CCN-PP-TP_EAL2

Protection Profile for Trusted Platform for secure communications. EAL2+

Version 1.0

29-09-2016

Centro Criptológico Nacional
C/ Argentona, 20
28023 Madrid, España
internet: https://www.ccn.cni.es

Foreword

This document presents the Common Criteria (CC) Protection Profile (PP) issued by the "**Centro Criptológico Nacional (CCN)**", Spain, to express the fundamental security properties for Mobile Devices providing a trusted platform for secure communications claiming conformance with this protection profile.

The Protection Profile has been prepared following the rules of Common Criteria Version 3.1 Revision 4. The assurance level is EAL2 augmented with ALC_FLR.2.

Revision History

Version	Date	Update Description
1.0	29 September 2016	Initial Release

Document organisation

This Protection Profile (PP) is divided into six sections.

Section 1 provides the introductory material for this PP. It includes a high level description of the security functionally expected to be implemented in TOEs claiming conformance with this PP.

Section 2 describes PP conformance claims.

Section 3 describes the TOE Security Problem Definition. This includes threats and policies that are to be addressed by the TOE.

Section 4 defines the security objectives for the TOE and for the operational environment. The Security Functional Requirements (SFRs) in Section 6 are a more formal description of the security features that achieve the security objectives for the TOE.

Section 5 defines functional requirements that can not be expressed using [CC31p2] catalogue. They are specified as extended functional requirements.

Section 6 contains the functional and assurance requirements that must be satisfied by the TOE and are derived from the CC, Parts 2 and 3 respectively and extended requirements defined in Section 5 of this PP. The functional requirements are a more formal specification of the Security Objectives for the TOE and the assurance requirements are a description of the activities undertaken by the developer and evaluator to achieve and verify a sound implementation of the functional requirements.

Contents

1 Introduction to the PP

1.1 PP Identification

Title	Protection Profile for Trusted Platform for secure communications. EAL2+
Version	1.0
Date	29-09-2016
Sponsor	Centro Criptológico Nacional (CCN)
CC Version	Common Criteria for Information Technology Security Evaluation Version 3.1 R4, September 2012.
CC Evaluation Level	Evaluation Assurance Level EAL2 + ALC_FLR.2

1.2 PP Overview of the TOE

1.2.1 TOE definition

The TOE type is a mobile device that provides trusted mechanisms for secure communications with external entities (other devices). It can be used, for example for voice and data communications applications using a trusted channel. The trusted channel is a VPN providing confidentially, integrity and end-points authenticity.

The TOE covered by this PP is part of a mobile infrastructure for secure communications that consists of a handset, trusted external entities and a key generation system. The TOE is limited to the mobile device (the handset).

The TOE connects to the internet using either a mobile network or wi-fi networks, but in either case, the communication with trusted external entities is through trusted channels so that the IP traffic is sent and/or received using the trusted channel. Trusted channels are used for application communications, for remote administration of the TOE or for sending audit records to an external entity.

The TOE allows applications installed in the mobile device to communicate securely with the protected networks over a trusted channel called VPN tunnel. These protected networks are behind a VPN endpoint. Most of application data flowing from the handset to the VPN endpoint is done through the VPN tunnel. Bypass capability is implemented for specific and allowed applications.

Depending on the applications running in the TOE and using the VPN tunnel, at its end-point, additional services can be installed (e.g. app-market, update server, NTP- and voice- and messaging-servers).

The mobile device could also serve as an external “crypto-modem” by connecting the mobile device to a computer using USB. In case of the TOE implements that feature, all ip-

traffic to and from the computer using this interface will be routed inside and protected by the VPN-tunnel.

In addition, the TOE supports white listing for software applications, user data storage encryption and integrity control.

1.2.2 TOE major security features

The main security features to be implemented in a TOE claiming conformance with this PP are:

- **Trusted channels**. Communication with external entities is mostly performed through trusted channels providing confidentiality, integrity of the data flowing through the channel and endpoints authenticity. Depending on the usage and the traffic flowing, the following trusted channels are considered:
    - VPN-tunnel. All IP-based traffic is encrypted by default within a VPN-tunnel terminating in the TOE VPN. The VPN-tunnel is used for applications communications. The tunnel is completely transparent to all applications using the tunnel on the device.
    - CIK-tunnel. The "Crypto Ignition Key" (CIK) is to be entered into the TOE from an external device through a dedicated trusted channel.
    - In addition, trusted channels are also used for remote administration or for sending the audit log to an external entity.
- **Application whitelisting**. Only applications included in the white list by the organisation can be installed on the mobile device.
- **Bypass capability**: VPN-policy for applications. Only applications specified by the organisation could communicate outside of the tunnel.
- **Disk encryption**. The TOE implements disk encryption with strong external keys. The system implements a key hierarchy consisting of the "Key Encryption Key" (KEK), used for the encryption of the "Disk encryption key" (DEK). The DEK is used for the encryption/decryption of the user data partition of the device.
- **Secure boot and operation continuity.** Boot of the TOE only success if the integrity of the OS is guaranteed and the proper KEK for the disk decryption is provided by the user or obtained by derivation from the user credentials input. For the operation continuity, periodically, the TOE verifies at configurable time intervals, the KEK stored in an external device (e.g. bluetooth device) through a trusted channel. In this case the KEK stored in the device is called "Crypto Ignition Key" (CIK). The user has to carry this external device jointly with the TOE, so that if both are separated and communication is lost, the TOE will be blocked. Entering in a blocking situation requires the KEK to be provided or derived from user's credentials and a similar secure boot process is performed.
- **Zeroization.** Remote and local secure wipe. If an emergency erase is invoked, the Critical Security Parameters and classified and personal (address book, calendar, etc.) information on the device will be actively overwritten.
- **Fail-safe functionality.**

- **Integrity assurance**. The integrity of the OS shall be checked during power-up. The integrity of the cryptographic mechanisms shall be checked during power-up and during their execution. The VPN-tunnel modules and VPN-tunnel configuration integrity shall be checked prior to the establishment of a VPN-tunnel connection. The integrity of the other CSPs shall be monitored prior to their usage. An integrity failure causes an emergency zeroization.
- **Selftesting**. The TOE runs a suite of self tests during initial start-up and periodically during normal operation to demonstrate the correct operation of the security functionality.

- **Trusted updates**. All updates must be signed by the organisation and the signature will be verified by the TOE before their installation.
- **Strong authentication for access control.** Strong authentication mechanisms shall be implemented together with the access control policy for the TOE administration. The TOE configuration (the crypto software and the tunnels and applications policies) shall be reserved for the corresponding authority (administrator) under the security management policy. For the normal operation, the user of the TOE accessing its functionality and user data stored must be authenticated. Authentication may be based on at least a PIN with minimum 4 digits, a biometry mechanism or any other mechanism with an equivalent strength. Failed authentications attempts shall be controlled and actions (secure wipe) shall be executed when meeting the threshold of failed attempts. Sessions shall be controlled locking them after a configurable period of inactivity, but no longer than 30 minutes.
- **Tamper evident**. The enclosure of the TOE is protected with tamper-evident seals.
- **Audit.** The TOE records security relevant events and associate each event with the identity of the user that caused the event. The audit trail is protected for unauthorized modification and loss of audit trail data. The TOE may be commanded to send the audit log to an external entity through a trusted channel.

1.2.3 TOE operational usage

The TOE is intended to be operated as an enterprise owned device for general purpose enterprise use and limited personal use. An enterprise owned device for general purpose business use entails a full enterprise control over configuration and software inventory.

The enterprise elects to provide users with mobile devices and additional applications in order to maintain control of their enterprise data and security of their networks. Users may use Internet connectivity to browse the web or run allowed applications, but this connectivity will be under control of the enterprise.

It will be able to communicate via its Wi-Fi or mobile data network, but only using the VPN tunnel with the VPN endpoint. Only the specific allowed application may bypass the VPN-tunnel.

1.2.4 Required non-TOE hardware/software/firmware

The following figure shows the TOE scope.

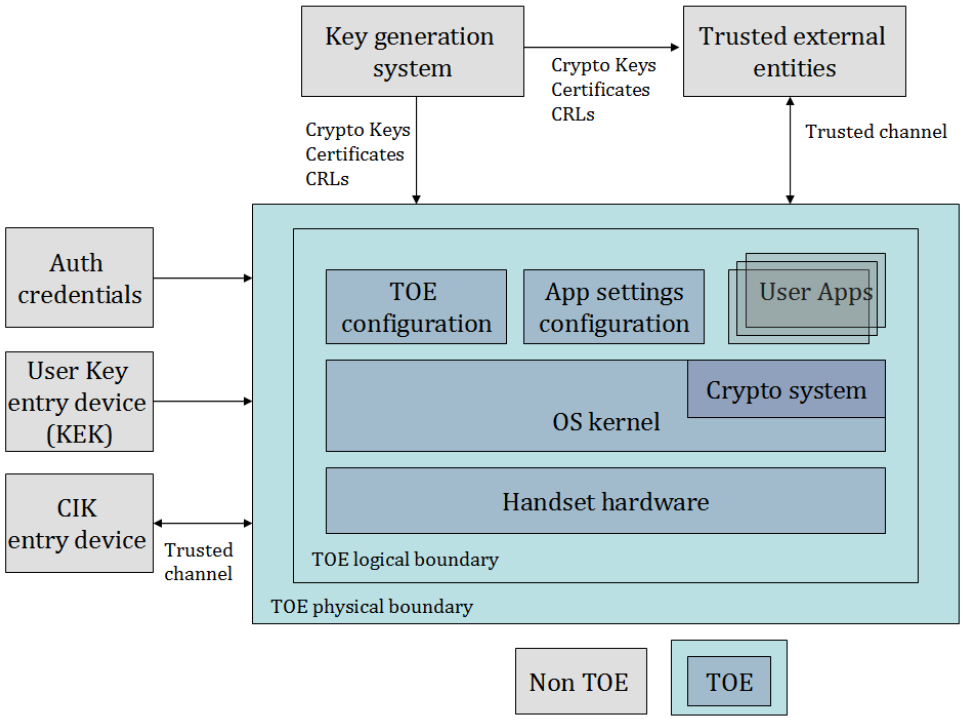


Figure 1: TOE boundary

The following components are part of the operational environment of the TOE:

- Trusted external entities, such as the VPN endpoint, or external audit logs storage;
- Key generation system providing the mobile device and trusted external entities (e.g. the VPN endpoint) with keys, certificates and CRLs;
- All user apps installed in the handset (within the physical boundary of the TOE) or available to be installed from an authorized repository, such as SIP clients, calendar apps, etc;
- External devices used for user authentication (e.g. NFC-tags or QRs for KEK);
- External devices used for the CIK provision;
- Any other hardware or software that is not necessary for the secure operation of the TOE, such as NTP server, log servers and analysis tools used for collecting and analyzing the audit records, app-market server, update server, servers for Voice over IP and instant messaging, etc.
- Laptops or any other devices that could be connected to the handset to give access to services available behind the VPN endpoint.
- Other networks devices belonging to the organisation to enforce policies for accessing other networks (firewalls).

