Security Target for Dencrypt Talk

Version 1.0

Editor: Staffan Persson, atsec information security GmbH

Executive summary

This document is the Common Criteria Security Target for Dencrypt Talk for the iPhone. 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	2016-07-13	Staffan Persson	First draft version
0.2	2016-07-22	Staffan Persson	Second draft version
0.3	2016-08-07	Staffan Persson	Third draft version with all section in place
0.4	2016-08-17	Staffan Persson	Fourth draft and first complete ST version
0.5	2016-08-25	Jens Callsen	Changes by Dencrypt A/S
0.6	2016-09-02	Staffan Persson	Finished updates
0.7	2016-09-06	Jens Callsen	Changes after internal review
0.8	2016-10-07	Jens Callsen	Update of TLS client key length
0.9	2016-10-17	Staffan Persson	Added key-pair generation and some minor corrections.
0.10	2016-10-30	Staffan Persson	Updated according to evaluator's comments
0.11	2016-11-15	Staffan Persson	Further updates based on evaluator's comments
0.12	2016-11-22	Staffan Persson	Updated sections 3, 4 and 6
0.13	2016-11-29	Staffan Persson	Updated based on further comments from the evaluator
0.14	2016-12-09	Staffan Persson	Added the name of guidance, fixed some inconsistencies in encryption modes and hash length
0.15	2017-01-04	Staffan Persson	Updated based on evaluator's comments
0.16	2017-01-05	Søren Sennels	Product renaming
0.17	2017-03-03	Jens Callsen	More details about live chat, the Dynamic Encryption implementation, FCS_CKM.1a and FCS_CKM.1c.
0.18	2017-03-19	Staffan Persson	Updated based on evaluator and certifier comments
0.19	2017-03-30	Staffan Persson	Fixed minor typos
0.20	2017-05-11	Staffan Persson	Updated the crypto SFRs and references
0.21	2017-06-23	Staffan Persson	Updated the references to guidance
1.0	2017-09-11	Staffan Persson	Updated the TOE and guidance versions

Contents

1 Introduction

1.1 Security Target identification and organisation

Title:	Security Target for Dencrypt Talk version 4.2.794
ST Version:	1.0
Status:	Final version
Date:	2017-09-11
Sponsor:	Dencrypt A/S
Developer:	Dencrypt A/S
Keywords:	Mobile application, 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 deployed. 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.

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

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

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

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

1.2 TOE identification

The TOE is Dencrypt Talk version 4.2.794 for the iPhone.

1.3 TOE type

The TOE is a VoIP application for iPhone that offers encrypted mobile voice communication and live chat within well-defined user groups. Once installed and configured, it allows two persons to talk or chat securely, as well as allowing group calls with more than two persons. Although the VoIP application is available for both iPhone and Android, only the iPhone version is considered to be the TOE and evaluated.

1.4 TOE overview

The Dencrypt Talk is a component in the Dencrypt Communication Solution. The TOE is an App, running on an iPhone. It is a VoIP client providing end-to-end voice encryption and live chat between iPhones.

The main security features of the TOE and its operational environment are:

- Encrypted end-to-end voice calls over VoIP (Secure Call)
- Encrypted live chat (Secure Live Chat)
- Encrypted group calls
- Secure Individual phone book
  - Centrally managed (TOE environment)
  - Pushed seamlessly to user devices
  - Supports individual groups settings
- Encrypted calls are restricted to the phone book
- Support secure provisioning to set up a new Dencrypt Talk installation
- Support its own key-pair generation

The Dencrypt Communication Solution consists of Dencrypt Talk (the TOE) and the Dencrypt Server System, which contains: a Dencrypt Communication Server (a SIP server), a Dencrypt Database Server (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 Talk 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 Server System 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 Talk and the Dencrypt Communication Solution is to provide mobile devices with a secure end-to-end voice communication (Secure Call) and live chat (Secure Live Chat) within closed user groups that are centrally managed.

1.5.2 The TOE architecture and key functions

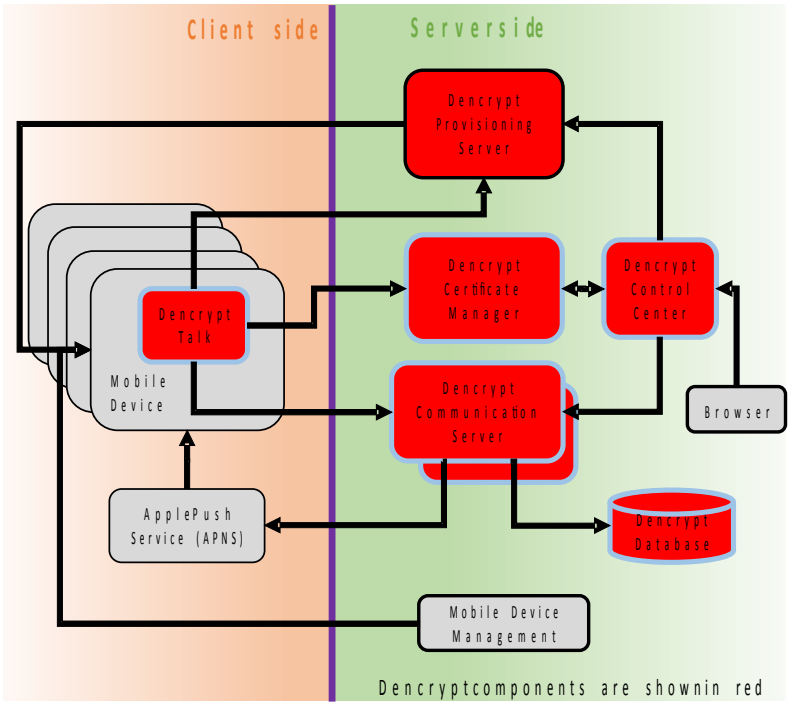

Illustration 1, The Dencrypt Communication Solution Overview

1.5.2.1 Introduction

The functionalities of the main components are described in more details below:

Dencrypt Talk

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 Dencrypt 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. The Mobile Client is the TOE.

Dencrypt Provisioning Server

The Provisioning Server 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

The Communication Server provides the SIP Services that are necessary for the Clients to establish voice and live chat communication between clients.

Dencrypt Database

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

Dencrypt Control Center

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

Dencrypt Certificate Manager (DCM) is the central point for TLS certificates in the system. Once provisioning has taken place, all connections between the Dencrypt Talk and server components 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. 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 the TOE has been successfully provisioned.

1.5.2.2 Provisioning and user registration process

The provisioning process consists of two independent steps:

- Installation of the Dencrypt Talk (the TOE)
- Provisioning of the provisioning data to the TOE 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 Communication Solution 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 the TOE. Additionally, there might be a separation of duties between MDM administrators and Dencrypt Talk administrators. Thus, the Dencrypt Communication Solution provides its own provisioning server to facilitate the initial configuration of the Dencrypt Talk.

Provisioning is started by the Dencrypt administrator adding the user to the Dencrypt Communication Solution 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 the TOE. The TOE 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.

The settings and phone book of the Dencrypt Talk (TOE) are updated as described in chapter “Managing settings and phone book”. After that the TOE is fully setup and operational.

1.5.2.3 Managing settings and phone book

The Dencrypt Talk only allows calls to persons listed in the Dencrypt Talk phone book. The phone book is individual for each user and contains only the persons which a user is allowed to call. Thus, each user may have a different phone book. The user administrator (TOE environment) can change the groups of users to whom a specific user can call to at any time.

The DCS (also TOE environment) takes care of distributing the phone book to the individual TOE users. When a user starts the TOE, the TOE establishes a TLS connection to the Dencrypt Communication Server (DCS) and makes a SIP registration. When registration is successful, the client will subscribe for phone book changes. Right after subscription and if the phone book has been changed, DCS notifies the client about the current phone book version. The client

downloads the phone book if its currently used phone book version does not match the advertised phone book version. Note, if the client has no phone book, it is considered as phone book version 0. The same method applies for settings distribution. The following figure displays the described process.

