:

- Secure HTTPS connection between Dencrypt Talk and web server (webAPI) on DCM (TOE), which is a mutually authenticated TLS connection.

Provisioning Server (DPS) accepts of the following connections:

- Secure HTTPS connection between Dencrypt Talk 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 or DCM (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 Dencrypt Talk clients and the link is only activate 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.

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.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.
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 Dencrypt Server System system, 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 Dencrypt Talk mobile clients and system servers.

When a new Dencrypt Talk user is created the DCC requests a temporary certificate from the DCM. The DCM generates a 3072-bit RSA key pair and creates a short lived certificate. The RSA private key and the temporary certificate are sent to the DCC which forwards them to the DPS for provisioning. The RSA key pairs are generated using OpenSSL that is part of Debian Linux.

After being provisioned with the temporary certificate, the Dencrypt Talk client detects that the 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 (long-lived) 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.

Administrators (Service Access) can renew the certificates for all externally visible servers, i.e. the DPS, DCM, 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.

7.4 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 DCM webAPI connection • Used for the DCC Web Interface TLS 1.2 Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 Elliptic curves: secp384r1 • Used for the SIP Server DCS connection
RSA key generation and signing	Used by **SF.CERTIFICATE** for generating keys and signing certificates • RSA 4096 bits • RSA 3072 bits (temporary for Dencrypt Talk)
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.

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
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, September 2012, Version 3.1 Revision 4, CCMB-2012-09-001; Part 2: Security functional Components, September 2012, Version 3.1 Revision 4, CCMB-2012-09-002; Part 3: Security Assurance Components, September 2012, Version 3.1 Revision 4, CCMB-2012-09-003.

[CEM] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, September 2012, Version 3.1 Revision 4, CCMB-2012-09-004.

[cPPND] Collaborative Protection Profile for Network Devices, Version 1.0, 27-Feb-2015. https://www.commoncriteriaportal.org/files/ppfiles/CPP_ND_V1.0.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.

[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

[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.

[Client MDM] Client Distribution by MDM iOS, Dencrypt, version 0.3, May 2016.

[Dynamic] Patent application WO 2013/060876 A1. http://orbit.dtu.dk/fedora/objects/orbit:128232/datastreams/file_be4c445c-73d5-4204-8805-67743afff6bf/content

[Guide Mod] Dencrypt Control Center Guide for Moderator, Dencrypt, September 2016.

[Guide Org] Dencrypt Control Center Guide for Moderator, Dencrypt, September 2016.

[Guide SysAdmin] Dencrypt Control Center Guide for Moderator, Dencrypt, September 2016.

[Guide UserAdmin] Dencrypt Control Center Guide for Moderator, Dencrypt, September 2016.

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