1.3 References

[CC31p1] Common Criteria for Information Technology Security Evaluation.
Part 1: Introduction and general model
Version 3.1, Revision 4

[CC31p2] Common Criteria for Information Technology Security Evaluation.
Part 2: Security Functional Components
Version 3.1, Revision 4

[CC31p3] Common Criteria for Information Technology Security Evaluation.
Part 3: Security Assurance Components
Version 3.1, Revision 4

[CEM31] Common Criteria for Information Technology Security Evaluation.
Evaluation Methodology
Version 3.1 Revision 4

[AIS31] A proposal for: Functionality classes for random number generators. Version 2.0, 18 September 2011

[AIS20] Functionality Classes and Evaluation Methodology for Deterministic Random Number Generators. Version 2.0, 2 December 1999

1.4 Acronyms

Apps	Applications
CC	Common Criteria
CIK	Crypto Ignition Key
CSP	Critical Security Parameter
DEK	Disk Encryption Key
EAL	Evaluation Assurance Level
FW	FirmWare
HW	HardWare
IC	Integrated Circuit
IP	Internet Protocol
KDF	Key Derivation Function
KEK	Key Encryption Key
NFC	Near Field Communication
OS	Operating System
PP	Protection Profile
QR-code	Quick Response code
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SPD	Security Problem Definition

SW	SoftWare
TOE	Target Of Evaluation
TSFS	TOE Security Functionality
TSS	TOE Summary Specification
USB	Universal Serial Bus
VPN	Virtual Private Network

1.5 Definitions

Critical Security Parameter (CSP): security-related information (e.g., secret and private cryptographic keys, and authentication data such as passwords, passphrases and PINs) whose disclosure or modification can compromise the security of a cryptographic module. This includes also the VPN configuration.

2 Conformance Claims

2.1 CC Conformance Claims

This Protection Profile claims conformance to

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model; CCMB-2012-09-001, Version 3.1, Revision 4, September 2012

- Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components; CCMB-2012-09-002, Version 3.1, Revision 4, September 2012

- Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components; CCMB-2012-09-003, Version 3.1, Revision 4, September 2012

as follows

- Part 2 extended,

- Part 3 conformant.

The "Common Methodology for Information Technology Security Evaluation, Evaluation methodology" CCMB-2012-09-004, Version 3.1, Revision 4, September 2012" has to be taken into account.

2.2 PP Claim

This Protection Profile does not claim any conformance to other Protection Profiles.

2.3 Package Claim

This Protection Profile is conformant to the assurance package EAL2 augmented by the assurance component ALC_FLR.2.

2.4 Conformance Rationale

This Protection Profile does not claim any conformance with other PPs. Therefore, no conformance claim rationale needs to be given here.

2.5 Conformance Statement

This Protection Profile requires ***strict*** conformance of any ST or PP claiming conformance to this PP.

3 Security Problem Definition

This section defines the security problem to be addressed by the TOE and its operational environment.

The mobile device being this TOE can be described as an enterprise owned device for general purpose enterprise use and limited personal use. For such a TOE there is a significant degree of 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 high degree of control of their enterprise data and security of their networks. Organisations need to decide and make the configurations oriented to allow or not allow end users to connect to third party services (i.e. internet).

It is assumed that the TOE is under physical control of the user or the organisation 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

The threats are defined by an adverse action performed by a threat agent on an asset.

3.1.1 Assets

The assets to be protected by the TOE are the following.

AS.USER_DATA
Data belonging to the TOE user such as files, agenda, contacts, SMS, call registers, etc. Dimensions to be protected: Confidentiality and Integrity.

AS.VPN_MODULES
Software components and modules used to establish VPN connections. Dimensions to be protected: Integrity.

AS.CRYPTOGRAPHIC_ASSETS
Cryptographic material stored and used within the TOE including the cryptographic mechanisms itself. Dimensions to be protected: Confidentiality and Integrity.

AS.SECURITY_CONFIGURATION DATA
Configuration data used by the TOE to establish the security properties of the TOE, such as policies or specific configurations for security services. Dimensions to be protected: Integrity.

AS.VPN_CONFIGURATION DATA

Configuration data used by the TOE to establish the VPN connection, such as the VPN termination agent IP address or URL. Dimensions to be protected: Integrity.

AS.INSTALLED_APPLICATIONS
Applications which are authorized to be installed in the TOE. Dimensions to be protected: Integrity and Authorization.

AS.CONFIDENTIAL_COMMUNICATIONS
Communications which are expected to be encrypted. Within this asset it is also included the integrity of the mechanisms that enable the TOE to differentiate between those applications whose communications must be encrypted thorough the VPN and those which are not confidential. Dimensions to be protected: Confidentiality and Integrity.

AS.OS
Operative system and core component software which runs below the installed applications. Dimensions to be protected: Integrity.

AS.HANDSET
The HW integrity is to be protected in such a way that any tamper attempt must leave evidence.

3.1.2 Agents

The threat agents in this SPD may be anyone who has interest in compromising the TOE and possesses a high expertise, resources, opportunity and motivation, commensurate with the **enhanced basic attack potential**.

AG.EXTERNAL: any agent who is not authorized to handle or operate the TOE.

AG.USER: any agent who is authorized to handle and operate the TOE and therefore constitutes a legitimate TOE user if he/she follows the established security policy.

3.1.3 Threats

The threats specified below are addressed by the TOE and the TOE environment.

T.UNAUTH_INST
A legitimate user or an attacker manages to install applications in the TOE which are not authorized by the consumer organization.
Assets: AS.INSTALLED_APPLICATIONS
Threat agent: AG.USER, AG.EXTERNAL

T.CRYPT_COMPROMISE
A legitimate user or an attacker retrieves or modifies cryptographic assets such as all the keys and certificates stored and managed by the TOE. This includes also the possible modification of the cryptographic mechanisms. This threat covers the use case for

legitimate users, but only when the legitimate user is not authorized to retrieve these assets.

Assets: AS.CRYPTOGRAPHIC_ASSETS.
Threat agent: AG.USER, AG.EXTERNAL

T.USR_DATA
An attacker retrieves, access or modifies user data stored or to be transmitted, protected by the TOE. This threat applies to all the external interfaces of the TOE (3G, Wi-Fi, USB, NFC, Bluetooth, etc.). VPN interface is addressed in other threats.

Assets: AS.USER_DATA
Threat agent: AG.EXTERNAL

T.VPN_CONFIG
A legitimate user or an attacker is able to modify the VPN configuration data and/or the software components and modules which handle the VPN connection. This threat covers the use case for legitimate user in these cases:

- when the legitimate user is not authorized to modify the VPN configuration;
- whenever the user modifies the software components.

Assets: AS.VPN MODULES, AS.VPN CONFIGURATION_DATA
Threat agent: AG.USER, AG.EXTERNAL

T.CONF_DATA
A legitimate user or an attacker is able to modify the security configuration data which is managed by the TOE. This threat covers the use case for a legitimate user, but only when the legitimate user is not authorized to modify this data.

Assets: AS.SECURITY CONFIGURATION_DATA
Threat agent: AG.USER, AG.EXTERNAL

T.UNAUTH_BOOT
An attacker manages to bypass the initial encryption mechanism used to encrypt the TOE and is able to boot and start up the TOE.

Assets: AS.USER_DATA, AS.VPN_MODULES, AS.CRYPTOGRAPHIC_ASSETS, AS.VPN_CONFIGURATION_DATA, AS.SECURITY_CONFIGURATION_DATA, AS.CONFIDENTIAL_COMMUNICATIONS.
Threat agent: AG.EXTERNAL

T.BYPASS
An attacker manages to access to TOE services, functions, installed applications or user data bypassing the TOE authentication mechanisms which unlocks these TOE features.

Assets: AS.APPLICATIONES, AS.USER_DATA.
Threat agent: AG.EXTERNAL

T.UNAUTH_VPN
An attacker or a legitimate user manages to redirect or extract confidential communications outside the VPN tunnel, bypassing the security mechanisms established to force the TOE applications to communicate through this channel.

Assets: AS.CONFIDENTIAL_COMMUNICATIONS, AS.USER_DATA.
Threat agent: AG.EXTERNAL, AG.USER

T.ATTACK_VPN
An attacker is able to disclose information or undetected modify information that is communicated between the TOE and endpoint of the VPN tunnel.
Assets: AS.CONFIDENTIAL_COMMUNICATIONS, AS.USER_DATA.
Threat agent: AG.EXTERNAL, AG.USER

T.UNAUTH_COM
An attacker manages to establish an unauthorized communication channel, extract information or access TOE assets using some of the TOE available interfaces.
Assets: AS.USER_DATA, AS.CRYPTOGRAPHIC_ASSETS, AS.VPN_CONFIGURATION DATA, AS.SECURITY_CONFIGURATION_DATA
Threat agent: AG.EXTERNAL

T.UNAUTH_ADMIN
An unauthorized user or attacker manages to access administrative, configuration or development functionalities established within the TOE.
Assets: AS.SECURITY_CONFIGURATION_DATA, AS.VPN_CONFIGURATION_DATA, AS.OS
Threat agent: AG.USER, AG.EXTERNAL

T.OS_MOD
An unauthorized user or attacker manages to modify operating system or core component software of the TOE.
Assets: AS.OS
Threat agent: AG.USER, AG.EXTERNAL

T.HW_TAMPER
An attacker manages to open the handset through the standard opening mechanisms (screws, covers) without leaving any evidence of the attack.
Assets: AS.HANDSET
Threat agent: AG.EXTERNAL

3.2 Organisational Security Policies

The organisational security policies are specified for the control of the management functions and demands on the accountability of users' actions:

P.SECURE_MGMNT
The TOE consuming organization shall be responsible for establishing a security policy which will define the processes to manage the TOE security. At least this policy should include that only authorized personnel (authority) may have access to security management functionalities and, whenever not necessary, this functionality shall be disabled.

P.CRYPTO_MGMNT
The TOE consuming organization shall be responsible for establishing a specific policy to manage the TOE cryptographic assets and their delivery.

P.SECURE_USE
The TOE consuming organization shall be responsible for establishing a specific policy which will define the TOE specific use policy applicable to users, establishing at least the different data which the TOE can manage and how a user shall handle that data.

