Water Meter Threat Model and Security Analysis (English language Protection Profile)

Architecture & Technology Group

Document number:	DEN0074	Version:	BET00
Date of Issue:	13/02/2018		
Author:	Arm Ltd		
Authorised by:	ATG		

Abstract

Security should start with a Threat Model and Security Analysis (TMSA) that lists the assets that need protection in a system and the threats that are considered in scope. From this starting point, a step by step process can be used to establish security objectives and Security Functional Requirements (SFRs). With the inherent diversity of IoT there will be a greater need for device manufacturers to have a reference TMSA for their product. Arm has created a series of reference English language Protection Profiles for IoT products to show how this might be done in a way that is understandable by non-security experts. These security analyses are accompanied by at a glance summary documents and useful appendices that show how Arm TrustZone and CryptoIsland technology can be used to meet some of the SFRs. We hope that you find these documents useful as a starting point for creating a TMSA for your IoT device.

Keywords

Platform Security Architecture, PP, Protection Profile, PSA, Threat Model Security Analysis, TMSA, TrustZone, Water Meter

Distribution list

Name	Function		Name	Function

Non-Confidential Proprietary Notice

Contents

1 About this document

1.1 PP Identification

Title:	Water Meter Security Module Protection Profile
Authors:	Arm Ltd
CC Version:	3.1 revision 5
Assurance Level:	EAL 2
Reference:	
Version Number:	
Keywords:	Water meter

1.2 Change control

This document is tracked in SharePoint internally.

1.3 Current status and anticipated changes

Current Status: Beta

1.4 Change history

Release Date	Version	Comments
24/11/2017	0.1	First complete version
26/12/2017	0.2	Added Appendix on TZ-PSA support
16/01/2018	0.3	Fixes and template modification

1.5 References

This document refers to the following documents.

Ref	**Doc No**	**Author(s)**	**Title**
[CC-1]	CCMB-2017-04-001		Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, April 2017. Part 1: Introduction and general model.
[CC-2]	CCMB-2017-04-002		Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, April 2017. Part 2: Security functional components
[CC-3]	CCMB-2017-04-003		Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, April 2017. Part 3: Security assurance components

[CEM]	CCMB-2017-04-004		Common Methodology for Information Technology Security Evaluation (CEM), Version 3.1, Revision 5, April 2017. Evaluation methodology
[Comp]			Joint Interpretation Library, Composite product evaluation for Smart Cards and similar devices, Version 1.2, January 2012
[GPRoT]	GP_REQ_025	GlobalPlatform	Root of Trust Definitions and Requirements, March 2017, Version 1.0.1

1.6 Terms

This document uses the following terms and abbreviations.

Term	**Meaning**
API	Application Programming Interface
CC	Common Criteria
EAL	Evaluation Assurance Level
LPWAN	Low-Power Wide-Area Network
MCU	Microcontroller Unit
NB-IoT	Narrow Band Internet of Things
OS	Operating System
OSP	Organisational Security Policy
OTP	One-Time-Programmable
PKI	Public Key Infrastructure
PP	Protection Profile
PSA	Platform Security Architecture
RAM	Random Access Memory
RNG	Random Number Generation
ROM	Read Only Memory
SFP	Security Function Policy
SFR	Security Functional Requirement
SoC	System-on-Chip
ST	Security Target

TEE	Trusted Execution Environment
TOE	Target of Evaluation
TRX	Transceiver
TSF	TOE Security Functionality
TSFI	TSF Interface
TSS	TOE Security Service

1.7 Terminology and Definitions

1. The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119]:

MUST: This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.

MUST NOT: This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.

SHOULD: This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

SHOULD NOT: This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.

MAY: This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

2 Introduction

2. This section provides an overview of the TOE.

2.1 TOE Overview

2.1.1 TOE Type

3. TOE of this PP is a security module for water meters as used for standard home and office locations. Such meters are owned by the water distribution company. In particular, the PP doesn’t cover industrial meters for large volumes of water and likely to include additional functions.

4. The TOE is a platform composed of a hardware device and a firmware implementing the water meter functionalities. The firmware itself may include a generic purpose operating system.

2.1.2 TOE Usage and Major Security Features

5. Water meters are deployed in individual homes and offices to automatically report on water consumption, and most of them share the following features:

Measuring water flows. Flow measurement is a meter’s main role, usually as input to the water distributor’s billing system.

Sensor data aggregation and analysis. Meters usually process the raw flow data before sending it, in order to transform it into “ticks” or volume; analysis may also be used to detect leaks or other abnormal situations.

Communication of measurements. A meter regularly transmits its measurements to a central authority, here through a LPWAN like LoRa or NB-IOT.

Battery-powered with long lifecycles. Meters are not expensive, but installation and maintenance costs are and must be minimized. A meter’s battery lifecycle must therefore be as long as possible, to avoid maintenance.

Limited over-the-air maintenance. Low-bandwidth protocol only support limited maintenance; over-the-air firmware update is not possible with the current LoRa specifications. Some maintenance may be possible through a local port (see the optional features below).

