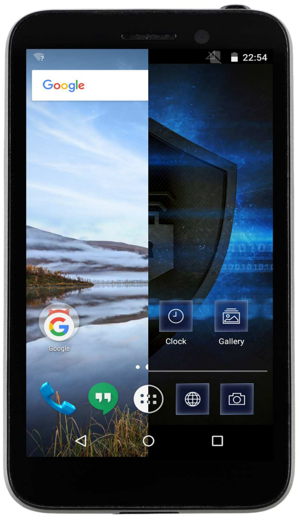
TOUGH MOBILE C



Security Target Lite

Common Criteria – EAL2

1 VERSION HISTORY

Version	Date	Author	Notes	Version
1.0.1	17.05.2019	Risto Vitikka	Corrections after review	Approved

Table of Contents

2 SECURITY TARGET INTRODUCTION

This document is the Common Criteria Security Target for Bittium Tough Mobile C. This document identifies the Security Target (ST), Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The Security Target document contains the following sections:

- Conformance Claims (Section 3)
- Security Problem Definition (Section 4)
- Security Objectives (Section 5)
- Extended Components Definition (Section 6)
- Security Requirements (Section 7)
- TOE Summary Specification (Section 8)
- Terminology, Acronyms and References (Section 9)

2.1 TOE reference

TOE Identification	TOE developer	TOE type	TOE HW version	SW/OS	Kernel version	Build
Bittium Tough Mobile C (BTMC)	Bittium	SD-42	Product code: 9304809A03	Android 5.1.1	3.4.0	S2_BSOS_1.1.5C_MR22_sapphire2

Table 1: TOE Identification

2.2 ST References

Title	Version	Author	Publication Date
Bittium Tough Mobile C Security Target Lite	1.0.1	Risto Vitikka	17.05.2019

Table 2: ST Identification

2.3 Product Overview

The device is a Dual OS mobile handset running with two separate operating systems. Operating systems are normal operating system which is regular Android OS and Bittium Secure OS Secure operating system which base on Android 5.1.1 operating system with Bittium modifications. Only one operating system can be active at the time. Switching between two operating systems is done using menu under power key.

Figure 1.Bittium Dual OS handset

2.4 TOE Overview

2.4.1 TOE type

The TOE is a mobile handset running with Bittium Secure OS Secure operating system which base on Android 5.1.1 operating system with Bittium modifications. Bittium Secure OS mode is focus of this Security Target documentation. Officials who are communicating sensitive information are intended to use device as reliable and secure mobile communication device for governmental.

2.4.2 Usage and major security features of the TOE

TOE 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 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.

TOE also protects user sensitive data, protects from unauthorized access and external tampering.

Major security features of the TOE are:

- Security Audit – collects and stores security information of the TOE's to the data partition.
- Cryptographic Support – provides cryptographic operations. These functionalities are used for data and communication protection and integrity verification.

- User data protection – performs user data communication channel protection using VPN. Protects user data and corresponding keys from unauthorized access and in case of security integrity violation.
- Identification and authentication – perform user authorization and identification and restrict usage accordance security policies.
- Security Management – allows user manage security settings of the TOE, download trusted applications and perform emergency procedures in case of security violation. Allows MDM to configure security features of the TOE via secure connection.
- Protection of the TOE Security Functions – TOE protection is performed during boot and run time. TOE protection is performed using secure boot and integrity checking. Tamper handling protects TOE from external attacks and from OS manipulating. Trusted SW update includes protection of FOTA update and application updates.
- TOE Access – protect device by handling use session with different protection mechanisms. Such mechanisms are like two factor authentication (i.e. user credentials and HOTP from NFC dongle to login), session restrictions and timings.
- Trusted path/channels – Provide trusted communication channel for IP traffic using TLS (between TOE and MDM server) and IPsec (for VPN gateway).

2.4.3 Required non-TOE hardware/software/firmware

The following figure shows the TOE scope.

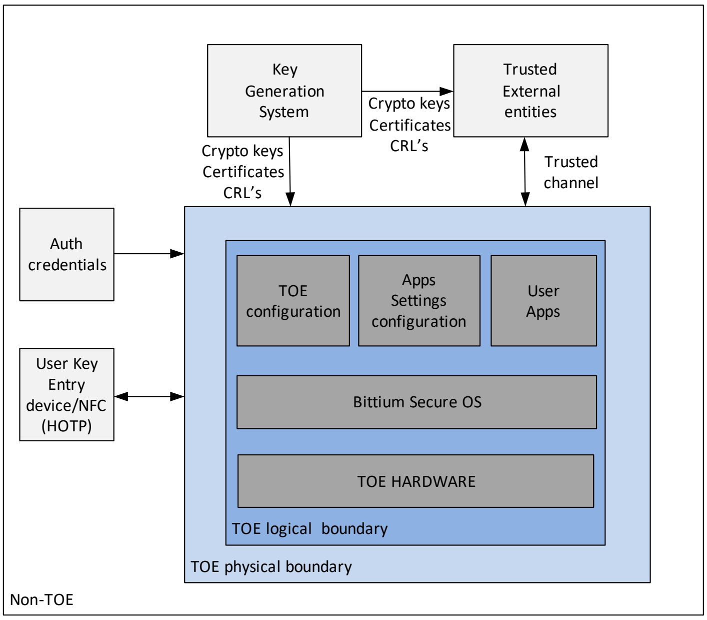


Figure 2: TOE boundary

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 (MDM, VPN, …);
- Key generation system providing the mobile device and trusted external entities (e.g. the VPN endpoint) with keys, certificates and CRLs;
- External devices used for user authentication (e.g. NFC-tags);
- Any other hardware or software that is not necessary for the secure operation of the TOE, such as NTP server and analysis tools used for collecting and analysing the audit records, app-library, update server, servers for Voice over IP and instant messaging, etc.
- Other networks devices belonging to the organisation to enforce policies for accessing other networks (firewalls).

2.4.4 Product Physical/Logical Features and Functionality not included in the TOE

Tough Mobile C product is Dual OS mobile handset running with two separate operating systems: Normal Android OS and Bittium Secure OS. Normal Android OS is out of the scope of the TOE and there is no assurance level associated with its functionality.

Tough Mobile C device records Audit Log information during both normal OS and Bittium Secure OS execution. TOE focus only for Audit functionality during Bittium Secure OS execution. Provisioning and configuration of TOE is done via MDM, but normal Android OS side handling is out of the scope of the TOE.

2.5 TOE description

The TOE is a mobile device running with Bittium Secure OS Secure operating system. SW version of evaluated TOE is S2_BSOS_1.1.5C_MR22_sapphire2. TOE HW environment is Bittium Tough Mobile C and Bittium Secure OS Secure operating system running in that HW environment.

2.5.1 Physical Scope

TOE provides wireless connectivity including secure VPN connectivity. TOE could be used as a mobile device within an enterprise environment where the configuration of the device is managed through a compliant MDM. MDM environment (i.e. Bittium Secure Suite and VPN Gateway) are out of the scope of the TOE, but it is used to support evaluation.

TOE is delivered in regular sales package including:

- Bittium Tough Mobile C HW and Bittium Secure OS Secure operating system
- Bittium Secure OS Secure operating system SW version is S2_BSOS_1.1.5C_MR22_sapphire2

- NFC dongle for two phase authentication
- Tough Mobile Quick start guide.

Note: NFC dongle and the Tough Mobile Quick start guide are part of the delivery but they are NOT part of the TOE on the common criteria evaluation.

TOE deployment needs to be performed before end user is able to take TOE in use. Common method is to do TOE delivery to enterprise author who does deployment and configuration before delivery to enterprise user.

Documentation for enterprise author is delivered accordance separate agreement. Common method is to do document delivery via email using pgp encrypting. Pgp key exchange between sender and received provide needed document security. Pgp encrypting use signature created with valid certificate and its proofs integrity of documents. Following documentation provide instructions for enterprise author how to perform initial provisioning and configuration of TOE via MDM, user and MDM administrator instructions of operations within enterprise environment:

- Tough Mobile C Preparative procedures.pdf (2.0.0)
- Tough Mobile C Operational User Guidance.pdf (4.0.0)

Note: The preparative procedures and the operation user guidance are part of the TOE on the common criteria evaluation.

2.5.2 Logical Scope

The logical boundary of the TOE includes all interfaces and functions within the physical boundary. The logical boundary of the TOE is described in chapters 7 Security Requirements and in chapter 8 TOE Summary Specification. The TOE includes the following functional sections:

- Security Audit
- Cryptographic support
- User data protection
- Identification and authentication
- Security management
- Protection of the TSF
- TOE access
- Trusted path/channels
- Security updates

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

- **Trusted channels**. Communication with external entities is 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.
    - 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.
- **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 the TOE verifies the HMAC SHA-256 based One-Time-Password is stored in an external device. In this case, the HMAC SHA-256 based One-Time-Password is stored in the NFC dongle. Successfully entered User credentials and HMAC SHA-256 based One-Time-Password are needed to unlock the TOE.
- **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. The VPN-tunnel 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 in case of HW or SW tampering.
- **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 and cryptographic token. 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 protected.** The enclosure of the TOE is protected with tamper-detection mechanism.
- **Audit.** The TOE records security relevant events and associate each event with the identity of the module 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 Bittium Secure Suite through a trusted channel.