P.VPN_BYPASS
The TOE shall implement a VPN-tunnel bypass capability managed by the corresponding VPN-policy for applications.

P.AUDIT
The TOE must record security relevant events and associate each event with the identity of the user that caused the event. The audit trail shall be protected for unauthorized modification and loss of audit trail data. The TOE shall provide authorized administrators with the ability to review the audit trail. The TOE shall provide management functionality to enable the capacity of sending the audit trail to an external entity.

P.RNG
The TOE must implement random number generators meeting the requirements of strength and quality metrics specified in [AIS20] and [AIS31].

3.3 Assumptions

This section specifies the assumptions that must be satisfied by the TOE environment.

A.NOEVIL
It assumed that those users belonging to the authority, who are authorized to securely manage the TOE and its operational environment, are trustworthy and they have been trained sufficiently to carry out these security management tasks in a proficient manner.

A.SINGLEUSER
It is assumed that the TOE is used and under the control of a single user only.

A.KEYS
It is assumed that the crypto-material (e.g. keys used for the encryption of TOE data storage or the key provided to the user) entered into the TOE are of good quality, not disclosed and only distributed to the appropriate handsets and users.

A.APPS
It is assumed that all applications that are white-listed does not reveal sensitive user data on the screen lock without user authentication.

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 IT security objectives are to be met by the TOE.

O.TUNNEL
The TOE shall provide trusted channels that will control the data traffic with external entities, authenticate the end point and ensure that data exchanged over the channel is protected against disclosure. The TOE must also transmit the data such as the remote entity can verify the integrity of data received. The TOE shall implement the following trusted channels:

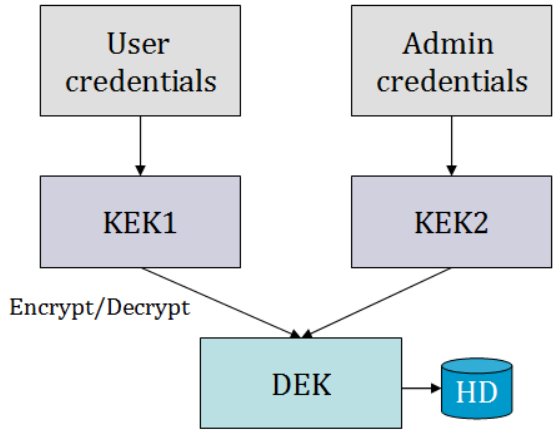
- **VPN-tunnel**. The TOE shall provide a VPN-tunnel terminating in the TOE VPN for the applications communications. All IP-based traffic flows by default through the VPN-tunnel. The applications may communicate securely with protected networks behind a VPN endpoint.
- **CIK-tunnel**. The TOE shall provide a trusted channel with an external device to enter the "Crypto Ignition Key" (CIK).
- The TOE shall provide **trusted channels** for the remote administration or when sending the audit log to an external entity.

O.INSTALLATION
The TOE shall be able to install only authorized applications. Only applications signed by the organisation can be installed on the mobile device.

O.SECURE_BOOT
This objective addresses the **secure boot** and the **operation continuity** processes.

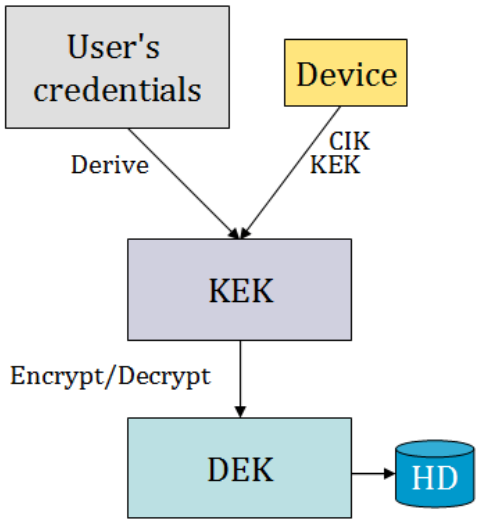
The TOE shall be able to authenticate the TOE user by means of the user credentials or directly the encryption key prior to decrypting and booting up the TOE. The security mechanism used to authenticate TOE user shall be resistant to brute force attacks. User's credentials may be passphrases, passwords, PIN numbers, etc.

The system implements a key hierarchy consisting of the key entered by the user, the "Key Encryption Key" (KEK), used for the encryption of the "Disk encryption key" (DEK). The DEK is used for the encryption/decryption of the user data partition of the device.



During boot, the TOE will need the KEK for the disk decryption. This key may be entered by typing it using the device keyboard, using an external device (e.g. QR-codes or NFC-tags) or may be derived from the user's credentials. Boot of the TOE only successes if the integrity of the OS is guaranteed and the proper external KEK for the disk decryption is provided by the user.

Periodically, the TOE shall verify at configurable time intervals, the KEK stored in an external device through a trusted channel. In this case the KEK stored in the device is called "Crypto Ignition Key" (CIK). Fail to the verification shall cause the TOE to be blocked. Entering in a blocking situation requires the KEK to be provided or derived from user's credentials and a similar secure boot process shall be performed.



O.ERASURE
The TOE shall be able to perform local and remote securely erase of Critical Security Parameters and classified and personal (address book, calendar, etc) upon request of an authorized user or in emergency situations. This process shall be implemented starting

with the zeroization of the KEK and then with the rest of the data. Emergency situations are the following:

- defined number of initial decryption unsuccessful attempts prior to the TOE boot up.
- the detection of an integrity violation

O.INTEGRITY
The TOE shall be able to verify the integrity of:

- the OS during power-up;
- cryptographic mechanisms during power-up and during their execution;
- VPN modules and VPN configuration prior to the establishment of a VPN connection;
- the other CSPs prior to their usage.

An integrity failure shall cause an immediate emergency zeroization.

O.SELFTEST
The TOE shall be able to run a suite of self tests during initial start-up and periodically during normal operation to demonstrate the correct operation of the security functionality. Self tests of cryptographic functions are also to be performed. If the cryptographic self test fails, no VPN tunnel shall be established. The user will be warned about the impossibility of the tunnel setting up. If the TOE is unable to set-up a tunnel all IP-traffic selected for its transmission through the tunnel shall be blocked.

O.OS_UPDATE
The TOE shall be able to verify that TOE updates are authorized prior to its installation.

O.AUTHENTICATION
The TOE shall be able to authenticate the authorized users and therefore this mechanism will be used to control the access to the user data once the TOE has been decrypted and booted up. The security mechanism used to authenticate TOE users shall be resistant to brute force attacks.

O.ADMIN
The TOE shall be able to restrict security configuration privilege escalation to authorized users.

O.SECURITY_POLICIES
The TOE shall be able to add and execute security policies and rules that prevent unauthorized access to the security features that the TOE manages. The VPN-tunnel bypass capability is managed by the corresponding VPN-policy for applications. Special applications that need to communicate outside of the tunnel can be signed using a specific certificate defined in the Mobile Device policy that will allow them to connect outside of the tunnel.

O.CRYPT_PROTECTION

The TOE shall be able to protect cryptographic assets from unauthorized access, retrieval or modification.

O.SECURITY_DATA
The TOE shall be able to protect the entire security configuration from unauthorized access or modification.

O.AUDIT
The TOE must record security relevant events and associate each event with the identity of the user that caused the event. The TOE must prevent unauthorized modification of the audit trail, prevent loss of audit trail data, and provide authorized administrators with the ability to review the audit trail. The TOE shall be able to send the audit trail to an external entity when a security management function requires it.

O.HW_TAMPER
The enclosure of the TOE shall be protected with tamper-evident seals.

O.RNG
The TOE must implement random number generators meeting the requirements of strength and quality metrics specified in [AIS20] and [AIS31].

4.2 Security Objectives for the operational environment

The following are the security objectives for the operational environment of the TOE that are necessary for the TOE to meet its security objectives.

OE.SECURE_MGMNT
The consuming organization shall be responsible for establishing a security policy which will define the processes to manage the TOE security and its operational environment. At least this policy should include that only authorized personnel may have access to security management functionalities and, whenever not necessary, this functionality shall be disabled.

OE.CRYPTO_MGMNT
The TOE consuming organization shall be responsible for establishing a specific policy to manage the TOE cryptographic assets and their delivery.

OE.SECURE_USE
The TOE consuming organization shall be responsible for establishing a specific policy which will define the TOE and its operational environment specific use policy applicable to users, establishing at least the different data which the TOE can manage and how a user shall handle that data.

OE.NOEVIL

Those users who are authorized to securely manage the TOE shall be trustworthy, and they shall be trained sufficiently to carry out these security management tasks in a proficient manner.

OE.SINGLEUSER
The TOE is used and under the control of a single user only.

OE.KEYS
Crypto-material (e.g. keys used for the encryption of TOE data storage or the key provided to the user) entered into the TOE are of good quality, not disclosed and only distributed to the appropriate handsets and users.

OE.APPS
Applications that are whitelisted are trustworthy.

4.3 Security Objectives rationale

4.3.1 Security Objectives Coverage

The following tables provide a mapping of security objectives for the TOE and the TOE environment to the defined 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.UNAUTH_INST	T. CRYPT_COMPROMISE	T.USR_DATA	T.VPN_CONFIG	T.CONF_DATA	T.UNAUTH_BOOT	T.BYPASS	T.UNAUTH_VPN	T.ATTACK_VPN	T.UNAUTH_COM	T.UNAUTH_ADMIN	T.OS_MOD	T.HW_TAMPER
O.TUNNEL					X		X	X	X	X	X		
O.INSTALLATION	X												
O.SECURE_BOOT						X	X			X	X		
O.ERASURE		X	X										
O.INTEGRITY		X		X				X	X	X		X	
O.SELFTEST				X				X	X	X		X	
O.OS_UPDATE												X	
O.AUTHENTICATION			X	X	X		X			X	X		
O.ADMIN				X	X						X		
O.SECURITY_POLICIES				X	X			X					
O.CRYPT_PROTECTION		X		X									
O.SECURITY_DATA				X	X								
O.HW_TAMPER													X
OE.KEYS									X				

OE.APPS										X			

Table 1: Threats to security objectives (TOE & ENV)

	P.SECURE_MGMNT	**P.CRYPTO_MGMNT**	**P.SECURE_USE**	**P.VPN_BYPASS**	**P.AUDIT**	**P.RNG**	**A.NOEVIL**	**A.SINGLEUSER**	**A.KEYS**	**A.APPS**
O.TUNNEL					X					
O.SECURITY_POLICIES				X						
O.AUDIT					X					
O.RNG						X				
OE.SECURE_MGMNT	X									
OE.CRYPTO_MGMNT		X								
OE.SECURE_USE			X							
OE.NOEVIL							X			
OE.SINGLEUSER								X		
OE.KEYS									X	
OE.APPS										X

Table 2: OSPs and Assumptions to Security objectives (TOE & ENV)

4.3.2 Security Objectives Sufficiency

4.3.2.1 Rationale for the threats

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.