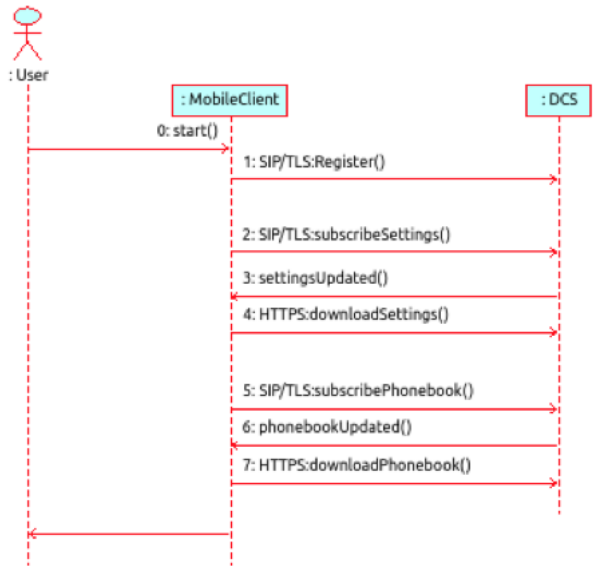

Illustration 2, The User registration process

1.5.2.4 Making a secure call

The uniqueness of this TOE is that the end-to-end encrypted voice and live chat use dynamic encryption, which ensures that each call session is encrypted with an additional layer using a randomly chosen algorithm parameters (S-boxes) and randomly chosen keys.

The following figure illustrates the steps for a secure call.

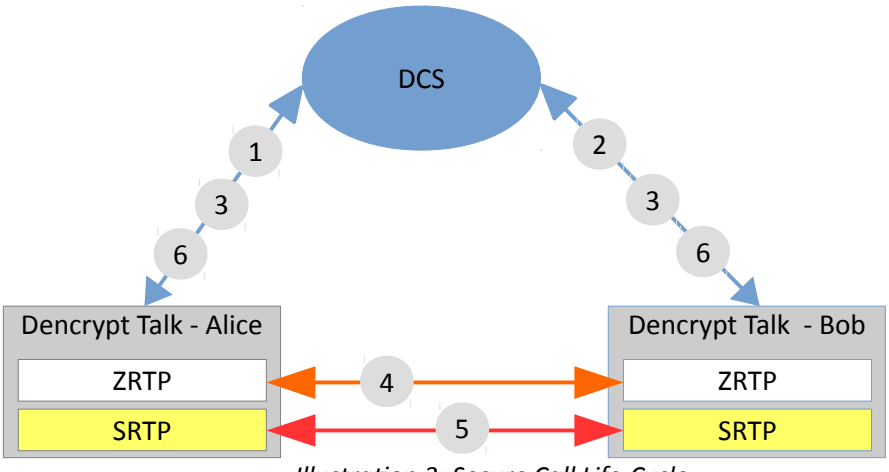

Illustration 3: Secure Call Life-Cycle

1. Alice's Dencrypt Talk contacts the DCS she is registered to.
2. The SIP server resolves Bob's address and contacts Bob's Dencrypt Talk. This resolution is limited because Alice can only contact the DCS for users listed in her phone book.

3. SIP takes care of signalling, i.e. triggers Bob's Dencrypt Talk to start ringing. As soon as Bob accepts the call, both Dencrypt Talks 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. 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 are erased.

The following list characterises the secure call in the TOE:

- Dynamic encryption of voice data is implemented as multiple layers of encryption optimized for voice data over the SRTP protocol.
- The voice stream is bidirectional, i.e. each direction uses different en/decryption keys.
- The TOE 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 uses the RNG of iOS (TOE environment).

1.5.2.5 Making a secure live chat

Secure live chat can be established during a secure audio call or as a chat only connection without audio. For both scenarios, a secure call is established first so that ZRTP establishes the different keys. In case of chat only, the audio stream continues but microphone and loudspeaker are muted and the UI does not allow to interact with the audio UI. The following list characterises the secure live chat:

- Dynamic encryption of live chat data is implemented as multiple layers of encryption optimized for text over SIP.
- The live chat stream is bidirectional, i.e. different directions use different de/encryption keys.

- Live chat uses the same keys as secure audio. Thus, also chat-only establishes a secure audio call and key exchange is established by ZRTP over SRTP
- In case of live-chat-only, the audio stream is hidden for the user but is still secured by Dynamic Encryption.
- Live chat data are secured by Dynamic Encryption and then transported by SIP messages where SIP is secured by TLS.
- ZRTP 's SAS cannot be confirmed in a chat-only scenario.

1.5.2.6 The TLS connection

All connections made between the TOE and any other component, including the one with the SIP server, are established using a trusted channel that is implemented using TLS version 1.2. The exception is the VoIP connection between the TOE and another instance of the TOE. This connection is encrypted using dynamic encryption as described in the previous section. Note that all TLS connections are initiated by the TOE and never by the server backend, such as the SIP server. The TOE is keeping the TLS connection alive as long as the TOE is running. It is necessary for being able make and receive calls and live chats.

The TLS connection is mutually authenticated to ensure both that only authorized instances of the TOE can establish connections, i.e. to retrieve the phone book information, and that the TOE is not connecting to a server system that may deceive the TOE user with false phone books or provisioning data. The TLS connection also ensures the confidentiality and integrity of any data transmitted. The TLS connection is always initiated by the TOE and never by the DCS, DCM or DPS.

Please note that the TLS connection to the provisioning web server is not mutually authenticated because the TOE is not yet configured and has no client key or certificate.

Details of the protocols and cipher suites used are provided in the TOE Summary Specification in chapter 7.

1.5.3 Security functions

This section provides a summary of the security functions implemented by the TOE. These security functions have previously been described in the previous section.

- Encrypted end-to-end voice calls over VoIP (Secure Call)
- Encrypted live chat over VoIP (Secure Live Chat)
- Encrypted group calls
- Secure individual phone book
  - Centrally managed (TOE environment)
  - Pushed seamlessly to user devices
  - Supports individual groups settings
- Encrypted calls are restricted to the phone book
- Support secure provisioning to set up a new Dencrypt Talk installation
- Support its own key-pair generation

1.5.4 Physical scope of the TOE

The TOE is limited to the Dencrypt Talk and user documentation. The following documentation is provided to the users:

- *Operational User Guide Dencrypt Talk v. 4.2 (iOS)*
- *Preparative Guide Dencrypt Talk v. 4.2 (iOS)*

The TOE is delivered to the user as an in-house app by the MDM system that is controlled by the user's organization. The user installs and configures the app by following the instructions given in the user documentation.

1.5.4.1 IT environment

The TOE environment consists of the IT components that make up the mobile devices but are outside of the TOE as well as any IT components that are outside of mobile device.

The IT components that are outside of the TOE but part of the mobile devices are:

- The mobile device hardware (iPhone) and iOS software on which the TOE is installed

The IT environment must contain the following:

- The mobile device (iPhone) where the TOE is installed
- Any additional mobile devices where the TOE is also installed (to have another party to securely communicate with)
- The Dencrypt Server System with DPS, DCC, DCM, DDB and DCS, as well as any standard MDM system. An MDM provides a local application store server for offer the Dencrypt Talk and other approved signed applications.

The Dencrypt Server System must be used and operated by administrators that are trustworthy and have been sufficiently trained to use and carry out the security management tasks in a proficient manner. The TOE users must be trustworthy and trained and are expected to follow instructions.

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 EAL4 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. For example, mobile devices will be more exposed to attackers than systems in a well-guarded environment, but exposure through communication channels may jeopardize even systems guarded in secure vaults.