2.5.3 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.

2.6 Overview of the operational environment

2.6.1 Identification

IT Identification	Developer	SW/OS	Kernel version
Bittium Secure Suite	Bittium SafeMove	OS: CentOS 7	3.10.0-514.10.2.el7.x86 64
		SecureSuite 9.2.100	
		Manager: 9.3.176	
SafeMoveCryptoIPSec VPN gateway	Bittium SafeMove	OS: CentOS 7	3.10.0-514.10.2.el7.x86 64
		Crypto IP: 4.10.141	

Table 3: Environment Identification

2.6.2 Bittium Secure Suite

Bittium Secure Suite is critical part of ensuring the security of the TOEs operational environment. It's responsible for providing a trusted channel between the TOEs other components in its operational environment and remote management capabilities that can be used to verify the integrity of the TOE and enforce policies related to TOEs various capabilities. TOE integrity and acceptable policies are enforced utilizing Bittium Secure Suite MDM service that provides centralized means to manage a fleet of TOE devices deployed in an organization.

2.6.3 SafeMoveCryptoIPVPN gateway

Trusted communication channel is provided using encrypted IPsec VPN tunnel with PKI certificate-based authentication between the TOE and its operational environment by utilizing SafeMoveCryptoIPVPN gateway as the endpoint. It supports relevant cryptographic algorithms and is compatible for example with NSA Suite B.

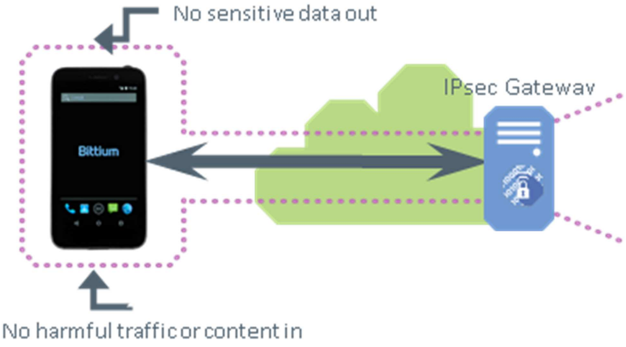


Figure 3 SafeMoveCryptoIPVPN Gateway

3 CONFORMANCE CLAIM

3.1 CC Conformance Claims

This Security Target 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.

This Security Target is conformant to Evaluation Assurance Level EAL2.

4 SECURITY PROBLEM DEFINITION

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

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 and MDM administrator are assumed to be trustworthy and trained, we cannot exclude that mistakes are being made.

4.1 Threats

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

4.1.1 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.INSTALLED_APPLICATIONS

Mechanisms of the TOE to check whether an application is authorized to be installed or not. Dimensions to be protected: integrity.

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.

4.1.2 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.

4.1.3 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**

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.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.INSTALLED_APPLICATIONS**, **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.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.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**

4.2 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 applications authorised to communicate outside of the tunnel are configured in the VPN-policy by using protocol, source and destination addresses and ports.

P.AUDIT

The TOE must record security relevant events and associate each event with the identity of the module 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.RBG

The TOE must implement random bit generators meeting the requirements of strength a minimum of 128 bits of entropy and quality metrics specified in NIST's SP800-90B.

4.3 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 (i.e. administrator and users), are trustworthy and they have been trained sufficiently to carry out these security management tasks in a proficient manner.

It is assumed that it’s not possible to access malicious WiFi Access Point and it is also assumed that no malicious files can be introduced or will be stored in the TOE.

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 do not reveal sensitive user data on the screen lock without user authentication.

5 SECURITY OBJECTIVES

5.1 Security Objective for the Target of Evaluation (TOE)

The TOE is intended to protect the asset, of High Security Level information, in accordance with the following objectives:

Objective	Description
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. - 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. 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 KEK for the disk decryption is provided by the user. User credentials and HOTP from NFC is needed to produce correct KEK for disk encrypting. The TOE shall verify HOTP for verification is stored in an external device. In this case the HOTP is stored in the NFC dongle. Fail to the verification shall cause TOE to

	remain locked.
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 encrypted DEK and parameters required by KEK derivation algorithm. Emergency situations are the following: - defined number of user PIN authentication attempts-during start-up. - positive integer number of consecutive failed PIN or KEY-admin authentication attempts; - the detection of an integrity violation in specific cases
O.INTEGRITY	The TOE shall be able to verify the integrity of: - the OS during power-up - VPN modules An integrity failure shall cause an immediate emergency zeroization.
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.
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 module that caused the event. The TOE must prevent unauthorized modification of the audit trail and prevent loss of audit trail data. 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-detection mechanism.
O.RNG	The TOE must implement random number generators meeting the requirements of strength a minimum of 128 bits of entropy and quality metrics specified in NIST's SP800-90B.

Table 4: Secure Objectives for the TOE

5.2 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.

Objective	Description
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. The administrator along with the regular user of the TOE will implement the necessary measures in order to get confidence that the files introduced in the TOE are not malicious. The administrator will provide trusted configuration which does not allow malicious Wi-Fi Access Point usage by TOE.
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.

Table 5: Secure objects for the Operational Environment

5.3 Security Objectives rationale

5.3.1 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		
O.INSTALLATION	X												
O.SECURE_BOOT						X	X			X			
O.ERASURE		X	X										
O.INTEGRITY		X		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 6: Threats to security objectives

	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 7: OSPs and Assumptions to Security objectives

5.3.2 Security Objectives Sufficiency

5.3.2.1 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 integrity before establishing a VPN connection (O.INTEGRITY).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 NFC is entered when required from an external device configured for that purpose (O.SECURE_BOOT).

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.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.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 and the audit log traffic 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.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).

T.HW_TAMPER

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

5.3.3 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 firewall. The applications is not authorised to communicate outside of the tunnel in the VPN-policy by using protocol, source and destination addresses and ports.

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.

5.3.3.1 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.

6 EXTENDED COMPONENTS DEFINITION

This chapter will introduce extended requirements which based on components in CC Part 2 or CC Part 3.

6.1 Extended Security Functional Requirements

This chapter lists and define extended security functional requirement. Requirements are included to the Chapter 7 Security requirements.

Class	Identifier	Description
Security Audit (FAU)	FAU_AUD_EXT.1	Extended: Security Audit Data generation
Cryptographic Support (FCS)	FCS_RBG_EXT.1	Extended: Random Number Generation
User data protection (FDP)	FDP_DSK_EXT.1	Extended : Protection of Data on Disk
	FDP_ZER_EXT.1	Extended: Zeroization
Protection of the TOE Security Functions (FPT)	FPT_SBT_EXT.1	Extended: Secure Boot and Operation continuity
	FPT_TUD_EXT.1	Extended: Trusted Update
	FPT_TST_EXT.2	Extended: TSF Integrity Testing
	FPT_PHY_EXT.1	Extended: Detection of physical attack

Table 8: Extended Security Functional Requirements

This extended components definition is included as per [CC31p1]. Extended components may be based on existing CC requirements. In this ST, 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 Chapter 7 Security requirements of this ST.

- The extended requirement FAU_AUD_EXT.1 is defined to specify requirements for recording of security relevant events
- The extended requirement FCS_RBG_EXT.1 is defined to specify requirements for the Random Bit Generation to be used for secrets generation or any operation requiring randomization.
- The extended requirement FDP_DSK_EXT.1 for stored data encryption is used to specify the transparent encryption performed on a mobile device.
- The extended requirement FDP_ZER_EXT.1 for zeroization is used to specify the ability of local and remote secure erase.
- The extended requirement FPT_SBT_EXT.1 is defined to specify the secure boot and the operation continuity processes.
- The extended requirement FPT_TUD_EXT.1 for trusted updates is used to specify requirements for automatic trusted updates.

- The extended requirement FPT_TST_EXT.2 is defined to extend the integrity testing requirements satisfying the O.INTEGRITY security objectives.
- The extended requirement FPT_PHY_EXT.1 is defined to specify requirements for detection of physical attack

6.2 FAU_AUD_EXT – Extended: Security Audit Data Generation

Family Behavior

This family defines requirements for recording the occurrence of security relevant events that take place under TSF control. This family identifies the level of auditing, enumerates the types of events that shall be auditable by the TSF, and identifies the minimum set of audit-related information that should be provided within various audit record types.

Component Levelling

FAU_AUD_EXT.1 Audit data generation defines the level of auditable events, and specifies the list of data that shall be recorded in each record.

Management

There are no management activities foreseen.

Audit

There are no events defined to be auditable.

6.2.1 FAU_AUD_EXT – Extended: Audit Data Generation

Hierarchical to: No other components.
Dependencies: FPT_STM.1 Reliable time stamps

FAU_AUD_EXT.1.1

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

a) All auditable events for the [selection, choose one of: *minimum, basic, detailed, not specified*] level of audit; and

b) [assignment: *other specifically defined auditable events*].

FAU_AUD_EXT.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*].

6.3 FCS_RBG_EXT – Extended: Random Bit Generation

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

Family behaviour

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

Component Levelling

FCS_RBG_EXT.1 is not hierarchical.

Management

There are no management activities foreseen.

Audit

There are no events defined to be auditable.

6.3.1 FCS_RBG_EXT.1 – Extended: Random Bit Generation

Hierarchical to: none

Dependencies: none

FCS_RBG_EXT.1.1

The deterministic RBG used by TSF shall provide a minimum of [selection: 128 bits, 256 bits] of entropy that meet [assignment: a defined quality metric].