T.UNAUTH_INST
This threat is addressed by requiring the TOE to install only authorized applications (O.INSTALLATION).

T. CRYPT_COMPROMISE
This threat is addressed by requiring the TOE to protect cryptographic assets from unauthorized access, retrieval or modification (O.CRYPT_PROTECTION). Unauthorised modification of cryptographic mechanisms will be detected thanks to the integrity control performed according to O.INTEGRITY which requires the TOE checking it during power-up and during the cryptographic mechanisms execution. An integrity failure shall cause an emergency zeroization. The TOE provides securely erase upon request or in case of errors (O.ERASURE).

T.USR_DATA

This threat is addressed by requiring authentication before access is given to user data (O.AUTHENTICATION) and by the TOE to provide securely erase upon request or in case of errors (O.ERASURE).

T.VPN_CONFIG
This threat is addressed as follows:

- To avoid the unauthorised modification of the VPN configuration data the TOE requires authentication of the users before giving access to the TSF (O.AUTHENTICATION) restricting the configuration changes to authenticated administrators (O.ADMIN) and allowing only authorized administrator to change the security policies (O.SECURITY_POLICIES). The TOE is to protect the entire security configuration from unauthorized access or modification (O.SECURITY_DATA).
- To avoid the modification of the VPN components, the TOE verifies the VPN modules and VPN configuration integrity before establishing a VPN connection (O.INTEGRITY). Also the self tests performed on cryptographic functions (O.SELFTEST) and the protection of the cryptographic assets (O.CRYPT_PROTECTION), contribute mitigating the threat.

T.CONF_DATA
This threat is addressed by requiring the TOE to authenticate the users before giving access to the TSF (O.AUTHENTICATION), by restricting the configuration changes to authenticated administrators (O.ADMIN) and by allowing authorized administrator to change the security policies (O.SECURITY_POLICIES). The TOE is to protect the entire security configuration from unauthorized access or modification (O.SECURITY_DATA). (O.TUNNEL) contributes also in mitigating the threat in that a trusted channel is required for remote administration.

T.UNAUTH_BOOT
This threat is addressed by requiring the TOE to authenticate the users before giving access to the TSF by preventing the TOE to boot unless the right decryption key (KEK) is given (O.SECURE_BOOT).

T.BYPASS
This threat is addressed by requiring the TOE to authenticate the users before giving access to the TSF (O.AUTHENTICATION) and by preventing the TOE to boot unless the right decryption key (KEK) is given or to continue with the normal operation unless the CIK is entered when required from an external device configured for that purpose (O.SECURE_BOOT). (O.TUNNEL) contributes also in mitigating the threat in that a trusted channel is required to enter the CIK.

T.UNAUTH_VPN
This threat is addressed by requiring the TOE to provide a VPN tunnel that will control the data traffic of applications (O.TUNNEL). The tunnel has been properly setup (correct behaviour and integrity of its components is assured by O.SELFTEST and O.INTEGRITY). Se security policies for the apps have been established and only authorized administrators can change them (O.SECURITY_POLICIES).

T.ATTACK_VPN
This threat is addressed by requiring the TOE to provide a VPN tunnel that will control the data traffic of applications, authenticate the end point and protect the data transmitted (O.TUNNEL). The tunnel has been properly setup (correct behaviour and integrity of its components is assured by O.SELFTEST and O.INTEGRITY). It is supported by the assumption that the cryptographic parameters and keys are of good quality and secure as in (OE.KEYS).

T.UNAUTH_COM
This threat is addressed by requiring the TOE to authenticate users before booting, before continue operating (at regular established periods), or accessing to the TSF (O.SECURE_BOOT and O.AUTHENTICATION).

The TOE will control the data traffic of applications, the traffic of the remote administration, the audit log traffic and the CIK transmission by means of the corresponding trusted channels (O.TUNNEL).

For applications, the VPN-tunnel has been properly setup (correct behaviour and integrity of its components is assured by O.SELFTEST and O.INTEGRITY). This is also supported by the TOE environment that will only allow certain whitelisted applications that must not reveal sensitive on the screen lock without user authentication (OE.APPS).

T.UNAUTH_ADMIN
This threat is addressed by requiring the TOE to prevent security configuration by other than authorized users (O.AUTHENTICATION and O.ADMIN). (O.TUNNEL) contributes also in mitigating the threat in that a trusted channel is required for remote administration.

T.OS_MOD
This threat is addressed by requiring the TOE to verify system updates are authorized prior to its installation (O.OS_UPDATE). To avoid modifications on the operating system or core software components, the TOE verifies the OS integrity during power up (O.INTEGRITY). Also the self tests performed (O.SELFTEST) contributes mitigating the threat.

T.HW_TAMPER
This threat is addressed directly by (O.HW_TAMPER) requiring that the TOE enclosure is protected with tamper-evident seals.

4.3.2.2 Rationale for OSPs

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

P.SECURE_MGMNT
OE.SECURE_MGMNT directly enforces this OSP.

P.CRYPTO_MGMNT
OE.CRYPTO_MGMNT directly enforces this OSP.

P.SECURE_USE
OE.SECURE_USE directly enforces this OSP.

P.VPN_BYPASS
This OSP is enforced by O.SECURITY_POLICIES which requires the implementation of a VPN-tunnel bypass capability managed by a VPN-policy for applications. The applications authorised to communicate outside of the tunnel are to be signed using a specific certificate defined in the policy that will allow them to connect outside of the tunnel.

P.AUDIT
O.AUDIT directly enforces this OSP. (O.TUNNEL) contributes also in mitigating the threat in that a trusted channel is required for sending the audit logs to an external entity.

P.RNG
O.RNG directly enforces this OSP.

4.3.2.3 Rationale for Assumptions

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

A.NOEVIL
OE.NOEVIL directly upholds this assumption.

A.SINGLEUSER
OE.SINGLEUSER directly upholds this assumption.

A.KEYS
OE.KEYS directly upholds this assumption.

A.APPS
OE.APPS directly upholds this assumption.

5 Extended components definition

This extended components definition is included as per [CC31p1]. Extended components may be based on existing CC requirements. In this PP, the extended components are based on existing classes and families from the CC.

New families and components are created to capture functionality required for this TOE. The extended components are defined in the following sections and are then instantiated as requirements in Section 6 of this PP. Application notes are included where appropriate to assist the ST author, vendor, or evaluator in understanding the intent of the requirement, identify implementation choices, or define pass-fail criteria for a requirement.

Consistent with the CC, these application notes presented in this section are normative text.

The extended requirement FDP_DSK.1 for stored data encryption is used to specify the transparent encryption performed on a mobile device.

The extended requirement FDP_ZER.1 for zeroization is used to specify the ability of local and remote secure erase.

The extended requirement FPT_SBT.1 is defined to specify the secure boot and the operation continuity processes.

The extended requirement FPT_TUD.1 for trusted updates is used to specify requirements for automatic trusted updates.

The extended requirement FPT_TST.2 is defined to extend the self testing requirements satisfying the O.SELFTEST and O.INTEGRITY security objectives.

The extended requirement FCS_RNG.1 is defined to specify requirements for the Random Number generators to be used for secrets generation or any operation requiring randomization.

5.1 FDP_DSK - Protection of Stored Data

Family behavior
This family is used to mandate the encryption/decryption of stored data.

Component leveling
FDP_DSK.1 is not hierarchical.

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

- change of the encryption key.

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

a) Basic: Start-up of the OS and kernel which involves the decryption of the user data partition of the device.

5.1.1 FDP_DSK.1 - Protection of Stored Data

Hierarchical to: none
Dependencies: FCS_COP.1 Cryptographic operation (to be included by the ST author)

FDP_DSK.1.1 The TSF shall perform encryption of [*assignment: type of data*] in accordance with [*assignment: cryptographic algorithm*], such that no such data is otherwise stored as plain text within the TOE.

FDP_DSK.1.2 The DEK shall be encrypted with a KEK. The DEK shall only exist in persistent memory on the disk.

FDP_DSK.1.3 The TSF shall encrypt all data without user intervention.

FDP_DSK.1.4 The TSF shall be able to decrypt the protected data once the corresponding KEK is presented.

5.2 FDP_ZER - Zeroization

Family behavior
This family is used to define the ability of local and remote secure erase of Critical Security Parameters and classified and personal (address book, calendar, etc) upon request of an authorized user or in emergency situations.

Component leveling
FDP_ZER.1 is not hierarchical.

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

- Perform zeroization upon a request of an authorized user.

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

a) Basic: zeroization execution.

5.2.1 FDP_ZER.1 - Zeroization

Hierarchical to: none

Dependencies: FMT_SMF.1 for the request of an authorised user

FDP_ZER.1.1 The TSF shall be able to securely erase Critical Security Parameters and classified and personal (address book, calendar, etc) data.

FDP_ZER.1.2 The TSF shall be able to perform securely erasure upon request of an authorized user or in the following emergency situations:

a) defined number of initial decryption unsuccessful attempts prior to the TOE boot up (configured by the authorised user);
b) defined number of consecutive failed PIN or KEY-admin authentications attempts (as defined in FIA_AFL.1);
c) the detection of an integrity violation

FDP_ZER.1.3 The TSF shall be able to perform securely erasure starting with the zeroization of the KEK and then with the rest of the data.

FDP_ZER.1.4 The TSF shall be able to perform securely erasure in accordance with a specified method [*assignment: erasure method*] that meets the following: [*assignment: list of standards*].

5.3 FPT_SBT – Secure Boot and Operation continuity

Family behavior
This family is used to specify requirements for secure boot and operation continuity.

Component leveling
FPT_SBT.1 is not hierarchical.

Management
There are no management activities foreseen.

Audit
There are no events defined to be auditable.

5.3.1 FPT_SBT.1 - Secure Boot and Operation continuity

Hierarchical to: none
Dependencies: FMT_SMF.1 for the operation continuity time interval definition.

FPT_SBT.1.1 For the secure boot to be guaranteed:

a) the TOE shall be able to obtain the proper KEK for the disk decryption AND
b) the TOE shall verify the integrity of the OS.

FPT_SBT.1.2 For the operation continuity to be guaranteed:

a) the TOE shall be able to obtain the proper CIK stored in an external device at a configurable time intervals for the disk decryption.
b) the TOE shall enter in a blocked status if the verification fails. Entering in a blocking situation requires the KEK to be provided or derived from user's credentials and OS integrity verification.

5.4 FPT_TUD - Trusted Updates

Family behavior
This family is used to define the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered.