Since the architecture addressed by the TOE specified in this ST includes systems where both the attack potential and the value of the assets are likely to be high, a sufficient level of assurance must be selected to provide system users with appropriate assurance that the system will be able to withstand such threats.

The TOE is expected to provide assurance to ensure separation of compartments, which requires a level of assurance that includes the evaluation of side channels between different compartments.

The EAL4 level was also deemed appropriate because this will provide a necessary assurance for an encrypted communication service , such as secure voice and live chat.

3 Security problem definition

A mobile device may be used in different ways. A device, where the TOE is installed, must be under significant enterprise control over the configuration and software inventory. The enterprise elects to provide users with mobile devices and control the configuration as well as the set of applications that can be installed in order to maintain a high degree of control of their enterprise data and security of their networks.

It is assumed that the TOE is under physical control of the user and that the users are trained and trusted to handle the TOE and to access to the enterprise data and services they are given access to. Although the users are assumed to be trustworthy and trained, we cannot exclude that mistakes are being made.

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 are entities such as unauthorized individuals or authorized users that are trying to act outside of their authorization or any other entities acting on behalf of unauthorized users, such as users. 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.

The term *threat agent* is used to indicate that a threat can be performed by an unauthorized external entity, an authorized external entity or an untrusted app. Threat agents are assumed to have moderate level of expertise, resources and motivation.

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

Threat	Description
T.DATA	An unauthorized user or attacker will gain access to user credentials, TOE settings or phone book entries to which they are not authorized.
T.MASQUERADE	A user within a closed user group is masquerading, pretending to be another user to mislead the receiver that a secure voice call or a secure chat is originating from another user belonging to the phone book of that user group.
T.TRAFFIC	An attacker (including network operators) may gain access (disclosure or modification) to secure voice or chat conversations between users within a closed user group.

3.2 Organisational security policies

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

OSP	Description
OSP.CLOSED	The TOE shall ensure that secure calls and secure chats are restricted to parties defined into the phone book held by the TOE.
OSP.FORWARD	The TOE must be able to prevent an unauthorized user that obtains a

OSP	Description
	handset to decrypt previously transmitted traffic (voice or chat) that has been encrypted using the obtained handset.
OSP.PRIVATEKEY	The TOE must be able to generate it's own private-public key pairs.
OSP.MANAGE	The TOE shall allow secure provisioning and remote update of certificates and phone book.
OSP.PHONEBOOK	The TOE must ensure that the phone book cannot be changed locally.
OSP.UPTODATE	The TOE must ensure that the phone book held by the TOE is up-to-date.

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.ADMIN	It is assumed that the TOE administrators (i.e the administrators using the Dencrypt Server System) are trustworthy and trained to perform the actions required by them for the management and maintenance of the Dencrypt Server System.
A.APPS	It is assumed that only approved, benign applications are running on the handset where the TOE is running.
A.BACKEND	It is assumed that the underlying hardware, firmware (BIOS and device drivers) and software of the server system used by the TOE are working correctly and have no undocumented security critical side effect on the TOE. Furthermore, the server system is operated in a physically secure and well managed environment.
A.HANDSET	It is assumed that the functions in the TOE environment related to memory management, program execution, access control and privilege management provided by the underlying iOS of the handset and the SIM card, work correctly and have no undocumented security critical side effects on the security functions of the TOE.
A.KEYS	It is assumed that random bits provided by the underlying platform are of good quality and have sufficient entropy.
A.SINGLEUSER	It is assumed that the TOE is under the physical control of a single authorized user.
A.USER	It is assumed that the users are trustworthy and trained to perform their actions in accordance with their instructions and security policies.
A.PROVISIONING	It is assumed that the operational environment ensures that the web link is not predictable, only active for a limited time and that access to the link is limited to one attempt only. It is also assumed that the operational environment provides the link to clients in a secure way so that the link is not disclosed to any potential attacker. Note: The link might be disclosed for the user's organisation, e.g. the link might be in cleartext on the organisation's local mail server.

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.CALLERID	The TOE must ensure that the end point of a secure call or chat connection is unique and that the caller display name associated with the caller identity is correctly shown to the TOE user making or receiving the secure call or live chat message.
O.GROUP	The TOE must ensure that secure calls and chats are restricted to users within the TOE phone book.
O.TRAFFIC	The TOE must ensure that secure calls and chats are protected against disclosure and modification.
O.CHANNEL	The TOE must ensure that there is a trusted path between the TOE and the Dencrypt Server System ensuring authenticity, confidentiality and integrity of any TSF or user data transmitted between the TOE and the server system, such as phone book updates and SIP connections made when establishing secure calls.
O.PHONEBOOK	The TOE must ensure that the phone book cannot be changed locally.
O.FORWARD	The TOE must ensure that an unauthorized user that obtains a handset cannot decrypt previously transmitted traffic (voice or chat) that has been encrypted using the obtained handset.
O.PRIVATEKEY	The TOE must be able to generate it's own private-public key pairs.
O.MANAGE	The TOE must support provisioning and ensure that settings and phone book can be updated whenever the TOE is running and registered on the DCS.

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.ADMIN	The operational environment shall ensure that the TOE administrators (i.e the administrators using the server system) are trustworthy and trained to perform the actions required by them for the management and maintenance of the Dencrypt Server System.
OE.APPS	The operational environment shall ensure that only approved, benign applications are running on the handset where the TOE is running.
OE.BACKEND	The operational environment shall ensure that the underlying hardware, firmware (BIOS and device drivers) and software of the Dencrypt Server System system used by the TOE are working correctly

Security Objective	Description
	and have no undocumented security critical side effect on the TOE. The operational environment shall also ensure that the server system is operated in a physically secure and well managed environment.
OE.HANDSET	The operational environment shall ensure that the functions in the TOE environment related to memory management, program execution, access control and privilege management provided by the underlying iOS of the handset and the SIM card, work correctly and have no undocumented security critical side effects on the security functions of the TOE.
OE.KEYS	The operational environment shall ensure that random bits provided by the underlying platform are of good quality and have sufficient entropy.
OE.SINGLEUSER	The operational environment shall ensure that the TOE is under the physical control of a single authorized user.
OE.USER	The operational environment shall ensure that the users are trustworthy and trained to perform their actions in accordance with their instructions and security policies.
OE.PROVISIONING	The operational environment shall ensure that the web link is not predictable, only active for a limited time and that access to the link is limited to one attempt only. It shall also provide the link to clients in a secure way so that the link is not disclosed to any potential attacker. Note: The link might be disclosed for the user's organisation, e.g. the link might be in cleartext on the organisation's local mail server.

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.DATA	T.MASQUERADE	T.TRAFFIC	OSP.CLOSED	OSP.FORWARD	OSP.PRIVATEKEY	OSP.MANAGE	OSP.PHONEBOOK	OSP.UPTODATE	A.ADMIN	A.APPS	A.BACKEND	A.HANDSET	A.KEYS	A.SINGLEUSERS	A.USER	A.PROVISIONING
O.CALLERID		X															
O.GROUP				X													
O.TRAFFIC			X														
O.CHANNEL	X	X															
O.PHONEBOOK				X				X									
O.FORWARD					X												
O.PRIVATEKEY						X											

	T.DATA	T.MASQUERADE	T.TRAFFIC	OSP.CLOSED	OSP.FORWARD	OSP.PRIVATEKEY	OSP.MANAGE	OSP.PHONEBOOK	OSP.UPTODATE	A.ADMIN	A.APPS	A.BACKEND	A.HANDSET	A.KEYS	A.SINGLEUSERS	A.USER	A.PROVISIONING
O.MANAGE							X		X								
OE.ADMIN										X							
OE.APPS											X						
OE.BACKEND												X					
OE.HANDSET													X				
OE.KEYS														X			
OE.SINGLEUSER															X		
OE.USER																X	
OE.PROVISIONING							X										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.DATA	This threat is addressed by O.CHANNEL that ensures that there is a trusted path between the TOE and the server system ensuring authenticity, confidentiality and integrity of any TSF or user data transmitted between the TOE and the server system, such as phone book updates and SIP connections made when establishing secure calls.
T.MASQUERADE	This threat is addressed by O.CHANNEL that enforces client authentication and by O.CALLERID that ensures that the end point of a secure call or chat connection is unique and that the caller display name associated with the caller identity is correctly shown to the TOE user making or receiving the security call or chat message.
T.TRAFFIC	This threat is addressed by O.TRAFFIC that ensures that secure calls and chats are protected against disclosure and modification.