6.4 FDP_DSK_EXT – Extended: Protection of Stored Data

Family behaviour

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

Component Levelling

FDP_DSK_EXT.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_AUD_EXT.1 Security audit data generation is included in the ST:

a) Two factor authentication login event.

6.4.1 FDP_DSK_EXT.1 – Extended: Protection of Stored Data

Hierarchical to: none

Dependencies: FCS_COP.1 Cryptographic operation

FDP_DSK_EXT.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_EXT.1.2

The [assignment: cryptographic *key*] shall be encrypted with a [assignment: cryptographic *key*]. Non-volatile decrypted DEK is located only within hardware encryption engine.

FDP_DSK_EXT.1.3

The TSF shall encrypt user data partition without user intervention

FDP_DSK_EXT.1.4

The TSF shall be able to decrypt the protected data once the corresponding [assignment: cryptographic *key*] is presented.

6.5 FDP_ZER_EXT – Extended: Zeroization

Family behaviour

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 Levelling

FDP_ZER_EXT.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_AUD_EXT.1 Security audit data generation is included in the ST:

a) There are no events defined to be auditable.

6.5.1 FDP_ZER_EXT.1 –Extended: Zeroization

Hierarchical to: none

Dependencies: FMT_SMF.1 specification of management Functions

FDP_ZER_EXT.1.1

The TSF shall be able to securely erase following Critical Security Parameters and classified and personal data in case of emergency erase by user or remote management or when maximum PIN or KEY-admin authentication attempts are exceeded or user PIN authentication attempts-during start-up are exceeded:

- Encrypted DEK
- Stored security policies
- All application data stored in user data partition
- MDM certificates

FDP_ZER_EXT.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 user PIN authentication attempts-during start-up (configured by the authorised user);

b) [*assignment: positive integer number*] of consecutive failed PIN or KEY-admin authentication attempts;

c) the detection of an integrity violation in specific cases

FDP_ZER_EXT.1.3

The TSF shall be able to perform securely erasure starting with the zeroization of the partition which contains the encrypted DEK.

6.6 FPT_SBT_EXT – Extended: Secure Boot and Operation continuity

Family behaviour

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

Component Levelling

FPT_SBT_EXT.1 is not hierarchical.

Management

There are no management activities foreseen.

Audit

There are no events defined to be auditable.

6.6.1 FPT_SBT_EXT.1 – Secure Boot and Operation continuity

Hierarchical to: none

Dependencies: FMT_SMF.1 specification of management Functions.

FPT_SBT_EXT.1.1

For the secure boot to be guaranteed:

a) the TOE shall be able to obtain the proper [*assignment: cryptographic key*] for the DEK decryption AND

b) the TOE shall verify the integrity of the OS.

FPT_SBT_EXT.1.2

For the operation continuity to be guaranteed:

a) the TOE shall be able to obtain the proper [*assignment: cryptographic key*] stored in an external device for unlocking the TOE.

b) the TOE shall enter in a blocked status if the verification of [*assignment: cryptographic key*] fails.

6.7 FPT_TUD_EXT – Extended: Trusted Updates

Family behaviour

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 Levelling

FPT_TUD_EXT.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_AUD_EXT.1 Security audit data generation is included in the ST:

- administrative actions of TOE configuration for Software update.

6.7.1 FPT_TUD_EXT.1 Extended Trusted Update

Hierarchical to: none

Dependencies: FCS_CKM.1 Cryptographic key generation and FCS_COP.1 Cryptographic operation

FPT_TUD_EXT.1.1

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

FPT_TUD_EXT.1.2

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

FPT_TUD_EXT.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_EXT.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.

6.8 FPT_TST_EXT – Extended integrity 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 integrity test requirements. The new component extends the integrity testing to require:

- the verification of the integrity of:
  - o the OS during power-up;

  - VPN modules

- the verification that only authorised Apps will be imported and installed;
- providing the capability to verify the integrity of data in key store;
- defining response actions to be executed in case of integrity test fails.

Family behaviour

In addition to what is expressed in [CC31p2]:

The family defines the requirements for the integrity testing of the TSF with respect to some expected correct operation. These tests can be carried out at start-up or when other conditions are met. The actions to be taken by the TOE as the result of integrity 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_EXT.2 extends the integrity testing and defines actions to be performed in case of a fail.

Management:

There are no management activities foreseen.

Audit:

The following actions should be auditable if FAU_AUD_EXT.1 Security audit data generation is included in the ST:

a) There are no events defined to be auditable.

6.8.1 FPT_TST_EXT.2 Extended Integrity Test

Hierarchical to: none

Dependencies:

- FPT_FLS.1 Failure with preservation of secure state.

FPT_TST_EXT.2.1

The TSF shall verify the integrity of

- the OS during power-up AND

- VPN modules

FPT_TST_EXT.2.2

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

FPT_TST_EXT.2.3

The TSF shall provide the capability to verify the integrity of [*selection, choose one of: VPN configuration, data in keystore, all TSF data*].

FPT_TST_EXT.2.4

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

FPT_TST_EXT.2.5

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

- An integrity failure shall cause an immediate emergency zeroization in case of HW or SW tampering
- A VPN-tunnel setup failure shall cause all IP-traffic selected for its transmission through the tunnel to be blocked.

6.9 FPT_PHY_EXT– Extended: Physical protection

Family Behavior

TSF physical protection components refer to restrictions on unauthorized physical access to the TSF, and to the deterrence of, and resistance to, unauthorized physical modification, or substitution of the TSF.

The requirements of components in this family ensure that the TSF is protected from physical tampering and interference. Satisfying the requirements of these components results in the TSF being packaged and used in such a manner that physical tampering is detectable, or resistance to physical tampering is enforced. Without these components, the protection functions of a TSF lose their effectiveness in environments where physical damage cannot be prevented. This family also provides requirements regarding how the TSF shall respond to physical tampering attempts.

Component Levelling

FPT_PHY_EXT.1 detection of physical attack provides for features that indicate when a TSF device or TSF element is subject to tampering. Determining if tampering has occurred and notification of tampering is automatic.

Management

The following actions could be considered for the management functions in

FMT:
a) management of the user or role that determines whether physical tampering has occurred.

Audit

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

a) Backup battery depletion event.

6.9.1 FPT_PHY_EXT.1 Passive detection of physical attack

Hierarchical to: No other components.
Dependencies: No dependencies.

FPT_PHY_EXT.1.1

In case the battery power is not exhausted, the TSF shall provide unambiguous detection of physical tampering that might compromise the TSF for following tampers:

- Frame tamper
- Serpentine tamper

FPT_PHY_EXT.1.2

In case the physical tampering has been detected, the TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.

6.10 Extended Security Assurance Requirement Components

Security Assurance Requirement Components based on components in CC Part 3.

7 SECURITY REQUIREMENTS

The Security Functional Requirements included in this section are derived from [CC31p2], with additional extended functional components. This chapter defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the Target of Evaluation (TOE) and to scope the evaluation effort.

7.1 Conventions

There are four types of operations, when performed on functional requirements that derive from CC Part 2, are identified in this ST in the following manner:

- Selection and assignment: Indicated by are identified by using **bold**
- Refinement: Refined components are identified by using **bold** for additional information, or ~~**strikeout**~~ for deleted text.
- Iteration: Indicated by appending a number in to the end of the functional component e.g., ‘FCM_CMK.1(1), FCM_CMK.1(2)’ and so on or with /CONTEXT.

7.2 TOE Security Functional Requirements

This section describes the Security Functional Requirements for the TOE. The following table identifies the SFRs that are satisfied by the TM C.

Class	Identifier	Description
Security Audit(FAU)	FAU_AUD_EXT.1	Audit data generation
	FAU_STG.1	Protected audit trail storage
	FAU_STG.4	Prevention of audit data loss
Cryptographic Support (FCS)	FCS_CKM.1	Cryptographic Key Generation
	FCS_COP.1	Cryptographic Operation
	FCS_RBG_EXT.1	Extended – Random Bit Generation
User data protection (FDP)	FDP_IFC.2	Complete information flow control
	FDP_IFF.1	Simple security attributes
	FDP_DSK_EXT.1	Extended – Protection of Data on Disk
	FDP_ZER_EXT.1	Extended – Zeroization
Identification and authentication (FIA)	FIA_UAU.2	User Authentication before any action
	FIA_AFL.1	Authentication failure handling
Security Management(FMT)	FMT_SMF.1	Specification of management functions

	FMT_SMR.1	Security management roles
	FMT_MSA.1	Management of security attributes
	FMT_MSA.3	Static attribute initialisation
Protection of the TOE Security Functions (FPT)	FPT_FLS.1	Failure with preservation of secure state
	FPT_SBT_EXT.1	Extended – Secure Boot and Operation continuity
	FPT_STM.1	Reliable time stamps
	FPT_TST_EXT.2	Extended – Integrity Test
	FPT_TUD_EXT.1	Extended – Trusted Update
	FPT_PHY_EXT.1	Extended – Passive detection of physical attack
TOE Access (FTA)	FTA_SSL.1	TSF-initiated session locking
	FTA_SSL.2	User-initiated locking
Trusted path/channels (FTP)	FTP_ITC.1/VPN-tunnel	Inter-TSF Trusted Channel (Application communications)
	FTP_ITC.1/REM-ADM	Inter-TSF Trusted Channel (remote administration)
	FTP_ITC.1/AUDIT	Inter-TSF Trusted Channel

Table 9: Security Functional Requirements

7.2.1 Security Audit (FAU)

7.2.1.1 Extended - Audit data generation (FAU_AUD_EXT.1)

FAU_AUD_EXT.1.1

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

- All auditable events for the **not specified** level of audit
- **Other specifically defined auditable events:**
  1. **Two factor authentication login event**
  2. **Changes / modifications in the administrative security policies enforced on the TOE**
  3. **Other events**
     - **Backup battery depletion event**
     - **Failed screen lock code entry attempts**

- **Application error messages**

FAU_AUD_EXT.1.2

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

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

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

7.2.1.2 Protected audit trail storage (FAU_STG.1)

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 **prevent** unauthorised modifications to the stored audit records in the audit trail.