Component leveling
FPT_TUD.1 is not hierarchical.

Management
While management functions have been specified as part of this component already, the following actions could be considered for the management functions in FMT:

- administrator initiation of updates,
- activation and deactivation of automatic updates,
- time for initiation of updates or specification of certificates used for signature verification.

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

- Minimum: Software update.
- Minimum: Failure of verification (digital signature, published hash or version number)

5.4.1 FPT_TUD.1 Trusted Update

Hierarchical to: none
Dependencies: FCS_CKM.1,2,3 & FCS_COP.1 (to be included by the ST author)

FPT_TUD.1.1 The TSF shall provide administrators with the ability to query the current version of the TOE software.

FPT_TUD.1.2 The TSF shall provide a mechanisms that [*selection: on a regular basis initiates, gives administrators the ability to initiate*] updates to TOE software.

FPT_TUD.1.3 The TSF shall provide a means to verify software updates to the TOE using a [*selection: digital signature mechanism, published hash*] prior to installing those updates.

FPT_TUD.1.4 The TSF shall provide a means to verify software updates to the TOE to ensure that software update version is newer than the current version of the TOE prior to installing those updates.

5.5 FPT_TST.2 - Extended integrity and self test

For the specification of these capabilities, it has been used the family FPT_TST defined in [CC31p2] which provides the FPT_TST.1 component specifying some self-test requirements. The new component extends the self-testing to require:

- the verification of the integrity of:
  - the OS during power-up;
  - cryptographic mechanisms during power-up and during their execution;
  - VPN modules and VPN configuration prior to the establishment of a VPN connection;
  - the other CSPs prior to their usage.
- the verification that only authorised Apps will be imported and installed;
- running self-tests during initial start-up and periodically during normal operation to demonstrate the correct operation of the cryptographic functions and the VPN-tunnel;

- providing authorised users with the capability to verify the integrity of VPN configuration and the OS, cryptographic functions and VPN modules executable code;

- defining response actions to be executed in case of integrity or self test fails.

Family behaviour
[*In addition to what is expressed in [CC31p2]*]:

The family defines the requirements for the self-testing of the TSF with respect to some expected correct operation. Integrity tests of the critical functions are also performed. These tests can be carried out at start-up, periodically, at the request of the authorised user, or when other conditions are met. The actions to be taken by the TOE as the result of self testing are defined in the extended component and are linked to components of other families (FPT_FLS.1). The requirements of this family are also needed to detect the corruption of TSF executable code (i.e. TSF SW/FW) and critical functions.

In addition, only apps that are signed with the certificate specified in the Mobile Device policy will be imported and installed into the TOE.

Component Levelling
Hierarchical to FPT_TST.1
FPT_TST.2 extends the self-testing and defines actions to be performed in case of a fail.

Management: FPT_TST.2
There are no management activities foreseen.

Audit: FPT_TST.2
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Basic: Execution of the TSF self tests and integrity tests and their results.

5.5.1 FPT_TST.2 Extended Integrity and selftest

Hierarchical to: none

Dependencies:

FPT_FLS.1 Failure with preservation of secure state.
FCS_COP.1 (to be included by the ST author)

FPT_TST.2.1 The TSF shall verify the integrity of

- the OS during power-up AND
- cryptographic mechanisms during power-up and before their invocation AND
- VPN modules and VPN configuration prior to the establishment of a VPN connection AND
- CSPs prior to their usage,
- *[assignment: other components]*.

FPT_TST.2.2 The TSF shall verify that only authorised Apps will be imported and installed.

FPT_TST.2.3 The TSF shall run a suite of self tests during initial start-up and periodically during normal operation to demonstrate the correct operation of *[selection, choose one of: cryptographic functions and VPN-tunnel set-up, the complete TSF*].

FPT_TST.2.4 The TSF shall provide authorised users with the capability to verify the integrity of [*selection, choose one of: VPN configuration, all TSF data*].

FPT_TST.2.5 The TSF shall provide authorised users with the capability to verify the integrity of [*selection, choose one of: OS and cryptographic functions and VPN modules, the complete TSF*].

FPT_TST.2.6 During integrity checks or self-testing, the TSF shall inhibit the output of data except [*assignment: data that may be output*].

FPT_TST.2.7 The TSF shall execute the following actions in case of fail:
- An integrity failure shall cause an immediate emergency zeroization and TOE powered off.
- An App not signed with the certificate specified in the Mobile Device policy shall not be imported and installed into the TOE.
- A cryptographic self test failure shall cause no VPN tunnel to be established. The user shall be warned that the VPN-tunnel has not been established.
- A VPN-tunnel setup failure shall cause all IP-traffic selected for its transmission through the tunnel to be blocked.
- Any failure shall cause the output data to be inhibited.

5.6 FCS_RNG – Random Number Generation

This section describes the functional requirements for the generation of random numbers to be used for secrets generation in cryptographic processes.

Family behavior
This family defines quality requirements for the generation of random numbers.

Component leveling
FCS_RNG.1 is not hierarchical.

Management
There are no management activities foreseen.

Audit
There are no events defined to be auditable.

5.6.1 FCS_RNG.1 – Random Number Generation

Hierarchical to: none
Dependencies: none

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

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

6 Security Requirements

The Security Functional Requirements included in this section are derived from [CC31p2], with additional extended functional components.

6.1 TOE Security Functional Requirements

This section describes the Security Functional Requirements for the TOE.

6.1.1 Class FAU – Security Audit

6.1.1.1 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 [*selection*: not specified] level of audit; and
c) *[assignment:*

1. Power-up and shutdown of the OS and kernel;
2. Self-tests and integrity checks;
3. User and administrator login;
4. Initiation and termination of trusted channel (VPN-tunnel or remote administration);
5. Initiation of software update;
6. Initiation of app installation or update by the user;
7. Success or failure of signature verification for software updates.
8. Success or failure of signature verification for applications.
9. All administrative actions of TOE configuration;
10. Attempt to bypass the VPN-tunnel;
11. Zeroization event;
12. *[assignment: other specifically defined auditable events]]*.

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, *[assignment: other audit relevant information]*.

6.1.1.2 FAU_GEN.2 User identity association

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

6.1.1.3 FAU_SAR.1 Audit review

FAU_SAR.1.1 The TSF shall provide [*assignment*: administrator] with the capability to read [*assignment*: all audited events and record contents] 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.

6.1.1.4 FAU_STG.1 Protected audit trail storage

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

FAU_STG.1.2 The TSF shall be able to [*selection*: prevent] unauthorised modifications to the stored audit records in the audit trail.

6.1.1.5 FAU_STG.4 Prevention of audit data loss

FAU_STG.4.1 The TSF shall [*selection, choose one of: "ignore audited events", "prevent audited events, except those taken by the authorised user with special rights", "overwrite the oldest stored audit records"*] and [*assignment: other actions to be taken in case of audit storage failure*] if the audit trail is full.

6.1.2 Class FCS - Cryptographic Support

All the cryptographic security functions shall provide, at least, 128 bits of security. For digital signature or authentication services hashes shall be at least of 256 bits.

The ST author claiming conformance to this PP shall specify the cryptographic functions implementing the following functionality:

1. TOE updates, apps installation and execution. Digital signature.

This mechanism is used for the verification of the TOE updates (FPT_TUD.1) and assuring that only allowed apps are installed.

Updates to the TOE and apps are signed by an authorized source. Only authorised Apps will be imported and installed (FPT_TST.2). The apps may have additional signatures that will ensure that they will be installed as apps with a privilege to communicate outside of the VPN tunnel (as specified in FDP_IFC.2/FDP_IFF.1).

If certificates are used, the ST author shall specify how the certificates used by the verification mechanism are contained on the device. If the certificate is to be entered or internally generated, the ST author shall include any of the following SRFs to satisfy the dependency:

- FDP_ITC.1 Import of user data without security attributes,

- FDP_ITC.2 Import of user data with security attributes,
- FCS_CKM.1 Cryptographic key generation.

Key destruction is covered by the FDP_ZER.1 SFR.

TOE updates verification may also be performed by published hashes.

2. Integrity verification. Cryptographic Hashing.
This mechanism is used for the integrity monitoring (FPT_TST.2 - OS, cryptographic functions, VPN-modules, VPN-configuration data and other CSPs.).

In addition cryptographic hash functions are also used for digital signatures.

3. Trusted channels. Cryptographic schemes are to be specified for the implementation of trusted channels defined:
   i. VPN-tunnel (FTP_ITC.1/VPN-tunnel)
   ii. CIK-tunnel FTP_ITC.1/CIK-tunnel)
   iii. Other trusted channels to be implemented for the remote administration (FTP_ITC.1/REM-ADM) and for sending the audit logs to an external entity (FTP_ITC.1/AUDIT).

4. Disk encryption. Symmetric algorithms are to be defined for the disk encryption capability with the DEK (FDP_DSK.1). In addition, the ST author shall specify also the following:
   i. KDF for the derivation of the KEK from the user's credentials;
   ii. Encryption / decryption of the DEK.

6.1.2.1 Random Number Generation

All the cryptography operations and other processes requiring the generation of secrets shall be supported by an RNG meeting the requirements specified in this section.

6.1.2.1.1 FCS_RNG.1 – Random Number Generation

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

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

Application note

The ST author shall resolve the operations according to [AIS 20] and [AIS31]. The strength and quality metrics shall meet [AIS20].

6.1.3 Class FDP - User Data Protection

6.1.3.1 FDP_IFC.2 - Complete information flow control

FDP_IFC.2.1 The TSF shall enforce the [*assignment:* TRAFFIC SFP] on [*assignment:* incoming data packages or frames based on incoming interface and when using the VPN-tunnel on the destination address of the package] 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 TSF shall enforce the TRAFFIC SFP on the IP data traffic between the TOE and an external IT entity (using the mobile data network or WiFi network interfaces) to ensure that all IP data traffic, with the exception from specific signed Apps on the TOE, is always sent and/or received using the VPN tunnel.

6.1.3.2 FDP_IFF.1 - Simple security attributes

FDP_IFF.1.1 The TSF shall enforce the [*assignment:* TRAFFIC SFP] based on the following types of subject and information security attributes: [*assignment:*

Subjects
- App communicating using IP data

Objects
- IP packet sent or received

Security attributes
- App being signed application
- App being signed for outside-channel

Operations
- pass or reject the package]

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: [*assignment:*
- incoming package arriving on the clear interface will be passed to the appropriate VPN channel, when using VPN-tunnel based on the destination address
- incoming package arriving on the crypto interface inside a VPN channel will be passed to the internal network to any signed App].

FDP_IFF.1.3 The TSF shall enforce the [*assignment:* no additional information flow control SFP rules].