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.CLOSED	This OSP is addressed by O.GROUP that ensures that secure calls and chats is restricted to users within the TOE phone book and by O.PHONEBOOK that ensures that the phone book cannot be changed locally.
OSP.FORWARD	This OSP is addressed by O.FORWARD that ensures that an unauthorized user that obtains a handset cannot decrypt previously transmitted traffic (voice or messages) that has been encrypted using the obtained handset.

OSP	Rationale for the security objectives
OSP.PRIVATEKEY	This OSP is addressed by O.PRIVATEKEY that ensures that the key pair for the TOE is generated by the TOE. This is supported by OE.KEYS that provides the random number for the key generation.
OSP.MANAGE	This OSP is addressed by O.MANAGE that ensures that settings and phone book can be updated whenever the TOE is running and registered on the DCS. The TOE provides management capabilities and ensuring that they are restricted to authorized subjects. The secure provisioning is supported by OE.PROVISIONING.
OSP.PHONEBOOK	This OSP is addressed by O.PHONEBOOK that ensures that phone book cannot be changed locally.
OSP.UPTODATE	This OSP is addressed by O.MANAGE that ensures the phone book is updated whenever the TOE is running and registered on the DCS.

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.ADMIN	Addressed by OE.ADMIN, which is identical to the assumption
A.APPS	Addressed by OE.APPS, which is identical to the assumption
A.BACKEND	Addressed by OE.BACKEND, which is identical to the assumption
A.HANDSET	Addressed by OE.HANDSET, which is identical to the assumption
A.KEYS	Addressed by OE.KEYS, which is identical to the assumption
A.SINGLEUSER	Addressed by OE.SINGLEUSER, which is identical to the assumption
A.USER	Addressed by OE.USER, which is identical to the assumption
A.PROVISIONING	Addressed by OE.PROVISIONING, 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 the FCS_TLSC (for cryptography).

These extended components have been taken directly from the extended components defined in [cPPND], the collaborative Protection Profile for Network Devices, Version 1.0, 27-Feb-2015.

6 Security requirements

6.1 Security functional policies

6.1.1 GROUP SFP

The TOE will implement an information flow control policy named **GROUP SFP**.

Policies are used to enforce security guidelines and restrict the users from undesired behaviour. The TOE is implementing the **GROUP** information flow control security functional policy, or simply **GROUP SFP**. This will ensure that the TOE does not allow any secure call or secure chat between a user and another user, unless the user is explicitly allowed to. This policy is set by the central managed phone book for each user. The phone book on the TOE is kept in sync with server system's phone book whenever the TOE is running and registered on the DCS. Secure Call and Secure Live Chats are restricted to parties in the phone book.

6.2 Security functional requirements

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

6.2.1 FCS_CKM.1a – Cryptographic key generation (SRTP/ZRTP key generation)

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **as defined in ZRTP [RFC6189] and SRTP [RFC3711] for AES-256 in CBC, CM mode and HMAC SHA-1 and HMAC SHA-256 keys** and specified cryptographic key sizes **256 bit (AES-256 and HMAC SHA-256) and 160 bit (HMAC-SHA1)** that meet the following: **[FIPS197], [NIST SP 800-38A], [RFC2104] and [FIPS180-4]**.

Application note: This SFR covers the generation of AES keys and keys for dynamic encryption that are used by FCS_COP.1d for the secure voice. The ZRTP key management protocol as specified in RFC6189 relies on Diffie-Hellman to establish keys to be used by the SRTP protocol. The SRTP protocol uses the master key provided by ZRTP to generate the key pair for encryption and integrity protection. HMAC-SHA1 is used by SRTP for Message Authentication and Integrity as specified in RFC3711, chapter 4.2. HMAC-SHA256 is used by ZRTP for Message Authentication and Integrity as specified in RFC6189, chapter 4.5.3.

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

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 SFR covers the generation of AES keys and keys for the TLS connection.

6.2.3 FCS_CKM.1c – Cryptographic key generation (RSA key generation)

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 ~~2014-bit or greater~~3072-bit that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3**.

Application note: This SFR covers the generation of the private-public key pair for the TOE. This requirement is a refinement of FCS_CKM.1.1 defined in [cPPND]. Before the client certificate expires, a new key pair is generated and the client certificate is renewed. After that, Dencrypt Talk has a new client certificate based on a new key pair.

6.2.4 FCS_CKM.2a – Cryptographic key distribution (ZRPT)

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **ZRTP** that meets the following: **RFC6189**.

Application note: This SFR covers the key distribution for secure voice and live chat. The ZRTP key agreement protocol performs a Diffie-Hellman key exchange during call setup in the media path. It generates a shared secret, which is then used to generate keys and salt for a Secure RTP (SRTP) [RFC3711] session.

6.2.5 FCS_CKM.2b – Cryptographic key distribution (Client 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: This SFR covers the key distribution which ensures that the TOE has a valid X.509v3 client certificate and the private client key for TLS connections to DCM and DCS. The client generates the private/public key pair (FCS_CKM.1c), 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 TOE.

6.2.6 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. Additionally, key destruction is applied on client private key after a new private client key is taken into usage.

6.2.7 FCS_COP.1a – Cryptographic Operation (AES Data Encryption and Decryption)

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

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

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

FCS_COP.1.1 The TSF shall perform **cryptographic signature services (generation and verification)** in accordance with a specified cryptographic algorithm **RSA Digital Signature Algorithm** and cryptographic key sizes **3072 bit for generation and 4096 bit for verification** 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 signature generation and verification performed as part of the TLS server authentication performed by the TOE. This SFR applies also to the signature of the Certificate Signing Requests (CSR) send by the TOE to the DCM.

6.2.9 FCS_COP.1c – Cryptographic Operation (Hash Algorithm)

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_TLSC_EXT.2 and FCS_TLSC_EXT.1 to ensure the integrity of the TLS connection.

6.2.10 FCS_COP.1d – Cryptographic Operation (Dynamic Encryption and Decryption)

FCS_COP.1.1 The TSF shall perform **decryption and encryption** in accordance with a specified cryptographic algorithm **AES and modified AES in CBC mode for Live Chat, AES and modified AES in CM mode for Dencrypt Talk** and cryptographic key sizes **256 bit** that meet the following: **[FIPS197] and [NIST SP 800-38A] and for the modified AES [Dynamic]**.

Application note: This requirement addresses both the AES data stream and modified (dynamic) encryption and decryption of the modified S-boxes as specified in the dynamic encryption for secure voice and live chat.

6.2.11 FCS_COP.1e – Cryptographic Operation (HMAC-SHA1 / HMAC-SHA256)

FCS_COP.1.1 The TSF shall perform **keyed-hash message authentication** in accordance with a specified cryptographic algorithm **HMAC-SHA1, HMAC-SHA256** and cryptographic key sizes **160 bit (HMAC-SHA1), 256 bit (HMAC-SHA256) and message digest sizes 160 bit (HMAC-SHA1), 256 bits (HMAC-SHA256)** that meet the following: **RFC2104**.

Application note: This requirement addresses the HMAC used by the SRTP.

6.2.12 FCS_TLSC_EXT.1 – TLS Client Protocol

FCS_TLSC_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_TLSC_EXT.1.2 The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.1.3 The TSF shall only establish a trusted channel if the peer certificate is valid.