7.2.1.3 Prevention of audit data loss (FAU_STG.4)

FAU_STG.4.1

The TSF shall **overwrite the oldest stored audit records** and **no other actions to be taken in case of audit storage failure** if the audit trail is full.

7.2.2 Cryptographic support (FCS)

7.2.2.1 Cryptographic Key Management (FCS_CKM.1(1))

FCS_CKM.1.1(1) – Cryptographic Key Generation (RSA)

The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **RSA** and specified cryptographic key sizes **2048, 3072 and 4096** that meet the following:
None

7.2.2.2 Cryptographic Key Management (FCS_CKM.1(2))

FCS_CKM.1.1(2) – Cryptographic Key Generation (ECC)

The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **ECC** and specified cryptographic key sizes **NIST curves P-256, P-384 and P-521** that meet the following: **FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4**.

7.2.2.3 Cryptographic Operation (FCS_COP.1(1))

FCS_COP.1.1(1) Cryptographic Operation (AES)

The TSF shall perform **encryption and decryption** in accordance with a specified cryptographic algorithm **AES-256 in CBC, GCM and XTS** mode and cryptographic key size **256-bits** that meets the following: **FIPS PUB 197, NIST SP 800-38A (CBC mode), NIST SP 800-38D (GCM mode) and NIST SP 800-38E (XTS mode)**.

7.2.2.4 Cryptographic Operation (FCS_COP.1(2))

FCS_COP.1.1(2) Cryptographic Operation (HASH)

The TSF shall perform **cryptographic hashing services** in accordance with a specified cryptographic algorithm: **SHA-1 SHA-256, SHA-384, SHA-512** and ~~**cryptographic key**~~ message digest sizes **160, 256, 384, 512** that meet following: **FIPS Pub 180-4**

7.2.2.5 Cryptographic Operation (FCS_COP.1(3))

FCS_COP.1.1(3) Cryptographic Operation (RSA Signature)

The TSF shall perform **cryptographic signature services (generation and verification)** in accordance with specified cryptographic algorithms: **RSA** with key size **2048, 3072, 4096** that meet following: **none**

7.2.2.6 Cryptographic Operation (FCS_COP.1(4))

FCS_COP.1.1(4) Cryptographic Operation (ECDSA Signature)

The TSF shall perform **cryptographic signature services (generation and verification)** in accordance with specified cryptographic algorithms: **ECDSA** with key size **NIST curves P-256, P-384 and P-521** that meet following: **FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6.**

7.2.2.7 Cryptographic Operation (FCS_COP.1(5))

FCS_COP.1.1(5) Cryptographic Operation (Keyed Hash)

The TSF shall perform **keyed-hash message authentication** in accordance with a specified cryptographic algorithm: **HMAC-SHA-1, HMAC-SHA-256,HMAC-SHA-384 , HMAC-SHA-512** and cryptographic key sizes **160, 256, 384, 512** that meets following **FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code, and FIPS Pub 180-4" Secure Hash Standard.**

7.2.2.8 Extended – Random Bit Generation (FCS_RBG_EXT.1)

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

FCS_RBG_EXT.1.1

The deterministic RBG used by TSF shall provide a minimum of **128 bits** of entropy that meet **non-IID min-entropy of 0.7 according to NIST's SP800-90B EntropyAssessment test tool**.

Application Note

The kernel's character special files /dev/random and /dev/urandom provide an interface to the entropy source.

7.2.3 User Data Protection (FDP)

7.2.3.1 Complete information flow control (FDP_IFC.2)

FDP_IFC.2.1

The TSF shall enforce the **TRAFFIC SFP** on the following subjects: **external entities sending IP data to the TOE and the TOE sending IP data to external entities**, **information IP data sent to or sent by the TOE** 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 Wi-Fi network interfaces) to ensure that all IP data traffic is always sent and/or received using the VPN tunnel.

7.2.3.2 Simple security attributes (FDP_IFF.1)

FDP_IFF.1.1

The TSF shall enforce the **TRAFFIC SFP** based on the following types of subject and information security attributes:

Subjects: external entities sending data to TOE and/or TOE sending data to external entities;

Subject security attributes: configuration defining restrictions for source and/or destination address, source and/or destination port for inbound/outbound IP data traffic passed into or received from VPN tunnel;

Information: IP data sent to or sent by TOE;

Information security attributes~~:~~IP protocol version, protocol type, source address of the subject and/or destination address of the subject, source port of the subject and/or destination port of the subject.

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: **all configured information security attribute values match and the rule type is set to pass.**

The security attributes can consist of one or more of the following:

- **IP protocol version**
- **Protocol type**
- **Source IP address or IP address range**
- **Destination IP address or IP address range**
- **Source port or port range**
- **Destination port or port range**

FDP_IFF.1.3

The TSF shall enforce the **no additional information flow control SFP rules**.

FDP_IFF.1.4

The TSF shall explicitly authorise an information flow based on the following rules: **incoming IP data packet matching configured information security attribute values is decrypted and read from VPN tunnel, outgoing IP data packet matching configured information security attribute values is encrypted and passed into VPN tunnel.**

FDP_IFF.1.5

The TSF shall explicitly deny an information flow based on the following rules: **incoming/outgoing IP data packet passing through the crypto interface without a matching information security attribute configuration is dropped, except for the following packets:**

- **Protocol packets part of establishing a VPN tunnel**

7.2.3.3 Extended – Protection of Data on Disk (FDP_DSK.1)

FDP_DSK_EXT.1.1

The TSF shall perform encryption of **user data partition** in accordance with **AES-XTS-256**, such that no such data is otherwise stored as plain text within the TOE.

FDP_DSK_EXT.1.2

The **DEK** shall be encrypted with a **KEK**. Non-volatile decrypted DEK is located only within hardware encryption engine.

FDP_DSK_EXT.1.3

The TSF shall encrypt user data partition without user intervention.

FDP_DSK_EXT.1.4

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

7.2.3.4 Extended – Zeroization (FDP_ZER_EXT.1)

FDP_ZER_EXT.1.1

The TSF shall be able to securely erase following Critical Security Parameters and classified and personal data in case of emergency erase by user or remote management or when maximum PIN or KEY-admin authentication attempts are exceeded or user PIN authentication attempts-during start-up are exceeded:

- Encrypted DEK
- Stored security policies
- All application data stored in userdata partition
- MDM certificates

<u>Application Note:</u>

DM-verity, Serpentine or Frame tamper will erase Encrypted DEK, which is used to decrypt other Critical Security Parameters, but not: stored security policies, all application data stored in user data partition or MDM certificates.

FDP_ZER_EXT.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 user PIN authentication attempts-during start-up (configured by the authorised user);
b) **number configured by the administrator in FIA_AFL.1** of consecutive failed PIN or KEY-admin authentications attempts;
c) the detection of an integrity violation in specific cases

Application note:

Integrity violation specific cases are a) SW or/and HW tamper detection and b) user or the administrator initiates an erasure or when maximum PIN or KEY-admin authentication attempts are exceeded or user PIN authentication attempts-during start-up are exceeded.

FDP_ZER_EXT.1.3

The TSF shall be able to perform securely erasure starting with the zeroization of the partition which contains the encrypted DEK.

7.2.4 Identification and authentication (FIA)

7.2.4.1 User Authentication before any action (FIA_UAU.2)

FIA_UAU.2.1

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

7.2.4.2 Authentication failure handling (FIA_AFL.1)

FIA_AFL.1.1

The TSF shall detect when **an administrator configurable positive integer within 3 to 20** unsuccessful authentication attempts occur related to*:*

- **user PIN authentication on a running system OR**
- **user PIN authentication-during start-up**.

FIA_AFL.1.2

When the defined number of unsuccessful authentication attempts has been **met**, the TSF shall **perform emergency erase of the data stored on the handset**.

7.2.5 Security Management (FMT)

7.2.5.1 Specification of management functions (FMT_SMF.1)

FMT_SMF.1.1

The TSF shall be capable of performing the following management functions*:*

- **Change password used in two factor authentication;**
- **Installation of apps from the local app market;**

- **Emergency erase;**
- **Change the following security policies enforced to the TOE such as**
  - **User password quality**
  - **Maximum login attempts before zeroization**
  - **Disabling/enabling camera**
- **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);**

Application Note

Active/currently used password quality is not changed when user password quality policy is updated by MDM. User password quality security policy effect only to new/next password.