FDP_IFF.1.4 The TSF shall explicitly authorise an information flow based on the following rules: [*assignment:* outgoing/incoming traffic from/to specific signed apps allowed communicating outside the channel].

FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: [*assignment:* data packages arriving on the crypto interface not coming through the VPN channel, except for the following packets:

- Protocol packages being part of the VPN connection
- ICMPv4 Fragmentation Required
- ICMPv4 Echo Request
- ICMPv4 Echo Reply]

6.1.3.3 FDP_DSK.1 Extended - Protection of Data on Disk

FDP_DSK.1.1 The TSF shall perform encryption of [*assignment: type of data*] in accordance with [*assignment: cryptographic algorithm*], such that no such data is otherwise stored as plain text within the TOE.

FDP_DSK.1.2 The DEK shall be encrypted with a KEK. The DEK shall only exist in persistent memory on the disk.

FDP_DSK.1.3 The TSF shall encrypt all data without user intervention.

FDP_DSK.1.4 The TSF shall be able to decrypt the protected data once the corresponding KEK is presented.

Application Note
The intent of this requirement is to specify that encryption of any protected data will not depend on a user electing to protect that data. The encryption specified in FDP_DSK.1 occurs transparently to the user and the decision to protect the data is outside the discretion of the user, which is a characteristic that distinguishes it from file encryption, which not only involves user action, but also allows the user to select the encryption key. The KEK used for the decryption of the DEK may be entered or derived from users credentials. The DEK will be used for the user data decryption. The cryptographic mechanism referenced in the first element must be specified in FCS_COP.1 (to be included by the ST author).

6.1.3.4 FDP_ZER.1 Extended - Zeroization

FDP_ZER.1.1 The TSF shall be able to securely erase Critical Security Parameters and classified and personal (address book, calendar, etc) data.

FDP_ZER.1.2 The TSF shall be able to perform securely erasure upon request of an authorized user or in the following emergency situations:

a) defined number of initial decryption unsuccessful attempts prior to the TOE boot up (configured by the authorised user);
b) defined number of consecutive failed PIN or KEY-admin authentications attempts (as defined in FIA_AFL.1);
c) the detection of an integrity violation

FDP_ZER.1.3 The TSF shall be able to perform securely erasure starting with the zeroization of the KEK and then with the rest of the data.

FDP_ZER.1.4 The TSF shall be able to perform securely erasure in accordance with a specified method [*assignment: erasure method*] that meets the following: [*assignment: list of standards*].

Application Note
The implementation of this SFR shall take into account the following destruction methods (to be used by the ST author to resolve the "assignment"):

- For volatile memory, the destruction shall be executed by a single direct overwrite consisting of zeroes following by a read-verify.
- For non-volatile EEPROM, the destruction shall be executed by a single direct overwrite consisting of a pseudo random pattern followed a read-verify.
- For non-volatile flash memory, the destruction shall be executed by a single direct overwrite consisting of zeros followed by a read-verify or by a block erase followed by a read-verify
- For non-volatile memory other than EEPROM and flash, the destruction shall be executed by overwriting three or more times with a random pattern that is changed before each write.

6.1.4 Class FIA - Identification and Authentication

6.1.4.1 FIA_UAU.2/KEK - User Authentication before any action

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

Application note
There is no user identification required since handsets are assumed to be used and under the control of one user only (OE.SINGLEUSER). Local authentication is performed during startup by providing a correct user key (KEK) using an allowed interface for that purpose (for example, NFC-tag, QR-code or hexadecimal key) or deriving it from the user's credentials. This will unlock the handset and simultaneously decrypt the DEK and subsequently, decrypt the user data partition of the device (see FDP_DSK.1). In addition, periodically, the CIK is entered to the TOE from an external device through a trusted channel. This prevents the TOE to be automatically blocked.

6.1.4.2 FIA_UAU.2/PIN - User Authentication before any action

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

Application note
There is no user identification required since handsets are assumed to be used and under the control of one user only (OE.SINGLEUSER). Local authentication is performed on a running system by providing a correct user PIN that will unlock the handset. Authentication may be based on at least a PIN with minimum 4 digits, a biometry mechanism or any other mechanism with an equivalent strength. It also may be used for the TOE secure boot deriving the proper KEK from the entry provided by the user.

6.1.4.3 FIA_UAU.2/KEY-admin - User Authentication before any action

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

Application note
This administrator key is necessary to unlock the device for TOE configuration changes. To update a mobile device with new crypto keys or policy it is required the TOE to be unlocked by an authorized administrator using a key or code unique for the particular device (for example a QR-code). After the new policy or keys are installed, the authorized administrator must set the device into locked state again using a special key or code. It also may be used for the TOE secure boot deriving the proper KEK from the entry provided by the user.

6.1.4.4 FIA_AFL.1 - Authentication failure handling

FIA_AFL.1.1 The TSF shall detect when [*selection:* an administrator configurable positive integer within [*assignment:* 3 to 15]] unsuccessful authentication attempts occur related to [*assignment:*

- user PIN or KEY-admin authentication on a running system OR
- user PIN or KEY-admin authentication or KEK during start-up].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been [*selection:* met], the TSF shall [*assignment:* perform emergency erase of the data stored on the handset].

Application note
The PIN or KEY-admin authentication is the authentication and unlocking mechanism of a running system that will give user access to the screen and the external interfaces. Like the KEK, the PIN or KEY-admin also may be used for the TOE secure boot deriving the proper KEK from the entry provided by the user. In order to prevent brute-force attacks, the

device will perform a secure wipe after ***n*** consecutive failed authentications attempts (the range of acceptable values of ***n*** shall be [3-15]).

The process and the data to be erased are detailed in FDP_ZER.1.

6.1.5 Class FMT - Security Management

6.1.5.1 FMT_SMF.1 - Specification of management functions

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

- Change encryption key (for local data storage);
- Installation of apps from the local app market;
- Query VPN-tunnel info;
- Change the settings (keep alive setting, reset user settings; restart the VPN, tunnel bypass capability, etc.);
- TOE updates management;
- Emergency erase;
- User sessions management;
- Operation continuity time interval;
- Number of initial decryption unsuccessful attempts to enter an emergency situation
- Number of consecutive failed PIN/KEY-admin/KEK authentications attempts to enter an emergency situation
- Audit management (sending audit logs to an external entity);
- *[assignment: Other functions*]]

Application note
The security management functions are available to the user of the handset through the corresponding App. Remote administration is allowed using the defined trusted channel (as defined in FTP_ITC.1/REM-ADM).

TOE updates management includes administrator initiation of updates, activation and deactivation of automatic updates, time for initiation of updates or specification of certificates used for signature verification.

6.1.5.2 FMT_SMR.1 - Security management roles

FMT_SMR.1.1 The TSF shall maintain the roles [*assignment:* user and administrator].

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

6.1.5.3 FMT_MSA.1 Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the [*assignment:* TRAFFIC SFP] to restrict the ability to [*selection: change_default, query, modify, delete, [assignment: other operations]*] the security attributes [*assignment*: security attributes belonging to the TRAFFIC SFP] to [*assignment:* administrator].

6.1.5.4 FMT_MSA.3 Static attribute initialisation

FMT_MSA.3.1 The TSF shall enforce the [*assignment:* TRAFFIC SFP] to provide [*selection:* restrictive] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [*assignment:* administrator] to specify alternative initial values to override the default values when an object or information is created.

6.1.6 Class FPT - Protection of the TOE Security Functions

6.1.6.1 FPT_FLS.1 - Failure with preservation of secure state

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: [*assignment:*

- failed the OS integrity check during power-up;
- failed cryptographic mechanisms integrity check during power-up and during their execution;
- failed VPN modules and VPN configuration integrity check prior to the establishment of a VPN connection;
- failed the other CSPs integrity check prior to their usage;
- failed self-test of cryptographic functions;
- failed self-test of VPN-tunnel set-up]

Application note
Preserve a secure state implies the applicable actions to be performed in case of error defined in FPT_TST.2.

6.1.6.2 FPT_SBT.1 - Secure Boot and Operation continuity

FPT_SBT.1.1 For the secure boot to be guaranteed:

a) the TOE shall be able to obtain the proper KEK for the disk decryption AND
b) the TOE shall verify the integrity of the OS.

FPT_SBT.1.2 For the operation continuity to be guaranteed:

a) the TOE shall be able to obtain the proper CIK stored in an external device at a configurable time intervals for the disk decryption.
b) the TOE shall enter in a blocked status if the verification fails. Entering in a blocking situation requires the KEK to be provided or derived from user's credentials and OS integrity verification.

Application Note
The intent of this requirement is to specify that boot of the TOE only success if the integrity of the OS is guaranteed and the proper external KEK for the disk decryption is provided by the user.

Periodically, the TOE shall verify at configurable time intervals, the KEK stored in an external device through a trusted channel. In this case the KEK stored in the device is called "Crypto Ignition Key" (CIK). Fail to the verification shall cause the TOE to be blocked. Entering in a blocking situation requires the KEK to be provided or derived from user's credentials and a similar secure boot process shall be performed.

6.1.6.3 FPT_STM.1 - Reliable time stamps

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

Application note
The TOE provides reliable time stamps as part of the internal clock of the handset. The time can be set by the network environment using the NTP service provided over a trusted channel. Should this be the case, the ST author shall include the associated SFRs to define the trusted channel.

Time stamps are used to verify the validity of certificates (e.g. the server certificate for the VPN-tunnel) and for the audit trail generation.

6.1.6.4 FPT_TST.2 Extended Integrity and self test

FPT_TST.2.1 The TSF shall verify the integrity of

- the OS during power-up AND
- cryptographic mechanisms during power-up and before their invocation AND
- VPN modules and VPN configuration prior to the establishment of a VPN connection AND
- CSPs prior to their usage,
- *[assignment: other components]*.

FPT_TST.2.2 The TSF shall verify that only authorised Apps will be imported and installed.

FPT_TST.2.3 The TSF shall run a suite of self tests during initial start-up and periodically during normal operation to demonstrate the correct operation of [*selection, choose one of: cryptographic functions and VPN-tunnel set-up, the complete TSF*].

FPT_TST.2.4 The TSF shall provide authorised users with the capability to verify the integrity of [*selection, choose one of: VPN configuration, all TSF data*].

FPT_TST.2.5 The TSF shall provide authorised users with the capability to verify the integrity of [*selection, choose one of: OS and cryptographic functions and VPN modules, the complete TSF*].

FPT_TST.2.6 During integrity checks or self-testing, the TSF shall inhibit the output of data except [*assignment: data that may be output*].