FCS_TLSC_EXT.1.4 The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: **secp384r1** and no other curves.

Application note: The unauthenticated TLS connection is only used for the provision connection of the TOE.

6.2.13 FCS_TLSC_EXT.2 – TLS Client Protocol with Authentication

FCS_TLSC_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**

FCS_TLSC_EXT.2.2 The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.2.3 The TSF shall only establish a trusted channel if the peer certificate is valid.

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

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

Application note: The ciphersuites used for the DPS webAPI connection, the DCS webAPI connection and the DCM webAPI connection are the same. The authenticated TLS connection is for all TLS connections with exception of the provisioning TLS connection, i.e. for the initial configuration of the TOE.

6.2.14 FDP_IFC.2 – Complete information flow control

FDP_IFC.2.1 The TSF shall enforce the **GROUP information flow control SFP** on **TOE instances and voice data and live chat data** and all operations that cause that information to flow to and from subjects covered by the SFP.

FDP_IFC.2.2 The TSF shall ensure that all operations that cause any information in the TOE to flow to and from any subject in the TOE are covered by an information flow control SFP.

Application note: The calls are restricted to the phone book entries held by the TOE. It is not possible for users to add, delete or modify these entries. The phone book is centrally managed.

6.2.15 FDP_IFF.1 – Simple security attributes

FDP_IFF.1.1 The TSF shall enforce the **GROUP information flow control SFP** based on the following types of subject and information security attributes: **TOE instances (user handsets) and user name**.

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: **A secure call or live chat can only be established from a TOE to another instance of the TOE when the name of the user of the second TOE instance (callee) is in the phone book of the caller**.

FDP_IFF.1.3 The TSF shall enforce **no additional rules**.

FDP_IFF.1.4 The TSF shall explicitly authorise an information flow based on the following rules: **none**.

FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: **none**.

Application note: A user's phone book only contains the names of the other users he/she is allowed to call to. Please note a pre-condition for this is that all these users have been issued a permanent certificate by the DCM. If the call receiver does not have the caller in the phone book (there are cases where a small group of users can call the users in a bigger group, but not everyone in the bigger group are allowed to call the users in the small group), the caller's display name is shown on the incoming call screen. The caller's display name is provided as metadata in call setup message, i.e. the server system defines what is displayed on the incoming call screen.

6.2.16 FMT_MTD.1 – Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to **modify** the **network settings and phone book** to **the remote DCS connection**.

Application note: The TOE management functions are performed in the TOE environment by the administrator using the Dencrypt Control Center. The settings are then pushed to the TOE from the DCS whenever a DCS connection is made. The TOE user is not allowed to modify the network settings and phone book on the TOE.

6.2.17 FMT_SMF.1 – Specification of management functions

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

- **download network settings and phone book**
- **replace the existing network settings and phone book with the downloaded versions**

Application note: The network settings and phone book on the TOE are kept in sync with server system's data whenever the TOE is running and registered on the DCS. Note: These management functions are not performed by any user by are performed automatic.

6.2.18 FTP_ITC.1a – Inter-TSF Trusted Channel (TLS)

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

FTP_ITC.1.2 The TSF shall permit the **TSF** to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for **connecting to Dencrypt Server System**.

Application note: This is the trusted channel (TLS) between the TOE and the DCS, DCM, and DPS. Note that this is only used for the TOE and not for access to an MDM system or to any corporate resources (such as Intranet or email). The cryptography is described in FCS_TLSC_EXT.1 and in FCS_TLSC_EXT.2.

Application note: The TLS connection to the provisioning web server is not authenticated by the client but the provisioning data can be established only once and for a limited time after the link has been provided and only by the TOE user that knows the unique link.

6.2.19 FTP_ITC.1b – Inter-TSF Trusted Channel (VoIP)

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

FTP_ITC.1.2 The TSF shall permit **the TSF or another instantiation of the TOE** to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for **secure voice and live chat**.

Application note: This channel is the end-to-end encrypted channel between the TOE and another instantiation of the TOE that is used for encrypted voice (VoIP) and encrypted live chat. SIP stands for the correct end-point and ZRTP's SAS ensures that the data channel is not intercepted.

6.3 Security functional requirements rationale

6.3.1 Coverage

The following table provides a mapping of SFR to the security objectives, showing that each security functional requirement addresses at least one security objective and that all security objectives are addressed by one or more SFRs.

	O.CALLERID	O.GROUP	O.TRAFFIC	O.CHANNEL	O.PHONEBOOK	O.PRIVATEKEY	O.FORWARD	O.MANAGE
FCS_CKM.1a			X					
FCS_CKM.1b			X	X				
FCS_CKM.1c						X		
FCS_CKM.2a			X					
FCS_CKM.2b			X	X				
FCS_CKM.4			X	X			X	
FCS_COP.1a			X	X				
FCS_COP.1b			X	X				
FCS_COP.1c			X	X				
FCS_COP.1d			X					
FCS_COP.1e			X					
FCS_TLSC_EXT.1								X
FCS_TLSC_EXT.2			X	X				
FDP_IFC.2	X	X						
FDP_IFF.1	X	X						
FMT_MTD.1					X			X
FMT_SMF.1					X			X
FTP_ITC.1a			X	X				
FTP_ITC.1b	X		X					

6.3.2 Sufficiency

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

Security Objective	Security objectives
O.CALLERID	The objective: • The TOE must ensure that the end point of a secure call or live chat connection is unique and that the caller display name associated with the caller identity is correctly shown to the TOE user making or receiving the secure call or live chat message. is met by:

Security Objective	Security objectives
	• FDP_IFC.2 and FDP_IFF.1 ensuring that information flow is only possible between parties represented in the caller's phone book. • FTP_ITC.1b ensuring that there is a mutually authenticated trusted channel between the two parties
O.GROUP	The objective: • The TOE must ensure that secure calls and live chats is restricted to users within the TOE phone book. is met by: • FDP_IFC.2 and FDP_IFF.1 ensuring that information flow is only possible between parties represented in the caller's phone book.
O.TRAFFIC	The objective: • The TOE must ensure that secure calls and live chats are protected against disclosure and modification. is met by: • FTP_ITC.1b that ensures that there is a trusted path for secure voice and live chat between the TOE and another instance of the TOE. • Key generation is done by FCS_CKM.1a and key distribution by FCS_CKM.2a. • Encryption is done by FCS_COP.1d, FCS_COP.1e and FCS_COP.1a. • Key destruction is performed by FCS_CKM.4. in addition O.TRAFFIC is relying on a secure channel for chat, address by: • FTP_ITC.1a that ensures that there is a trusted path between the TOE and the server system. • FCS_CKM.2b ensures that the TOE obtains a valid client certificate. • Key generation is done by FCS_CKM.1b and key distribution is done by FCS_CKM.2b as part of the TLS 1.2 protocol implemented by FCS_TLSC_EXT.2. • Encryption is done by FCS_COP.1a. • Authentication is ensured by FCS_COP.1b. • Integrity is ensured by FCS_COP.1c. • Key destruction is performed by FCS_CKM.4.
O.CHANNEL	The objective: • The TOE must ensure that there is a trusted path between the TOE and the server system ensuring authenticity, integrity and confidentiality and integrity of any TSF or user data transmitted between the TOE and the server system, such as phone book updates and SIP connections made when establishing secure calls. is met by: • FTP_ITC.1a that ensures that there is a trusted path between the TOE and the server system. • FCS_CKM.2b ensures that the TOE obtains a valid client certificate. • Key generation is done by FCS_CKM.1b and key distribution is done by FCS_CKM.2b as part of the TLS 1.2 protocol implemented by FCS_TLSC_EXT.2. • Encryption is done by FCS_COP.1a. • Authentication is ensured by FCS_COP.1b.