7.2.5.2 Security management roles (FMT_SMR.1)

FMT_SMR.1.1

The TSF shall maintain the roles **user and administrator**.

7.2.5.3 Management of security attributes (FMT_MSA.1)

FMT_MSA.1.1

The TSF shall enforce the **TRAFFIC SFP** to restrict the ability to **change default, modify and delete** the security attributes **belonging to the TRAFFIC SFP** to **administrator**.

7.2.5.4 Static attribute initialisation (FMT_MSA.3)

FMT_MSA.3.1

The TSF shall enforce the **TRAFFIC SFP** to provide **permissive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2

The TSF shall allow the **administrator** to specify alternative initial values to override the default values when an object or information is created.

7.2.6 Protection of TOE Security Functions (FPT)

7.2.6.1 Failure with preservation of secure state (FPT_FLS.1)

FPT_FLS.1.1

The TSF shall preserve a secure state when the following types of failures occur:

- **failed boot image integrity check during power-up**
- **failed system applications integrity check during runtime**
- **hardware tampering is detected**

Application Note
Secure states and integrity checks are presented in chapter 8.6.

7.2.6.2 Secure Boot and Operation continuity (FPT_SBT_EXT.1)

FPT_SBT_EXT.1.1

For the secure boot to be guaranteed:

- the TOE shall be able to obtain the proper **KEK** for the DEK decryption AND
- the TOE shall verify the integrity of the OS.

Application Note
Secure boot and integrity verification is presented in chapter 8.6.

FPT_SBT_EXT.1.2
For the operation continuity to be guaranteed:

- the TOE shall be able to obtain the proper **HMAC SHA-256 based One-Time-Password** stored in an external device for unlocking the TOE.
- the TOE shall enter in a blocked status if the verifications of the **HMAC SHA-256 based One-Time-Password** fails.

7.2.6.3 Reliable time stamps (FPT_STM.1)

FPT_STM.1.1

The TSF shall be able to provide reliable time stamps.

7.2.6.4 Extended Integrity (FPT_TST_EXT.2)

FPT_TST_EXT.2.1

The TSF shall verify the integrity of

- the OS during power-up AND
- VPN modules

Application Note
Integrity verification is presented in chapter 8.6.

FPT_TST_EXT.2.2

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

FPT_TST_EXT.2.3

The TSF shall provide the capability to verify the integrity of **data in key store**.

FPT_TST_EXT.2.4

The TSF shall provide the capability to verify the integrity of **OS**.

Application Note
Integrity verification is presented in chapter 8.6.

FPT_TST_EXT.2.5

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

- An integrity failure shall cause an immediate emergency zeroization in case of HW or SW tampering
- A VPN-tunnel setup failure shall cause all IP-traffic selected for its transmission through the tunnel to be blocked

Application Note
Integrity failures related to HW and SW tampering is presented in chapter 8.6.

7.2.6.5 Extended – Trusted Update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1.1

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

FPT_TUD_EXT.1.2

The TSF shall provide a mechanism that on a regular basis checks update to TOE software.

FPT_TUD_EXT.1.3

The TSF shall provide a means to verify software updates to the TOE using a **digital signature mechanism** prior to installing those updates.

FPT_TUD_EXT.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.

7.2.6.6 Extended - Passive detection of physical attack (FPT_PHY_EXT.1)

FPT_PHY_EXT.1.1

In case the battery power is not exhausted, the TSF shall provide unambiguous detection of physical tampering that might compromise the TSF for following tampers:

- Frame tamper
- Serpentine tamper

FPT_PHY_EXT.1.2

In case the physical tampering has been detected, the TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.

Application Note
In case the battery power is exhausted, the TSF provide the capability to determine that battery related physical tamper has occurred. This battery related tamper is handled as a warning and does not cause emergency erase procedure. Battery related tamper requires user interaction to continue operation.

7.2.7 TOE Access (FTA)

7.2.7.1 TFS initiated session locking (FTA_SSL.1)

FTA_SSL.1.1

The TSF shall lock an interactive session after **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: **user authentication**

7.2.7.2 User-initiated locking (FTA_SSL.2)

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*:* **user authentication**

7.2.8 Trusted Path/Channels (FTP)

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

FTP_ITC.1.1(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(1)(VPN-tunnel)

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

FTP_ITC.1.3(1) (VPN-tunnel)

The TSF shall initiate communication via the trusted channel for **allowing IP network traffic communications between external entities (different to the VPN endpoint) and the TOE.**

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

FTP_ITC.1.1(2) (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(2)(REM-ADM)

The TSF shall permit ~~another~~ remote **trusted IT product** to initiate communication via the trusted channel.

FTP_ITC.1.3(2) (REM-ADM)

The TSF shall initiate communication via the trusted channel for **remote administration**.

Application Note
TLS used for MDM traffic inside VPN tunnel. This channel is the trusted channel established for the remote administration. These cryptographic algorithms shall provide, at least, 128 bits of security.

7.2.8.3 Inter-TSF Trusted Channel (FTP_ITC.1/Audit)

FTP_ITC.1.1(3) (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(3) (AUDIT)

The TSF shall permit **~~another~~ remote trusted IT product** to initiate communication via the trusted channel.

FTP_ITC.1.3(3) (AUDIT)

The TSF shall initiate communication via the trusted channel for **sending audit logs to an external entity**.

Application note
TLS protocol used to establish secure channel for audit functionality. This channel is the trusted channel established for sending the audit log to an external entity. These cryptographic algorithms shall provide, at least, 128 bits of security.

7.3 Rationale for Security Functional Requirements

7.3.1 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.OS_UPDATE	O.AUTHENTICATION	O.ADMIN	O.SECURITY_POLICIES	O.CRYPT_PROTECTION	O.SECURITY_DATA	O.AUDIT	O.HW_TAMPER	O.RNG
FAU_AUD_EXT.1												X		
FAU_STG.1												X		
FAU_STG.4												X		
FCS_RBG_EXT.1	X						X			X				X
FCS_CKM.1(1)	X	X												
FCS_CKM.1(2)	X	X												
FCS_COP.1(1)	X		X							X				
FCS_COP.1(2)	X	X	X		X					X				
FCS_COP.1(3)	X	X					X							
FCS_COP.1(4)	X	X				X								
FCS_COP.1(5)			X				X							
FDP_IFC.2	X													
FDP_IFF.1	X													
FDP_DSK_EXT.1			X				X			X				
FDP_ZER_EXT.1				X						X				
FIA_UAU.2			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				
FPT_SBT_EXT.1			X											
FPT_STM.1	X						X					X		
FPT_TST_EXT.2		X	X	X	X					X				
FPT_TUD_EXT.1						X								
FPT_PHY_EXT.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		

Table 10: Mapping of SFRs to the security objectives of the TOE

7.3.2 Sufficiency

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

O.TUNNEL

The TOE implements the following trusted channels: VPN-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.
- FPT_STM.1 provides the time for certificate verification.

The VPN tunnel supports RSA (FCS_COP.1(3), FCS_CKM.1(1)) and ECDSA (FCS_COP.1(4), FCS_CKM.1(2)) digital signature algorithms for authentication, AES-CBC (FCS_COP.1(1) for encryption, SHA-256, SHA-384, SHA-512 (FCS_COP.1(2)) for hashing and RSA (FCS_CKM.1(1)) and ECC (FCS_CKM.1(2)) for key generation during key exchange and AES-GCM (FCS_COP.1(1)) for IP traffic encryption. Random bit generation is done according FCS_RBG_EXT.1.

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_EXT.2 – FPT_TST_EXT.2.2 requires the TOE to verify that the app is authorised to be imported and installed.

Authorized applications certificate is verified using SHA-256 (FCS_COP.1(2)) hash value. The authorized application content is verified through the certificate and jar file signing that supports RSA (FCS_COP.1(3), FCS_CKM.1(1)), ECDSA (FCS_COP.1(4), FCS_CKM.1(2)) and SHA (FCS_COP.1(2)).

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 using HMAC SHA-256 based One-Time-Password stored in an external device is used to authenticate user utilizing FCS_COP.1(5)) processes.

Boot of the TOE only successes if the integrity of the OS is guaranteed and the proper KEK for the disk decryption are provided by the user or obtained by derivation from the user's credentials input (FPT_TST_EXT.2, FIA_UAU.2 FPT_SBT_EXT.1, FDP_DSK_EXT.1). Cryptographic Hashing (FCS_COP.1(2)) is used for the integrity monitoring during secure boot. Encrypting and decrypting of user data partition is done using AES256 (XTS) accordance (FCS_COP.1(1)).

For the operation continuity the TOE shall verify HOTP for verification is stored in an external device. In this case the HOTP stored in the NFC dongle. Fail to this verification shall cause the TOE to be blocked.

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_EXT.1. This process is to be implemented starting with the zeroization of the encrypted DEK and parameters required by KEK derivation algorithm. Emergency situations described are:

- defined number of user PIN authentication attempts-during start-up.
- defined number of consecutive failed authentications attempts (as defined in FIA_AFL.1);
- the detection of an integrity violation in specific cases (as defined in FDP_ZER_EXT.1.2)

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_EXT.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;
- VPN modules

The TOE shall be able to verify that only authorised applications can be installed via App-Library or build in into the FOTA image. These scenarios along with the actions in case of integrity violation are specified in FPT_TST_EXT.2 and FPT_FLS.1. The verified firmware contains the information to enable runtime integrity check of system applications and therefore the integrity of the OS is verified (as stated in FPT_SBT_EXT.1.1 and FPT_TST_EXT.2.1).