FPT_TST.2.7 The TSF shall execute the following actions in case of fail:

- An integrity failure shall cause an immediate emergency zeroization and TOE powered off.
- An App not signed with the certificate specified in the Mobile Device policy shall not be imported and installed into the TOE.
- A cryptographic self test failure shall cause no VPN tunnel to be established. The user shall be warned that the VPN-tunnel has not been established.
- A VPN-tunnel setup failure shall cause all IP-traffic selected for its transmission through the tunnel to be blocked.
- Any failure shall cause the output data to be inhibited.

Application note

The TSF integrity testing shall consist of an integrity check of the software executables implementing the whole TSF or, at least, the executable code of the OS (during power-up), the cryptographic mechanisms (during power-up and before their invocation) and the VPN modules and VPN configuration (prior to the establishment of a VPN connection).

In case of apps, its digital signature is to be verified to check that the app is authorised for its installation. The cryptographic algorithm is to be specified in the corresponding FCS_COP.1 SFR.

Self-test are run during start-up and on a regular basis to demonstrate the correct operation of, at least, the cryptographic functions (for example executing known answer tests of the algorithms) and VPN-tunnel set-up.

A user cannot influence the TSF integrity and self testing, but can force it by restarting the TOE.

6.1.6.5 FPT_TUD.1 Trusted Update

FPT_TUD.1.1 The TSF shall provide administrators with the ability to query the current version of the TOE software.

FPT_TUD.1.2 The TSF shall provide a mechanism that [*selection:* gives administrators the ability to initiate] updates to TOE software.

FPT_TUD.1.3 The TSF shall provide a means to verify software updates to the TOE using a [*selection: digital signature mechanism, published hash*] prior to installing those updates.

FPT_TUD.1.4 The TSF shall provide a means to verify software updates to the TOE to ensure that software update version is newer than the current version of the TOE prior to installing those updates.

Application note
The trusted update is a user controlled mechanism where updates are made available, but the user has to cooperate and download and install the update. Updates to the TOE could be signed (their hashes) by an authorized source or published hashes are available. The ST author should choose the mechanism implemented by the TOE; it is acceptable to implement both mechanisms. The digital signature and/or hashes mechanisms shall be defined by the ST author including the corresponding FSC_COP.1 SFRs.

6.1.6.6 FPT_PHP.1 Passive detection of physical attack

FPT_PHP.1.1 The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.

FPT_PHP.1.2 The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.

Application note
The enclosure of the TOE shall be protected with tamper-evident seals.

6.1.7 Class FTA - TOE access

6.1.7.1 FTA_SSL.1 TSF-initiated session locking

FTA_SSL.1.1 The TSF shall lock an interactive session after [*assignment:* a user configurable time that is less than 30 minutes of user inactivity] by:

a. clearing or overwriting display devices, making the current contents unreadable;
b. disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.1.2 The TSF shall require the following events to occur prior to unlocking the session: [*assignment:* user authentication OR administrator authentication OR CIK provided by an external entity according to the scenarios defined in the application note].

Application note
The session will be locked in two situations:

1. Because of user inactivity: user configurable time that is less than 30 minutes. The unlocking requires the user authentication as described in FIA_UAU.2/PIN or administrator authentication as described in FIA_UAU.2/KEY-admin.
2. Periodically, at a configurable time interval the TOE requires an external entity to provide the CIK for the operation continuity. Fail to the verification shall cause the TOE to be blocked. Entering in a blocking situation requires the KEK to

be provided or derived from user's credentials and a similar secure boot process shall be performed.

6.1.7.2 FTA_SSL.2 User-initiated locking

FTA_SSL.2.1 The TSF shall allow user-initiated locking of the user's own interactive session, by:

a. clearing or overwriting display devices, making the current contents unreadable;
b. disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.2.2 The TSF shall require the following events to occur prior to unlocking the session: [*assignment:* user PIN OR KEY-admin authentication].

Application note
The unlocking requires the user authentication as described in FIA_UAU.2/PIN or administrator authentication with as described in FIA_UAU.2/KEY-admin.

6.1.8 Class FTP - Trusted channels

6.1.8.1 FTP_ITC.1/VPN-tunnel Inter-TSF Trusted Channel (Application communications)

FTP_ITC.1.1/VPN-tunnel 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/VPN-tunnel The TSF shall permit [*selection:* the TSF] to initiate communication via the trusted channel.

FTP_ITC.1.3/VPN-tunnel The TSF shall initiate communication via the trusted channel for [*assignment:* VPN services].

Application note
This channel is the VPN communication channel (VPN tunnel) that the TOE may establish from the TOE to the VPN endpoint. The VPN tunnel is used for all IP communication between the TOE applications and the outside world, with the exception of the apps that explicitly are allowed to break out of the tunnel. See the TRAFFIC SFP described in section 6.1.3.1. The cipher suites used for the implementation of this channel shall be defined by the ST author including the corresponding SFRs of the FCS_CKM and FCS_COP families. These cryptographic algorithms shall provide, at least, 128 bits of security.

6.1.8.2 FTP_ITC.1/CIK-tunnel Inter-TSF Trusted Channel

FTP_ITC.1.1/CIK-tunnel 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/CIK-tunnel The TSF shall permit [*selection:* the TSF] to initiate communication via the trusted channel.

FTP_ITC.1.3/CIK-tunnel The TSF shall initiate communication via the trusted channel for [*assignment:* entering the CIK for operation continuity].

Application note
This channel is the trusted channel established to enter the Crypto Ignition Key" (CIK) into the TOE. The cipher suites used for the implementation of this channel shall be defined by the ST author including the corresponding SFRs of the FCS_CKM and FCS_COP families. These cryptographic algorithms shall provide, at least, 128 bits of security.

6.1.8.3 FTP_ITC.1/REM-ADM Inter-TSF Trusted Channel (remote administration)

FTP_ITC.1.1/REM-ADM 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/REM-ADM The TSF shall permit [*selection:* ~~another~~ a remote trusted IT product] to initiate communication via the trusted channel.

FTP_ITC.1.3/REM-ADM The TSF shall initiate communication via the trusted channel for [*assignment:* remote administration].

Application note
This channel is the trusted channel established for the remote administration. The cipher suites used for the implementation of this channel shall be defined by the ST author including the corresponding SFRs of the FCS_CKM and FCS_COP families. These cryptographic algorithms shall provide, at least, 128 bits of security.

6.1.8.4 FTP_ITC.1/AUDIT Inter-TSF Trusted Channel

FTP_ITC.1.1/AUDIT 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/AUDIT The TSF shall permit [*selection:* ~~another~~ a remote trusted IT product] to initiate communication via the trusted channel.

FTP_ITC.1.3/AUDIT The TSF shall initiate communication via the trusted channel for [*assignment:* sending audit logs to an external entity].

Application note
This channel is the trusted channel established for sending the audit log to an external entity. The cipher suites used for the implementation of this channel shall be defined by the ST author including the corresponding SFRs of the FCS_CKM and FCS_COP families. These cryptographic algorithms shall provide, at least, 128 bits of security.

6.2 Rationale for Security Functional Requirements

NOTE: for the necessary and sufficiency analysis of the SFRs to meet the security objectives, the ST author will have to map the cryptographic SFRs belonging to the FCS_COP and FCS_CKM families to the specific security objectives that shall be supported by cryptographic functions (see section 6.1.2 Class FCS - Cryptographic Support).

6.2.1 Coverage

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

	O.TUNNEL	O.INSTALLATION	O.SECURE_BOOT	O.ERASURE	O.INTEGRITY	O.SELFTEST	O.OS_UPDATE	O.AUTHENTICATION	O.ADMIN	O.SECURITY_POLICIES	O.CRYPT_PROTECTION	O.SECURITY_DATA	O.AUDIT	O.HW_TAMPER	O.RNG
FAU_GEN.1													X		
FAU_GEN.2													X		
FAU_SAR.1													X		
FAU_STG.1													X		
FAU_STG.4													X		
FCS_RNG.1															X
FDP_IFC.2	X														
FDP_IFF.1	X														
FDP_DSK.1			X					X			X				
FDP_ZER.1				X							X				
FIA_UAU.2/KEK			X					X							
FIA_UAU.2/PIN			X					X							
FIA_UAU.2/KEY-admin			X					X	X	X	X	X			

FIA_AFL.1				X				X							
FMT_SMF.1				X				X		X	X	X	X		
FMT_SMR.1									X	X		X			
FMT_MSA.1	X														
FMT_MSA.3	X														
FPT_FLS.1					X	X					X				
FPT_SBT.1			X												
FPT_STM.1	X							X					X		
FPT_TST.2		X	X	X	X	X					X				
FPT_TUD.1							X								
FPT_PHP.1														X	
FTA_SSL.1								X							
FTA_SSL.2								X							
FTP_ITC.1/VPN-tunnel	X														
FTP_ITC.1/REM-ADM	X														
FTP_ITC.1/AUDIT	X												X		
FTP_ITC.1/CIK-tunnel	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.

O.TUNNEL
The TOE implements the following trusted channels: VPN-tunnel, CIK-tunnel and trusted channels for the remote administration or when sending the audit log to an external entity.

FDP_IFC.2, FDP_IFF.1, FMT_MSA.1, FMT_MSA.3: implement the information flow control policy of the IP traffic for the VPN-tunnel.
FTP_ITC.1/VPN-tunnel ensures the VPN trusted channel properties.

FTP_ITC.1/REM-ADM ensures the trusted channel properties for remote administration

FTP_ITC.1/AUDIT ensures the trusted channel properties when sending the audit logs to an external entity.

FTP_ITC.1/CIK-tunnel ensures the trusted channel properties when entering the Crypto Ignition Key" (CIK).

FPT_STM.1 provides the time for certificate verification.

In addition, the applicable cryptographic support SFRs (FCS class) included by the author of the ST as per section 6.1.2, contribute to meet and achieve the security objective. They shall be included in this section together with the corresponding justification to complete this rationale.

O.INSTALLATION

The TOE shall be able to install only authorized applications. Only applications signed by the organisation can be installed on the mobile device.

FPT_TST.2 - FPT_TST.2.1 requires the TOE to verify that the app is authorised to be imported and installed.

In addition, the applicable cryptographic support SFRs (FCS class) included by the author of the ST as per section 6.1.2, contribute to meet and achieve the security objective. They shall be included in this section together with the corresponding justification to complete this rationale.

O.SECURE_BOOT
This objective addresses the **secure boot** (disk encryption/decryption using the credentials/KEK provided by the user) and the **operation continuity** (periodic verification of the KEK stored in an external device) processes.

Boot of the TOE only successes if the integrity of the OS is guaranteed and the proper KEK for the disk decryption is provided by the user or obtained by derivation from the user's credentials input (FPT_TST.2, FIA_UAU.2/KEY-admin, FIA_UAU.2/PIN, FIA_UAU.2/KEK, FPT_SBT.1, FDP_DSK.1).