Security Objective	Security objectives
	• Integrity is ensured by FCS_COP.1c. • Key destruction is performed by FCS_CKM.4.
O.PHONEBOOK	The objective: • The TOE must ensure that the phone book cannot be changed locally. is met by: • FMT_SMF.1 ensures that the network settings and the phone book can be managed • FMT_MTD.1 ensures that this is restricted to the remote network connection to the DCS
O.PRIVATEKEY	The objective: • The TOE must be able to generate it's own key pair. is met by: • FCS_CKM.1c ensures that the private/public key pair for the TOE is generated by the TOE itself.
O.FORWARD	The objective: • The TOE must ensure that an unauthorized user that obtains a handset cannot decrypt previously transmitted traffic (voice or messages) that has been encrypted using the obtained handset. is met by: • FCS_CKM.4 ensures that symmetric keys used for encryption and decryption of secure voice and live chat are destroyed after a secure voice call or live chat is completed.
O.MANAGE	The objective: • The TOE must support provisioning and ensure that settings and phone book can be updated whenever the TOE is running and registered on the DCS. is met by: • FMT_SMF.1 ensures that the network settings and the phone book can be managed. • FMT_MTD.1 ensures that this is restricted to remote management. • FCS_TLSC_EXT.1 ensures that the initial credentials are provisioned to the TOE by a trusted server system and the credentials are protected by TLS during transmission.

6.3.3 Dependency analysis between security functional components

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

SFR	Dependencies	Resolved?
FCS_CKM.1a	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	Yes, by FCS_CKM.2a and FCS_COP.1d Yes, by FCS_CKM.4
FCS_CKM.1b	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	Yes, by FCS_COP.1a Yes, by FCS_CKM.4

SFR	Dependencies	Resolved?
FCS_CKM.1c	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	Yes, by FCS_CKM.2b Yes, by FCS_CKM.4
FCS_CKM.2a	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	Yes, by FCS_CKM.1a Yes, by FCS_CKM.4
FCS_CKM.2b	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	Yes, by FCS_CKM.1c Yes, by FCS_CKM.4
FCS_CKM.4	No dependencies	--
FCS_COP.1a	[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	[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 for importing certificates. No, no key destruction needed.
FCS_COP.1c	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	No, no key is needed for SHA-384 No, no key destruction needed
FCS_COP.1d	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	Yes, by FCS_CKM.1a Yes, by FCS_CKM.4
FCS_COP.1e	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	Yes, by FCS_CKM.1a Yes, by FCS_CKM.4
FCS_TLSC_EXT.1	FCS_COP.1(1) FCS_COP.1(2) FCS_COP.1(3) FCS_RBG_EXT.1	Yes, by FCS_COP.1a Yes, by FCS_COP.1b Yes, by FCS_COP.1c No, but addressed by OE.KEYS
FCS_TLSC_EXT.2	FCS_COP.1(1) FCS_COP.1(2) FCS_COP.1(3) FCS_RBG_EXT.1	Yes, by FCS_COP.1a Yes, by FCS_COP.1b Yes, by FCS_COP.1c No, but addressed by OE.KEYS
FDP_IFC.2	FDP_IFF.1	Yes, by FDP_IFF.1
FDP_IFF.1	FDP_IFC.1 FMT_MSA.3	Yes, by FDP_IFC.2 No, attributes are static and assigned as part of the

SFR	Dependencies	Resolved?
		FMT_MTD.1
FMT_MTD.1	FMT_SMR.1 FMT_SMF.1	No, management functions are performed by the backend connection to the DCS and the TOE. Yes, by FMT_SMF.1
FMT_SMF.1	No dependencies	–
FTP_ITC.1a	No dependencies	–
FTP_ITC.1b	No dependencies	–

6.4 Security assurance requirements

The security assurance requirements of this Security are those defined for the assurance level EAL4 augmented with ALC_FLR.2.

Assurance class	Assurance components
ADV: Development	ADV_ARC.1 Security architecture description
	ADV_FSP.4 Complete functional specification
	ADV_IMP.1 Implementation representation of the TSF
	ADV_TDS.3 Basic modular design
AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
ALC: Life-cycle support	ALC_CMC.4 Production support, acceptance procedures and automation
	ALC_CMS.4 Problem tracking CM coverage
	ALC_DEL.1 Delivery procedures
	ALC_DVS.1 Identification of security measures
	ALC_FLR.2 Flaw reporting procedures (augmentation)
	ALC_LCD.1 Developer defined life-cycle model
	ALC_TAT.1 Well-defined development tools
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.2 Analysis of coverage
	ATE_DPT.1 Testing: basic design
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing – sample

Assurance class	Assurance components
AVA: Vulnerability assessment	AVA_VAN.3 Focused vulnerability analysis

6.5 Security assurance requirements rationale

Dependencies within the EAL package selected (EAL4) 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 assurance level EAL4 augmented with ALC_FLR.2 has been chosen since EAL4 is the lowest assurance package which includes source-code analysis. The source code analysis is necessary to assess the implementation quality and ensure that the TOE does not contain any malicious code.

EAL4 is augmented by ALC_FLR.2 as during operations new vulnerabilities may be discovered, either through developer actions (e.g., developer testing) or those discovered by others. It requires the developer to have procedures addressing theses vulnerabilities. The process used by the developer corrects any discovered vulnerabilities and performs an analysis to ensure that no new vulnerabilities are created while fixing the discovered ones.

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.PROVISIONING	FTP_ITC.1a FMT_MTD.1
SF.MANAGEMENT	FCS_CKM.1c (private-public key generation) FCS_COP.1b (for certificate signing request) FCS_CKM.2b (client public key) FMT_MTD.1 FMT_SMF.1
SF.MESSAGING	FCS_CKM.1a (used for ZRTP) FCS_CKM.2a (used by ZRTP) FCS_COP.1d (AES + mod. AES) FCS_COP.1e (HMAC-SHA1 used by SRTP) FCS_CKM.4 FDP_IFC.2 (GROUP) FDP_IFF.1 (GROUP) FTP_ITC.1b
SF.CHANNEL	FCS_COP.1a (AES for TLS) FCS_COP.1b (Sign. Verificat.) FCS_COP.1c (Hashing) FCS_CKM.1b (used by TLS) FCS_CKM.2b (client public key) FCS_TLSC_EXT.1 FCS_TLSC_EXT.2 FTP_ITC.1a

7.1 SF.PROVISIONING – Secure initialisation

The provisioning process consist of two steps, the installation of the Dencrypt Talk and the provisioning of the user credentials. The installation of the Dencrypt Talk is not part of the TSF, but the provisioning of the user credentials to the TOE is, which is described here.

Provisioning starts by the Dencrypt administrator by adding the user to the Dencrypt Server System. Afterwards, the administrator provides an invitation link to the user. This link must be provided in a secure way to the user, i.e. the link is not disclosed during transmission to the TOE user. The invitation link might be mailed to the TOE if the mail transmission between mail server and handset's mail client is encrypted and the mail server is controlled by the organisation of the TOE's user. 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 invitation link points to the web server of the DPS. This is all done in the TOE environment.

The user shall tap the invitation link which starts the TOE. The TOE parses the link, fetches the provisioning data from the DPS and installs the provisioning 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.

The network settings and phone book of the Dencrypt Talk (TOE) are updated as described in chapter “Managing settings and phone book”. After that the TOE is fully setup and operational.

7.2 SF.MANAGEMENT – Update of TOE settings, phone book and certificate

When a SIP registration is successful, the client will subscribe for changes of network settings. This allows the TOE to keep its local settings in sync with the server system's settings. Whenever the TOE is running and registered on the DCS, the TOE downloads network settings as soon as the checksum of its local settings differs from the settings checksum advertised by the DCS. The same mechanism applies to keep the phone book up-to-date.