Integrity of OS during power-up and VPN modules is verified using SHA (FCS_COP.1(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_EXT.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)

- Local authentication is performed during start up 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_EXT.1). In addition, periodically, the HMAC SHA-256 based One-Time-Password is entered to the TOE from an external device (i.e. NFC dongle) to authenticate user.

- 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.

- A 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. MDM authenticates itself through certificates.

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.

KEK is derived using keyed hash (FCS_COP.1(5)) and RSA signature (FCS_COP.1(3)). One-time password generation is utilizing keyed hash (FCS_COP.1(5)). Respective random bit generation is done according FCS_RBG_EXT.1.

O.ADMIN

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

FIA_UAU.2, 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 will ensure that the administrator is properly authenticated by authenticating with the MDM 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_EXT.1, FDP_ZER_EXT.1) contributes protecting the cryptographic material.

FIA_UAU.2 will ensure that the administrator is properly authenticated by presenting the administrator certificate before being allowed to make the changes.

FMT_SMF.1 will ensure that the administrator and only the administrator are able to make changes to the security policies, including the cryptographic properties of the TOE.

FPT_TST_EXT.2 and FPT_FLS.1 will ensure the integrity of the TSF before the VPN tunnel is established protecting this way the associated cryptographic material.

Respective integrity is verified using SHA (FCS_COP.1(2)) and encryption is done using (FCS_COP.1(1)). Respective random bit generation is done according FCS_RBG_EXT.1.

O.SECURITY_DATA

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

FIA_UAU.2 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 privileges to 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 module that caused the event (FAU_AUD_EXT.1, FPT_STM.1).

The TOE must prevent unauthorized modification of the audit trail, prevent loss of audit trail data (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-detection mechanism (FPT_PHY_EXT.1).

O.RNG

The TOE must implement random number generators meeting the requirements of strength a minimum of 128 bits of entropy and quality metrics specified in NIST's SP800-90B. (FCS_RBG_EXT.1).

7.3.3 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 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_AUD_EXT.1	FPT_STM.1 Reliable time stamps	YES
FAU_STG.1	FAU_GEN.1 Audit data generation	YES, Satisfied by FAU_AUD_EXT.1 FAU_AUD_EXT definition is specified for TMC and audit data collection is done accordance

		FAU_GEN.1 definition.
FAU_STG.4	FAU_STG.1 Protected audit trail storage	YES
FCS_RBG_EXT.1	NA	NA
FCS_CKM.1(1)	[FCS_CKM.2 or FCS_COP.1] and FCS_CKM.4	Satisfied by FCS_COP1.(3) Satisfied by data zeroization which will clear user data and corresponding keys (FDP_ZER_EXT.1).
FCS_CKM.1(2)	[FCS_CKM.2 or FCS_COP.1] and FCS_CKM.4	Satisfied by FCS_COP1.(4) Satisfied by data zeroization which will clear user data and corresponding keys (FDP_ZER_EXT.1).
FCS_COP.1(1)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] and FCS_CKM.4	Satisfied by FCS_RBG_EXT.1. DRBG data is used as a key for AES algorithm. Satisfied by data zeroization which will clear user data and corresponding keys (FDP_ZER_EXT.1).
FCS_COP.1(2)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] and FCS_CKM.4	NO, hashing services do not use keys.
FCS_COP.1(3)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] and FCS_CKM.4	Satisfied by FCS_CKM.1.(1) Satisfied by data zeroization which will clear user data and corresponding keys (FDP_ZER_EXT.1).
FCS_COP.1(4)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] and FCS_CKM.4	Satisfied by FCS_CKM.1.(2) Satisfied by data zeroization which will clear user data and corresponding keys (FDP_ZER_EXT.1).
FCS_COP.1(5)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] and FCS_CKM.4	Satisfied by FCS_RBG_EXT.1. DRBG data is used as a secret key for HMAC algorithm. Satisfied by data zeroization which will clear user data and corresponding keys (FDP_ZER_EXT.1).

FDP_IFC.2	FDP_IFF.1 Simple security attributes	YES
FDP_IFF.1	FDP_IFC.1 Subset information flow control	YES
	FMT_MSA.3 Static attribute initialisation	YES
FDP_DSK_EXT.1	FCS_COP.1 Cryptographic operation	Satisfied by FCS_COP1.(1)
FDP_ZER_EXT.1	FMT_SMF.1	YES. Satisfied for the request of an authorised user.
FIA_UAU.2	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]	YES by FDP_IFC.2 (TRAFFIC policy)
	FMT_SMR.1 Security roles	YES
	FMT_SMF.1 Specification of Management Functions	YES
FMT_MSA.3	FMT_MSA.1 Management of security attributes	YES, for FDP_IFC.2 (TRAFFIC policy)
	FMT_SMR.1 Security roles	YES
FPT_FLS.1	NA	NA
FPT_SBT_EXT.1	FMT_SMF.1	YES. Satisfied for the number of consecutive failed PIN/KEY-admin/KEK authentications attempts to enter an emergency situation.
FPT_STM.1	NA	NA

FPT_TST_EXT.2	FPT_FLS.1 Failure with preservation of secure state.	YES
	FCS_COP.1	Satisfied by FCS_COP1.(1)
FPT_TUD_EXT.1	FCS_CKM.1 & FCS_COP.1	Satisfied by FCS_CKM.1(1) &FCS_COP.1(4)
FPT_PHY_EXT.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

Table 11: Dependencies of Security Functional Requirements

7.4 Security Assurance Requirements (SARs)

The assurance level selected for this ST is EAL2.

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 ST.

Assurance Class	Assurance Component	
	Identifier	Name
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
ASE: Security Target	ASE_CCL.1	Conformance claims

evaluation	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: Test	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

Table 12: Security Assurance Requirements

7.5 Rationale for Security Assurance Requirements

The current ST is claimed to be conformant with the assurance package EAL2.

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 organization being the threats those specified for the intended environment.

8 TOE SUMMARY SPECIFICATION

This chapter describes how the secure functionality meets TOE security requirements. Chapter gives high level view of TOE functionality. Following table describes security functions and corresponding security requirement.

TSF	SFR	Description
Security Audit	FAU_AUD_EXT.1	Extended -Audit data generation
	FAU_STG.1	Protected audit trail storage
	FAU_STG.4	Prevention of audit data loss
Cryptographic Support	FCS_CKM.1(1)	Cryptographic Key Generation (RSA)
	FCS_CKM.1(2)	Cryptographic Key Generation (ECC)
	FCS_COP.1(1)	Cryptographic Operation (AES)
	FCS_COP.1(2)	Cryptographic Operation (HASH)
	FCS_COP.1(3)	Cryptographic Operation (RSA Signature)
	FCS_COP.1(4)	Cryptographic Operation (ECDSA Signature)
	FCS_COP.1(5)	Cryptographic Operation (Keyed HASH)
	FCS_RBG_EXT.1	Extended - Random Bit Generation
Protection of user data	FDP_IFC.2	Complete information flow control
	FDP_IFF.1	Simple security attributes
	FDP_DSK_EXT.1	Extended - Protection of Data on Disk
	FDP_ZER_EXT.1	Extended – Zeroization
Identification and authentication	FIA_UAU.2	User Authentication before any action
	FIA_AFL.1	Authentication failure handling
Security Management	FMT_SMF.1	Specification of management functions
	FMT_SMR.1	Security management roles
	FMT_MSA.1	Management of security attributes
	FMT_MSA.3	Static attribute initialisation
Protection of the TSF	FPT_FLS.1	Failure with preservation of secure state
	FPT_SBT_EXT.1	Extended - Secure Boot and Operation continuity

	FPT_STM.1	Reliable time stamps
	FPT_TST_EXT.2	Extended - Integrity Test
	FPT_PHY_EXT.1	Extended - Passive detection of physical attack
TOE Access	FTA_SSL.1	TSF-initiated session locking
	FTA_SSL.2	User-initiated locking
Trusted path/channels	FTP_ITC.1/VPN-tunnel	Inter-TSF Trusted Channel (Application communications)
	FTP_ITC.1/REM-ADM	Inter-TSF Trusted Channel (remote administration)
	FTP_ITC.1/AUDIT	Inter-TSF Trusted Channel
Trusted Update	FPT_TUD_EXT.1	Extended - Trusted Update

Table 13: SFR of TOE Security Functionality

8.1 Secure Audit

Audit data generation

TOE collects security related information and store audit information to file system. **All administrative security policies enforced on the TOE** is stored to the audit log. Security information content is configurable and following event could be collected:

- Two factor authentication login event
- Failed screen lock code entry attempts
- Application error messages
- All administrative security policies enforced on the TOE

Audit Log events are associate to entity that caused the event. Generally all events are associated to user except few events which are originally triggered by MDM. These MDM events generates trace information via certain applications in the TOE.

TOE Security Functional Requirements addressed: FAU_AUD_EXT.1.1, FAU_AUD_EXT.1.2

Audit Log review

Audit log information is collected by TOE. MDM request audit log information from TOE and TOE identify is based on IMEI. TOE sends collected audit log information, accordance Audit Data Generation, to the MDM server using

MDM application. MDM administrator is able to inspect received data. Audit log Information is formatted such that it's suitable for later interpretation and analysis.