For the operation continuity, periodically, the TOE shall verify at configurable time intervals, the KEK stored in an external device through a trusted channel. In this case the KEK stored in the device is called "Crypto Ignition Key" (CIK). Fail to this verification shall cause the TOE to be blocked. Entering in a blocking situation requires the KEK to be provided or derived from user's credentials and a similar secure boot process shall be performed (FTP_ITC.1/CIK-tunnel, FPT_SBT.1).

O.ERASURE
The TOE shall be able to perform local and remote securely erase of Critical Security Parameters and classified and personal (address book, calendar, etc) upon request of an authorized user or in emergency situations.

Security zeroization is specified in FDP_ZER.1. This process is to be implemented starting with the zeroization of the KEK and then with the rest of the data. Emergency situations described are:

- defined number of initial decryption unsuccessful attempts prior to the TOE boot up.
- defined number of consecutive failed authentications attempts (as defined in FIA_AFL.1);
- the detection of an integrity violation

FMT_SMF.1 specifies management functions to set the number of initial decryption unsuccessful attempts and the number of consecutive failed PIN/KEY-admin/KEK authentications attempts which cause entering in an emergency situation. It also provides the "Emergency erase" function.

FPT_TST.2 specifies the integrity violation scenarios causing the emergency zeroization.

O.INTEGRITY
The TOE shall be able to verify the integrity of:

- the OS during power-up;
- cryptographic mechanisms during power-up and during their execution;
- VPN modules and VPN configuration prior to the establishment of a VPN connection;
- the other CSPs prior to their usage.

The TOE shall be able to verify that only authorised Apps will be imported and installed.

These scenarios along with the actions in case of integrity violation are specified in FPT_TST.2 and FPT_FLS.1.

O.SELFTEST
Selftests to be executed by the TOE are defined in FPT_TST.2: cryptographic functions and VPN-tunnel set-up. Scenarios and actions in case of failure are specified in FPT_FLS.1 and FPT_TST.2.

O.OS_UPDATE
This objective addresses the capability of verifying that TOE updates are authorised before installing them.
This process is defined in FPT_TUD.1 where it is expected that the user downloads and install the update. Updates to the TOE could be signed (their hashes) by an authorized source or published hashes are available.

O.AUTHENTICATION
The TOE shall be able to authenticate the authorized users. The security mechanism used to authenticate TOE users shall be resistant to brute force attacks.

Authentication mechanisms are specified in

- FIA_UAU.2/KEK: Local authentication is performed during startup by providing a correct user key (KEK) using an allowed interface for that purpose (for example, NFC-tag, QR-code or hexadecimal key). This will unlock the handset and simultaneously decrypt the DEK and subsequently, decrypt the user data partition of the device (FDP_DSK.1). In addition, periodically, the CIK is entered to the TOE from an external device through a trusted channel.
- FIA_UAU.2/PIN: Local authentication is performed on a running system (or TOE start-up) by providing a correct user PIN that will unlock the handset. It also may be used for the TOE secure boot deriving the proper KEK from the entry provided by the user.
- FIA_UAU.2/KEY-admin: This administrator key is necessary to unlock the device for TOE configuration changes. It also may be used for the TOE secure boot deriving the proper KEK from the entry provided by the user.

Authentication failure handling also contributes in meeting the objective through FIA_AFL.1 to prevent brute force attacks over the PIN/KEY-admin/KEK as the device will perform a secure wipe after **n** consecutive failed authentications attempts (**n** is set by FMT_SMF.1).

FPT_STM.1 provides the time for session locking.
FTA_SSL.1 ensures the timeout for session locking.
FTA_SSL.2 ensures the user initiated session locking.

O.ADMIN
The TOE shall be able to restrict security configuration privilege escalation to authorized users.

FIA_UAU.2/KEY-admin, allows the administrator to authenticate and unlock the configuration of the handset, and prevent any other user can do this.
FMT_SMR.1, the TOE identifies two distinct roles, the user and the administrator.

O.SECURITY_POLICIES
The TOE shall be able to add and execute security policies and rules that prevent unauthorized access to the security features that the TOE manages.

FIA_UAU.2/KEY-admin, will ensure that the administrator is properly authenticated by presenting the administrator unlock key before being allowed to make the changes.
FMT_SMF.1, specifies the management functions for setting up VPN-tunnel bypass capability and other security policies.
FMT_SMR.1, the TOE identifies two distinct roles, the user and the administrator.

O.CRYPT_PROTECTION
The TOE shall be able to protect cryptographic assets from unauthorized access, retrieval or modification.

Disk encryption and zeroization process (FDP_DSK.1, FDP_ZER.1) contributes protecting the cryptographic material.
FIA_UAU.2/KEY-admin, will ensure that the administrator is properly authenticated by presenting the administrator unlock key before being allowed to make the changes.
FMT_SMF.1, will ensure that the administrator, and only the administrator is able to make changes to the security policies, including the cryptographic properties of the TOE.
FPT_TST.2 and FPT_FLS.1 will ensure the integrity of the TSF and perform cryptographic tests during startup, before the VPN tunnel is established protecting this way the associated cryptographic material.

O.SECURITY_DATA
The TOE shall be able to protect the entire security configuration from unauthorized access or modification.

FIA_UAU.2/KEY-admin, will ensure that only authorized administrators can change the configuration of the TOE.

FMT_SMF.1, specifies the management functions to change the configuration of the TOE. FMT_SMR.1, the TOE identifies two distinct roles, the user and the administrator. Only the administrator has the key to unlock the handset and change the configuration.

O.AUDIT
This objective addresses the audit logs generation and their protection.

The TOE must record security relevant events and associate each event with the identity of the user that caused the event (FAU_GEN.1, FAU_GEN.2, FPT_STM.1).

The TOE must prevent unauthorized modification of the audit trail, prevent loss of audit trail data, and provide authorized administrators with the ability to review the audit trail (FAU_SAR.1, FAU_STG.1, FAU_STG.4).

The TOE shall be able to send the audit trail to an external entity when a security management function requires it. FMT_SMF.1 specifies the management functions for audit management including sending the audit logs to an external entity. This shall be performed through a trusted channel (FTP_ITC.1/AUDIT).

O.HW_TAMPER
The enclosure of the TOE shall be protected with tamper-evident seals (FPT_PHP.1).

O.RNG
The TOE must implement random number generators meeting the requirements of strength and quality metrics specified in [AIS20] and [AIS31] (FCS_RNG.1).

6.2.3 Security Requirements Dependency Analysis

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 following table demonstrates the dependencies of SFRs modelled in CC Part 2 and how the SFRs for the TOE resolve those dependencies.

SFR	Dependencies	Resolution / Rationale
FAU_GEN.1	FPT_STM.1 Reliable time stamps	YES
FAU_GEN.2	FAU_GEN.1 Audit data generation FIA_UID.1 Timing of identification	YES NO. There is no user identification required since handsets are assumed to be used and under the control of one user only.
FAU_SAR.1	FAU_GEN.1 Audit data	YES

	generation	
FAU_STG.1	FAU_GEN.1 Audit data generation	YES
FAU_STG.4	FAU_STG.1 Protected audit trail storage	YES
FCS_RNG.1	NA	NA
FDP_IFC.2	FDP_IFF.1 Simple security attributes	YES
FDP_IFF.1	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation	YES YES
FDP_DSK.1	FCS_COP.1 Cryptographic operation	NO. Cryptographic functions are to be included by the ST author
FDP_ZER.1	FMT_SMF.1	YES. Satisfied for the request of an authorised user.
FIA_UAU.2/KEK	FIA_UID.1 Timing of identification	NO. There is no user identification required since handsets are assumed to be used and under the control of one user only.
FIA_UAU.2/PIN	FIA_UID.1 Timing of identification	NO. There is no user identification required since handsets are assumed to be used and under the control of one user only.
FIA_UAU.2/KEY-admin	FIA_UID.1 Timing of identification	NO. There is no user identification required since handsets are assumed to be used and under the control of one user only.
FIA_AFL.1	FIA_UAU.1 Timing of authentication	YES
FMT_SMF.1	NA	NA
FMT_SMR.1	FIA_UID.1 Timing of identification	NO. There is no user identification required since handsets are assumed to be used and under the control of one user only.
FMT_MSA.1	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	YES by FDP_IFC.2 (TRAFFIC policy) YES YES
FMT_MSA.3	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	YES, for FDP_IFC.2 (TRAFFIC policy) YES
FPT_FLS.1	NA	NA
FPT_SBT.1	FMT_SMF.1	YES. Satisfied for the operation continuity time interval definition.
FPT_STM.1	NA	NA
FPT_TST.2	FPT_FLS.1 Failure with preservation of secure state. FCS_COP.1	YES NO. Cryptographic functions are be included by the ST author
FPT_TUD.1	FCS_CKM.1,2,3 &	NO. Cryptographic functions are be included by the ST

	FCS_COP.1	author
FPT_PHP.1	NA	NA
FTA_SSL.1	FIA_UAU.1 Timing of authentication	YES
FTA_SSL.2	FIA_UAU.1 Timing of authentication	YES
FTP_ITC.1/VPN-tunnel	NA	NA
FTP_ITC.1/REM-ADM	NA	NA
FTP_ITC.1/AUDIT	NA	NA
FTP_ITC.1/CIK-tunnel	NA	NA

6.3 Security Assurance Requirements

The assurance level selected for this PP is EAL2 with the ALC_FLR.2 augmentation.

The TOE security assurance requirements, summarized in the next table, identify the management and evaluative activities required to address the threats and policies identified in Section 3 of this PP.

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

Application note

ADV_ARC.1 includes the separation of domains as a security property that shall be exhibited by the TOE design. For the TOEs claiming conformance with this PP, it is not allowed the declaration of a single domain as it is expected the isolation of the cryptographic material and also the separation of classified and unclassified data. Therefore, the developer shall describe the security domains and the TOE shall implement suitable mechanisms to keep them separate.

6.4 Rationale for Security Assurance Requirements

The current PP is claimed to be conformant with the assurance package EAL 2 augmented by the assurance component ALC_FLR.2.

EAL2 was chosen to provide a low to moderate level of assurance that is consistent with good commercial practices. As such, minimal additional tasks are placed upon the vendor assuming the vendor follows reasonable software engineering practices and can provide support to the evaluation for design and testing efforts. The TOE is expected to be in possession of a single user and controlled by the organisation being the threats those specified for the intended environment.

The augmentation of ALC_FLR.2 was chosen to give greater assurance of the developer's on-going flaw remediation processes.