The security settings that are updated are the:

- Network settings
- Phone book

Note: These security settings are managed only by remote administrators that are identified and authenticated by the TOE environment of the Dencrypt Server System The TOE simply downloads the settings when new versions become available. Although the TOE user is the single individual that is authorized to use the TOE, the user cannot change any phone book entries or make calls from a dial-pad:

- This minimizes the risk of a phishing attack.
- This keeps the administrator in full control of the phone book content.
- This adds a layer of security because only people that have been given permission can call you using the Dencrypt Talk.

Besides, the mobile’s built-in phone book is kept completely separate from the Dencrypt Talk’s phone book. Hence, removing Dencrypt Talk from the mobile phone does not leave “traces” in the built-in phone book and deploying a new mobile simply means to provision the user on a new phone.

Apart from phone book updates, the TOE will generate by itself an RSA 3072-bit private-public key pair and send its public key with a certificate signing request (CSR) to the DCM server which signs it and delivers the client certificate. This is to ensure that the private key never must leave the TOE.

The client creates a new private key and CSR if the client certificate expires soon (e.g. within the next 2 months). Since the TLS connection to the DCM also requires a client authentication, the client is provisioned with a temporary client certificate and private key. With this set of temporary key and certificate, the SIP client can connect to the DCM and send a CSR after creation of a new private key.

7.3 SF.MESSAGING – Secure voice and live chat

The secure communication channel between two handsets, the TOE and other instance of the TOE goes as followed: Dencrypt's secure voice extends a Voice-over-IP (VoIP) system by a patent-pending dynamic encryption for voice data. Dencrypt's VoIP system employs the Session Initiated Protocol (SIP), Secure Realtime Transmission Protocol (SRTP – RFC 3711) and ZRTP – RFC 6189 standard components that are partly modified to integrate dynamic encryption.

The core elements of the SIP VoIP system are the SIP clients and the SIP server. The SIP server is the signalling center of the communication system, e.g. contacting the SIP client for call, giving the start signal for the waiting tone of the calling party and terminating the call as soon as one side is releasing the call. Hence, the SIP session is named “call session”. SIP transmissions are protected by mutually authenticated TLS. The SIP VoIP system is designed for the IP network, i.e. Dencrypt

Communication Solution's can be deployed where an IP network connection exits but is limited to Dencrypt Communication Solution users.

The SRTP components are implemented in the SIP clients and secure the audio data stream. SRTP is the secure variant of RTP. (S)RTP sessions are named media or content sessions. The ZRTP protocol is used to establish the keys that are needed by Dynamic Encryption of the call and live chat. The following figure displays the relationship between different keys for Dynamic Encryption in Dencrypt Talk.

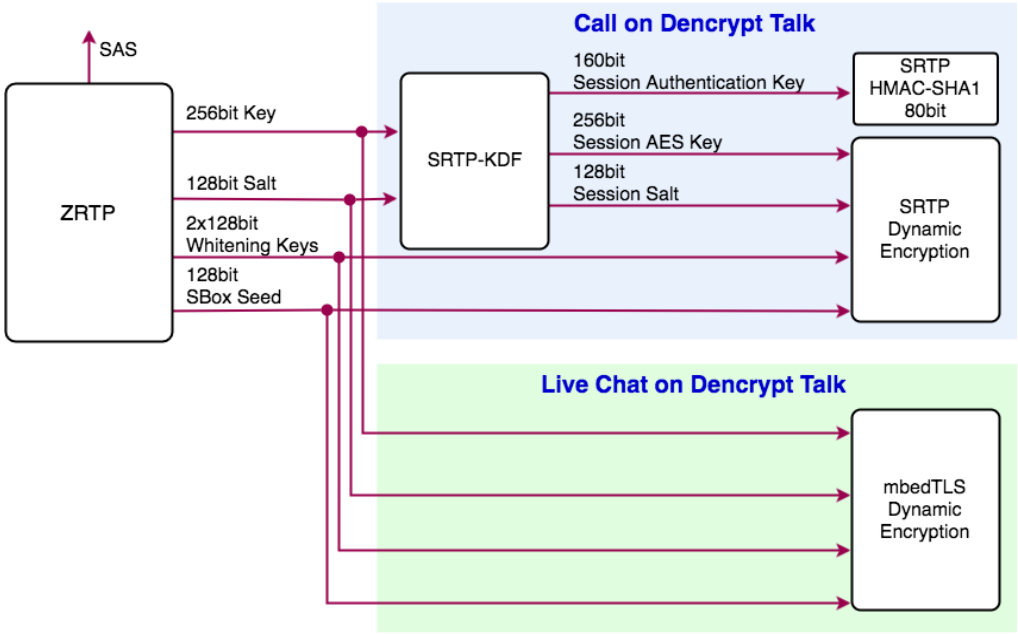

Illustration 4, The Encryption Key Dependencies

More specifically, SRTP/ZRTP uses Diffie-Hellman key exchange to establish a common secret. The common secret is then hashed into a 4-letter phrase called Short Authentication String (SAS). If this SAS is the same for both sides, the connection is authenticated (i.e. no man-in-the-middle attack has occurred) and the equality of the negotiated common secret is confirmed. The common secret can now be used to derive a 256bit key and 128bit salt which is fed into SRTP's key derivation function (KDF). As specified in RFC3711, the KDF generates a message authentication session key, an AES session key and a session salt session keys. To complete the Dynamic Encryption input, the additional parameters: Two whitening keys and the S-Box seed are provided by Dencrypt's modified ZRTP implementation directly without a session calculation by the KDF. Once the SRTP session ends all the keys are destroyed (overwritten with zeros).

In Dencrypt's case, the use of SRTP is limited to the call, i.e. audio. Dencrypt Talk's live chat messages uses SIP messaging as transportation protocol where the chat text is ciphered by Dynamic Encryption. As shown in the figure above, both call's and live chat keys originate from the same keys provided by ZRTP. In case of live chat, the keys, salt and S-Box seed are deployed direct. Once the call or live chat ends, all the keys are destroyed (overwritten with zeros). But both of the content, i.e. SRTP stream for audio and text for SIP messaging, are dynamically encrypted. Details of the Dynamic Encryption are given in section 7.3.1.

Note that secure voice and live chat are restricted to users within the caller’s phone book. The phone book is centrally managed and the TOE user is not able to add, delete or modify it. If the call receiver does not have the caller in the phone book (there are cases where a small group of users can call the users in a bigger group, but not everyone in the bigger group are allowed to call the users in the small group), the caller’s display name is shown on the incoming call screen. This display name is defined by the Dencrypt Server System and is provided as metadata in the call setup message.

7.3.1 Dynamic Encryption

Dencrypt's patent-pending dynamic encryption was invented by Lars Ramkilde Knudsen. The patent is identified by its patent number WO2013060876[1].

Key elements of the patent:

1. The decryption method is not pre-defined but will be determined during the communication establishment.
2. The sender of the encrypted data determines how the receiver shall decrypt the received encrypted data.

Dynamic encryption relies on AES encryption for security, but adds a layer of obfuscation to AES.

Dencrypt's Implementation Concept

The patent itself does not define how to setup the decryption method on the receiver. Dencrypt's implementation of Dynamic Encryption adds an additional layer around the standard AES cipher. The additional layer and the standard AES cipher together form a new cipher: The Dynamic Encryption cipher. The additional layer is determined by the whitening keys and the substitution-box (S-box) where the S-box is determined by a given S-Box seed that is given before the encryption session. The S-Box generator is responsible to return a secure S-Box based on a given seed. Additional xor-ing by two different whitening keys is applied before and after AES encryption and S-Box substitution to strengthen the Dynamic Encryption cipher.

The following figure shows the described implementation.