Audit Data protection

Device will protect audit log from unauthorized access and use circular buffering methods to perform data circulation in case of insufficient memory space for log information. Memory spaces for audit log records are configurable only during SW build time.

TOE Security Functional Requirements addressed: FAU_STG.1.1, FAU_STG.1.2, FAU_STG.4.1

8.2 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.

Digital signature is used for the verification of the TOE updates (FPT_TUD_EXT1) 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 accepted and installed (FPT_TST_EXT.2). Key destruction is covered by the FDP_ZER_EXT.1 SFR.

TOE updates are verified against built-in public key and by secure boot chain, through a VPN tunnel.

Whitelisted applications are verified against a list of hashes provided to device via Operational Environment after device has successfully enrolled to the system.

Cryptographic Hashing is used for the integrity monitoring (FPT_TST_EXT.2 - OS, cryptographic functions, VPN-modules, VPN-configuration data and other CSPs.). In addition cryptographic hash functions are also used for digital signatures.

Cryptographic schemes are to be specified for the implementation of trusted channels:
VPN-tunnel (FTP_ITC.1/VPN-tunnel), remote administration (FTP_ITC.1/REM-ADM) and the audit logs sending to an external entity (FTP_ITC.1/AUDIT).

TOE cryptographic operation could be done with SW or with HW. SW operations could be done with OpenSSL or processor library. HW operation could be performed by main processors HW block (QC) or via external secure HW processors.

Cryptographic Key management

TOE supports asymmetric cryptographic key generation using RSA algorithm (with key sizes 2048, 3072 or 4096) and with ECC algorithm (with NIST curves P-256, P-384 or P-521).

Bittium RSA key generation implementation supports only PKCS#1 (pkcs1#1.5 and pss) format and for that reason support of X9.31 is missing. For this reason **FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3**. Standard is not claimed.

TOE Security Functional Requirements addressed:FCS_CKM.1(1), FCS_CKM.1(2)

Cryptographic operations

TOE supports encrypting and decrypting with AES-256 symmetric algorithm in different modes. Cryptographic hashing functionality is supported with SHA1 and SHA2 (224,256,384 and 512bits) hashing algorithms. Cryptographic signature generation and verification is possible to perform with RSA scheme with key sizes 2048, 3072, 4096 bits) or with ECDSA scheme with ECC curves P-256, P-384 and P-521. Hashed message authentication could be performed using HMAC-SHA (HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512) functions with different key sizes and message digest sizes. Supported HMAC cryptographic key sizes are 160, 224,256,384 and 512 with message digest sizes 160, 256,384 and 512bit.

Bittium RSA signature and verification implementation supports only PKCS#1 (pkcs1#1.5 and pss) format and for that reason support of X9.31 is missing. For this reason **FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 5** standard is not claimed.

TOE Security Functional Requirements addressed: FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3) , FCS_COP.1(4), FCS_COP.1(5)

Random bit generation

TOE uses different random sources for different usages. This (DRNG) HW random generator is used by processor internal modules (i.e. Trusted Zone applications).

External secure processor implements (TRNG) HW random generator. This (TRNG) HW random generator is used internally (by secure chip) for HW based crypto operations and by keymaster. Kernel random pool is filled also by this External secure processor random generator combined with interrupt.

Kernel maintains primary entropy, secondary entropy, urandom entropy pools. Primary entropy pool is filled from kernel internal events (interrupts) and from random daemon. Entropy bits from primary pool are derived to next pools when needed by secondary and urandom pools. /dev/random has limited output, when bits are not available reader process is blocked, /dev/urandom reuses existing bits to provide user some random bits.

Random generators are tested accordance NIST test specifications: *NIST specifications, see http://csrc.nist.gov/groups/ST/toolkit/rng/documentation_software.html*

TOE Security Functional Requirements addressed: FCS_RBG_EXT.1

8.3 User data protection

VPN operates by inspecting IP data packets flowing from and to TOE network interface(s) from the network and, based on configuration, drops or allows the packet. If the packet is allowed it is encrypted and passed to the VPN tunnel. Incoming or outgoing IP data packets not part of establishing a VPN tunnel or not inside a VPN tunnel are dropped by the kernel module.

TOE Security Functional Requirements addressed: FDP_IFF.1.1

VPN, by default, blocks all inbound and outbound network connection attempts not explicitly allowed in the VPN configuration. When VPN is enabled and TOE either attempts to send or receive a network packet, VPN configuration rules are queried for a rule matching for the connection attempt. If a matching rule is not found the packet is processed using the default policy which drops the packet. If a matching rule is found, it is applied on the packet. If it is a drop-rule the packet is dropped. If it is a pass-rule the packet is processed based on the rule attributes. Applications installed on TOE can communicate with all of the endpoints allowed in the VPN configuration. Tethering to TOE and accessing the VPN channel using an external device is not possible.

TOE Security Functional Requirements addressed: FDP_IFF.1.2

VPN configuration can be changed directly from the device by importing a configuration file containing a VPN policy or remotely retrieving it from the Bittium Secure Suite server. The former is used when TOE is bootstrapped to an existing Bittium Secure Suite server environment for the first time and the latter when device is already a member of such environment and administrator updates VPN policies through secure trusted channel. All consecutive configuration import attempts directly from TOE must be signed with a valid certificate assigned by the Bittium SafeMove CA.

TOE Security Functional Requirements addressed: FDP_IFF.1.3

VPN authorizes inbound and outbound information flow to communicate outside of the encrypted IPsec tunnel using the following attributes:

- Ipv4
- Protocol
- Source address
- Source port or port range (TCP/UDP only)
- Destination address
- Destination port or port range (TCP/UDP only)

All authorizations are done using the VPN tunneling rules constructed using the attributes described above.

TOE Security Functional Requirements addressed: FDP_IFF.1.4

VPN denies inbound and outbound information flows not part of the VPN connection. ICMPv4 packets can be completely allowed or dropped.

TOE Security Functional Requirements addressed: FDP_IFF.1.5

All data traffic from and to TOE is passed only through the VPN tunnel. Data traffic outside of the VPN tunnel is not possible. Additional information for information flow control policy and functionality:

- TOE user that sends and/or receives data through the TOE to one or more trusted endpoints in the operational environment of the TOE. Administrator or other trusted endpoint user that sends and/or receives data through the operational environment to/from the TOE;
- Data sent from TOE to the trusted endpoints in the operational environment or Data sent from the trusted operational environment endpoints to the TOE;
- All data leaving from the TOE is passed to an encrypted VPN tunnel and to the VPN gateway which will decide to either pass or drop the traffic based on the security attributes configured in the VPN gateway tunnel configuration and rules.
- Data sent to the TOE inside the VPN tunnel is allowed or dropped based on the VPN tunnel traffic selector rules (source or destination address, source or destination port, protocol) configured in the VPN gateway tunnel configuration and rules.
- Data sent to the TOE outside the VPN tunnel is dropped without additional processing by the TOE.

TOE Security Functional Requirements addressed: FDP_IFC.2.1, FDP_IFC.2.2

Symmetric algorithms are to be defined for the disk encryption capability with the DEK (FDP_DSK_EXT.1). User input PIN code/password and user credentials are used to calculate (KEK). Calculated KEK is used to encrypt the Disk Encryption Key (DEK). After user has entered credentials KEK is derived..

TOE Security Functional Requirements addressed: FDP_DSK_EXT.1.1, FDP_DSK_EXT.1.2, FDP_DSK_EXT.1.3, FDP_DSK_EXT.1.4

There are two categories of securely erasures that happen during different types of emergency situations.

1. Securely erase secure storage, user data

2. Securely erase secure storage

The first category happens when the user or the administrator initiates an erasure or when maximum PIN or KEY-admin authentication attempts are exceeded or user PIN authentication attempts-during start-up are exceeded. The second category happens when DM-verity, Serpentine or Frame tamper is detected.

TOE Security Functional Requirements addressed: FDP_ZER_EXT.1.1, FDP_ZER_EXT.1.2, FPT_TST_EXT.2.5

eMMC secure erase is performed accordance with a methods that follows JEDEC eMMC 5.0.standard.

8.4 Identification and authentication

A two factor authentication is performed to authenticate user. A used two factor authentication method includes user credentials and NFC-crypto key (dongle). HMAC-based one-time-password (HOTP) is used by NFC-crypto key to authenticate user. Shared secret is provisioned to NFC key (dongle) during Tough Mobile device initialization by the user organization. Combination of NFC-crypto key and user credentials authenticate user. Administrator authentication is performed utilizing X.509 public key certificates that are stored on the TOE during provisioning. The TSF requires the user or the administrator to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

TOE Security Functional Requirements addressed: FIA_UAU.2.1

Unsuccessful authentication attempts are monitored and safety actions are performed if attempt count exceeds maximum attempt count value. Attempt count value is configurable by administrator. If unsuccessful authentication attempts exceed threshold value TOE goes to erase state and TOE's data storage is cleared.

The PIN 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.

TOE Security Functional Requirements addressed: FIA_AFL.1.1, FIA_AFL.1.2

8.5 Security management

Security management functionality allows user to manage security settings of TOE, download content from application store, allows TOE to enter emergency mode after several number of unsuccessful authentication attempts such PIN key query or unsuccessful decrypting, allows perform emergency secure erase and allows MDM upload audit logs to trusted remote server.