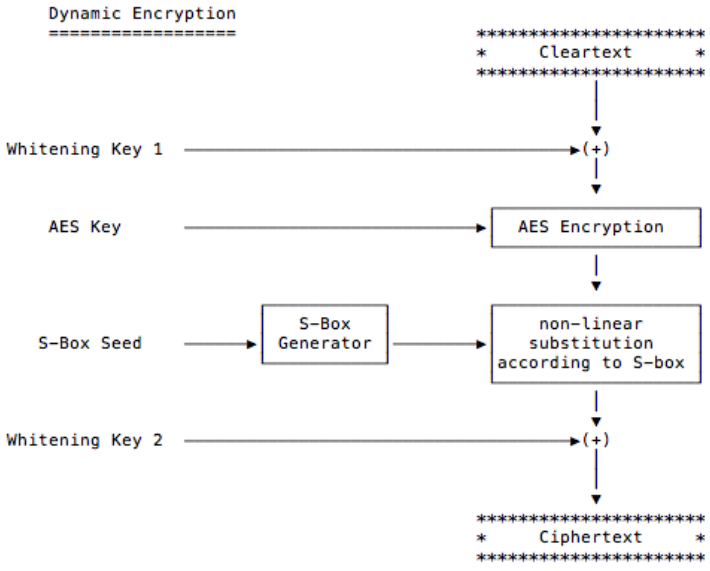

Illustration 5, The Dynamic Encryption Concept

For decryption, the inverse S-Box is calculated and the algorithm is executed backwards.

The AES Encryption in the figure above uses AES in CM/CBC mode, while the non-linear substitution according to S-box uses modified AES in CM/CBC mode as described below.

1 See also http://orbit.dtu.dk/fedora/objects/orbit:128232/datastreams/file_be4c445c-73d5-4204-8805-67743afff6bf/content

Mode of Operation

Dynamic encryption uses AES which is a block cipher that handle blocks of 128 bits. To ensure confidentiality, the blocks are interconnected. Different modes of operations are used in SRTP (voice) and Live Chat, as described below:

- Dynamic Encryption in SRTP – The mode of operation used is the Counter Mode (CM). CM allows decryption even if single blocks are not available due to transmission failures or latency[2].
- Dynamic Encryption in Live Chat – Dynamic Encryption is used both for Secure Call and Live Chat. When Live Chat is started during an ongoing audio call encryption receives and use the same keys and algorithm selection parameters as the dynamically encrypted call. Live Chat deploys SIP messages where content is plain text. The connection to the SIP server is secured by TLS as described in SF.CHANNEL and the text is dynamically encrypted.

  Live Chat runs in Cipher Block Chain[3] (CBC) mode of operation, because SIP signalling takes care of the completeness of the transmitted content and there is no realtime requirements, i.e. all blocks are available for decryption.

7.4 SF.CHANNEL – Secure communication channel (TLS)

The TOE can establish a secure channel between the TOE and Dencrypt Server System components. All TLS connections are initiated by the TOE.

The secure SIP connection between the TOE and the SIP server on the Dencrypt Communication Server (DCS) uses mutually authenticated TLS 1.2, with RSA signature verification, AES 256-bit encryption and SHA-384 hashing.

The HTTPS connection to the web server (webAPI) on the Dencrypt Provisioning Server (DPS) also uses TLS 1.2, with RSA signature verification, AES 256-bit encryption and SHA-384 hashing. However in this case the TOE is not authenticated to the DPS (i.e. only the server side of the TLS channel is authenticated). This is only performed once during the provisioning and the connection is only activate with a limited time after the link has been provided.

For TLS 1.2 mutual authentication, the TOE has a 3072-bit RSA key pair while each server component has a 4096-bit RSA key pair. So the TOE shall verify server signature by using the server system's 4096-bit RSA public key.

7.5 Cryptographic functions and parameters

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

Dynamic Encryption	Used by **SF.MESSAGING** Uses AES-256 128 bit seed for S-Box generation, i.e. 128 bit for algorithm selection 2 x 128 bit Whitening keys (xor'ing) Used by SRTP Used by Live Chat
SRTP (RFC 3711)	Used by **SF.MESSAGING** Modified by Dencrypt to support dynamic encryption Hash: HMAC-SHA1 Key derivation function (KDF) of session keys: AES

2 See https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#Counter_.28CTR.29

3 See https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#Cipher_Block_Chaining_.28CBC.29

	Mode of operation: Counter Mode (CM) Employs Dynamic Encryption
Live Chat	Used by **SF.MESSAGING** Deploys Dynamic Encryption Mode of operation: Cyclic Block Chain (CBC) PKCS7 padding Text is base 64 encoded
ZRTP (RFC 6189)	Used by **SF.MESSAGING** Modified by Dencrypt DH-RSA 3072 bits AES-256 (hard coded algorithm)
SIPS	Used by **SF.CHANNEL** for the connection to the SIP server Employs TLS SIPS is SIP [RFC3261] that runs over Transport Layer Security (TLS) [RFC5246]. Note that the security relies on TLS connection and not on SIP or MD5.
TLS	Used by **SF.CHANNEL** TLS 1.2 Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 Elliptic curve: secp384r1 • Used for the DPS webAPI connection • Used for the DCS webAPI connection • Used for the DCM webAPI connection TLS 1.2 Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 Elliptic curve: secp384r1 • Used for the SIP Server DCS connection
X509 Certificates	Used by **SF.CHANNEL** for the TLS authentication • RSA 3072bits • SHA512
RNG	Random number generation is using the Yarrow algorithm on iOS (TOE environment).

The TOE relies on two open source projects for encryption technology:

- **mbedTLS**. This is an Open Source cryptography library that provides a wide variety of algorithms. The TOE uses mbedTLS's AES-256, SHA1, SHA2, MD5 and RSA, as well as its TLS implementation. Dencrypt has extended mbedTLS's functionality by Dynamic Encryption. See also https://tls.mbed.org/
- **libSRTP**. The libSRTP library is an open-source implementation of the Secure Real-time Transport Protocol (SRTP) originally authored by Cisco Systems, Inc. It implements AES-256 and SHA1. Dencrypt has extended libSRTP's functionality by Dynamic Encryption. See also http://srtp.sourceforge.net/srtp.html

8 Abbreviations and references

8.1 Abbreviations

AES	Advanced Encryption Standard
AES-CM	AES – counter mode
CBC	Cipher Block Chaining
CC	Common Criteria
CM	Counter Mode
CRL	Certificate Revocation List
CSR	Certificate Signing Request
CUG	Closed User Group
DCM	Dencrypt Certificate Manager
DCC	Dencrypt Control Center
DCS	Dencrypt Communication Server
DDB	Dencrypt DataBase
DES	Data Encryption Standard
DH	Diffie-Hellman key exchange
DPS	Dencrypt Provisioning Server
EAL	Evaluation Assurance Level
ECC	Elliptic Curve Cryptography
GCM	Galois/Counter Mode
HMAC	Keyed-Hash Message Authentication Code
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
MDM	Mobile Device Management
OSP	Organisational Security Policy
PKI	Public Key Infrastructure
PP	Protection Profile
RSA	Acronym for Rivest, Shamir, Adleman, the creators of the RSA algorithm
RTP	Real-Time Transport Protocol
SAR	Security Assurance Requirement
SAS	Short Authentication String
SDP	Session Description Protocol
SFP	Security Function Policy
SFR	Security Functional Requirement
SIM	Subscriber Identification Module
SIP	Session Initiation Protocol

SMS	Short Message Service
SMS-C	Short Message Service Center
SRTP	Secure Real-time Transport Protocol
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TSF Interface
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.

[FIPS180-4] Federal Information Processing Standards Publication 180-4, Secure Hash Standard (SHS), 2012 March.

[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

[RFC2104] HMAC: Keyed-Hashing for Message Authentication, February 1997.

[RFC3261] SIP: Session Initiation Protocol, June 2002

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

[RFC3830] MIKEY: Multimedia Internet KEYing, Ericsson Research, August 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.

[RFC6125] Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS), March 2011.

[RFC6189] ZRTP: Media Path Key Agreement for Unicast Secure RTP, April 2011.

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