Session between the TOE and the MDM service is managed using a TLS connection with a client certificate and utilizing mutual authentication. When the TOE has a working network connection and has established a VPN tunnel, it attempts to connect to the Bittium Secure Push service. If the connection fails or if existing connection breaks because of network issues, a new connection is established. The applications is not authorised to communicate outside of the tunnel in the VPN-policy by using protocol, source and destination addresses and ports.

TOE Security Functional Requirements addressed: FMT_SMF.1.1, FMT_SMR.1.1

VPN-policy (and other configuration types) can only be configured by the administrator. The initial bootstrap configuration is encrypted and signed by the Bittium SafeMove CA and imported directly from the device. Successive configurations are delivered using the MDM functionality using a device certificate. If a configuration import is attempted directly from the device, the configuration signature is verified against a certificate deployed from the Bittium SafeMove CA.

TOE Security Functional Requirements addressed: FMT_MSA.1.1, FMT_MSA.3.1, FMT_MSA.3.2

8.6 Protection of the TSF

TOE will enter a secure state if security related failure is detected. There are three different types of secure states that depend on the triggering failure. The secure states are

1. Device compromised state
2. Boot loop state
3. Boot halt state

Device compromised state is a User Interface which shows text “Device is tampered, please contact Service”. Screen contains also information about the failure that led to this state. Following tamper failures are possible

- HW tamper (hardware tampering is detected)
  - Serpentine tamper
  - Frame tamper
  - Backup battery depletion
- SW tamper (failed system applications integrity check during runtime)
  - dm-verity tamper (i.e. System image tamper is detected)

Restrictions are implemented to this UI which prevents user to exit from this screen or making any network connections with the exception of backup battery depletion failure that can be ignored by the user which enables normal state of the TOE.

Boot loop state can be caused by failed firmware integrity check during power-up. The firmware integrity check is part of secure boot process which verifies that firmware (boot image that contains the kernel and initial ramdisk) is not modified. The verified firmware contains a root hash which is used by the dm-verity kernel feature to verify the runtime integrity of system image (which contains system applications and VPN module). This is the process how the integrity of the OS is verified.

Boot halt state happens in case of severe HW or SW corruptions that prevent the TOE from continuing the boot flow.

TOE Security Functional Requirements addressed: FPT_FLS.1.1, FPT_SBT_EXT.1.1, FPT_TST_EXT.2.1, FPT_TST_EXT.2.5

Device generates time information which could be used for certificate validation and audit log information.

TOE Security Functional Requirements addressed: FPT_STM.1.1

Whitelist can be used to restrict installation of applications. Authorized applications are verified, in case application whitelist is enabled. Whitelist is provided to MDM application as provisioning parameter and verification is based on that list. Provisioning new whitelist takes effect immediately and causes forced uninstallation for all the applications not in whitelist.

VPN tunnel setup failure will disallow tunnel establishment and usage. During system image tests (at load time) audit logs are available.

TOE Security Functional Requirements addressed: FPT_TST_EXT.2.2

Keystore integrity is verified by TOE per key basis when the encrypted key is accessed. Only the integrity failures are manifested through applications that use the keystore.

TOE Security Functional Requirements addressed: FPT_TST_EXT.2.3

OS integrity is verified automatically during power up by the TOE. Only the integrity failures are manifested through secure states.

TOE Security Functional Requirements addressed: FPT_TST_EXT.2.4

Whitelist can be used to restrict installation of applications. VPN tunnel setup failure will disallow tunnel establishing and usage. Data integrity verification covers only encrypted data in key store.

8.7 TOE access

To start user interactive session with TOE user needs to perform two factor authentications successfully. User credentials and HOTP from NFC key needs to authorize to perform initiated locking. TOE will lock user interactive session after defined time period is expired. Time period is configurable by MDM. User is able to unlock device by entering user credentials.

TOE Security Functional Requirements addressed: FTA_SSL.1.1, FTA_SSL.1.2, FTA_SSL.2.1, FTA_SSL.2.2

8.8 Trusted path/channels

MDM uses PUSH framework owner TLS 1.2, VPN tunnel uses IPsec.

Bittium Secure Push provides a simple, lightweight mechanism that servers can use to tell mobile applications to contact the application server directly, to fetch updated application or user data. The service handles queueing of messages and delivery to the target application running on the TOE.

TOEs main point of interaction is Bittium Secure Push service that provides a trusted channel between the TOE and the Bittium SafeMove Management Server. The connection is protected with TLSv1.2 and authenticated with a device certificate.

Bittium SafeMoveCryptoIP VPN Gateway supports relevant cryptographic algorithms and is compatible for example with NSA Suite B. Supported algorithms and modes include:

- Block ciphers: AES-CBC and SERPENT with key lengths up to 256 bits.
- Hash functions: SHA-2 with hash function lengths up to 512 bits.
- Authenticated encryption mode of AES-GCM with 16-octet Integrity Check Value (ICV).
- Digital signature algorithms RSA and ECDSA.
- Key exchange algorithms using the DH groups 1, 2, 5, 14, 15, 16, 22, 23 and 24, elliptic curve groups 19, 20 and 21 (P-256, P-384 and P-521) and for IKEv2 brainpool elliptic curve groups 27-30 (224bit, 256bit, 384bit, 512bit).

X.509 certificates are recommended for VPN authentication. Bittium Secure Suite comes with an embedded certificate authority (CA) that allows generation and provisioning of X.509 certificates without external CA dependencies. Using external certificate authorities is possible as well and CAs can be integrated into the Secure Suite environment.

TOE Security Functional Requirements addressed: FTP_ITC.1.1(1)(VPN-tunnel), FTP_ITC.1.2(1)(VPN-tunnel), FTP_ITC.1.3(1)(VPN-tunnel), FTP_ITC.1.1(2)(REM-ADM), FTP_ITC.1.2(2)(REM-ADM), FTP_ITC.1.3(2)(REM_ADM), FCS_COP.1(1)

TOE collects log information to the devices /data partition at the runtime. Data collection starts after user is logged to the device. Collected log information is encrypted. Log file handling and sending to the MDM via protected connection using TLS is done by device.

TOE Security Functional Requirements addressed: FTP_ITC.1.1(3)(AUDIT), FTP_ITC.1.2(3)(AUDIT), FTP_ITC.1.3(3)(AUDIT)

8.9 Security updates

FOTA Update

TOE supports secure firmware update over the air functionality. FOTA application is independent from Android's own update service and devices FOTA application downloads content from Bittium own server. FOTA application use HTTPS connection. HTTPS connection is established between device and software update server for control activities and between device and file server for download connection.

New SW package is loaded to SWUP server by server administrator not by the MDM administrator. TOE should check SW Update package availability manually or set automatic update polling period by MDM to get automatically SW indication to the screen when SW is available to downloading. Update process will be triggered by user. MDM administrator cannot perform force installation. Forced FOTA update is not supported due security risks.

Application Update

TOE supports secure application update where only authorized applications are accepted to install. Authorized applications are digitally signed and signature is verified and accepted before installation is allowed. Authorized applications are installed to TOE via Application Library. Whitelist feature could be used to control Application installation. Whitelist feature is configurable by MSM. Application Library and Whitelist is controlled by MDM.

TOE Security Functional Requirements addressed: FPT_TUD_EXT.1.1, FPT_TUD_EXT.1.2, FPT_TUD_EXT.1.3, FPT_TUD_EXT.1.4

9 TERMINOLOGY ACRONYMS AND REFERENCES

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.

Remote Attestation:

Remote Attestation feature provide authorized remote party the requested attestation data. Remote attestation tests the kernel integrity and device HW integrity. Collected values are verified in remote attestation process and integrity of HW and kernel is validated.

User:

In the context of TOE, user is defined as the day to day user for the TOE that has the credentials and the NFC token for operating the TOE.

Administrator:

In the context of TOE, administrator is defined as the SafeMove MDM application that is used to impose usage restrictions and configurations for the user utilizing a trusted secure channel connecting to a remote MDM server available in the Operational environment. Administrating a TOE requires that it has been provisioned with a configuration file and X.509 public key certificates.

9.1 Acronyms

Apps	Applications
BTMC	Bittium Tough Mobile-C
CC	Common Criteria
CSP	Critical Security Parameter
DEK	Disk Encryption Key
EAL	Evaluation Assurance Level
FW	Firmware
HOTP	HMAC-based One-time Password algorithm

HW	Hardware
IC	Integrated Circuit
IP	Internet Protocol
KDF	Key Derivation Function
KEK	Key Encryption Key
NFC	Near Field Communication
MDM	Mobile Device Management
OPS	Organizational Security Policies
OS	Operating System
PP	Protection Profile
QC	Qualcomm
QR-code	Quick Response code
REK	Remote Encrypting Key
SAR	Security Assurance Requirement
SFP	Security Function Policy
SFR	Security Functional Requirement
SPD	Security Problem Definition
SW	Software
TOE	Target Of Evaluation
TPM	Trusted Platform Module
TSFS	TOE Security Functionality
TSS	TOE Summary Specification
TZ	Trust Zone
USB	Universal Serial Bus
VPN	Virtual Private Network

9.2 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