Massive deployments. Large distributors need to install and maintain millions of meters. This causes difficulties for local maintenance, both in terms of costs and in terms of delays.

Long lifecycle. Water meters are not expected to be changed often, and they are expected to be installed every 10 to 25 years, with minimal maintenance between changes. This may cause security concerns, in particular as meters age and hardware-related vulnerabilities arise.

2.1.3 Required non-TOE Hardware/Software/Firmware

6. The flow sensor, which measures water consumption and reports it to the security module, is out of the TOE.

7. The LPWAN transceiver (TRX), which provides low power communication, is also out of the TOE.

2.2 TOE Description

8. The figure below illustrates the main components for a water meter and the TOE for this PP.

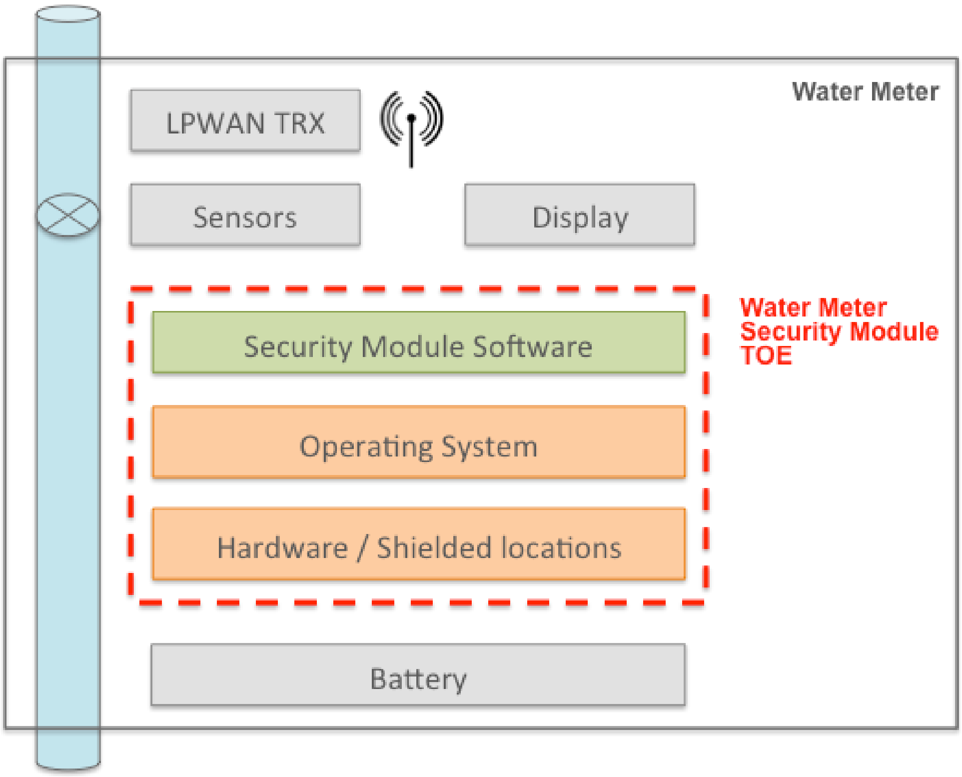

Figure 1: Water Meter TOE

2.2.1 TOE Features

9. In order to protect the metering function and access to management and administration interfaces, water meters include at least the following security features:

Device and server authentication. Maintenance operations will be performed either remotely (from a server) or locally through a dedicated device, so authentication is likely to be performed from one of these.

Authorization. Even if the authenticated "users" are not human, it remains important not to associate all principals to the same rights, and authorization remains important.

Network authentication. The establishment of a network connection requires a mutual authentication between the device and the remote server or user.

Secure communication. Any network communication is performed using a protocol that includes integrity and confidentiality protections.

Measurement authenticity. It may be required to include a proof of authenticity of measurements at the application level (i.e. a proof that is not linked to communications and can be archived with the content).

Log of measurements and security events. Measurements are logged for a given amount of time. Security events are also logged locally on the meter, to be made available in the forensic analysis of an attack or other suspicious event.

Software update. The software running on the meter can be updated in order to fix vulnerabilities identified after the device's deployment. Depending on the network, over-the-air update may not be available; in that case, it may be interesting to have the ability to remotely modify the meter's configuration to disable a potentially vulnerable function if that is possible.

Tampering detection. The software must be able to detect attempts to tamper with the measurement, possibly by analyzing measurement patterns.

10. Water meters only offer a limited interfaces. Final customers don't connect directly to the meters. They only have access to a display with their water consumption. They can also access their data through a Web-based interface on backend systems.

11. All management and maintenance operations are performed either through the backend, or by operators through other devices (such as a portable device); the operators will be authenticated on their device, which will itself authenticate to the meter before to perform an operation.

2.2.1.1 Hardware

12. Hardware for a Water Meter Security Module is typically composed of a microcontroller with embedded flash memory and a LPWAN controller.

13. The microcontroller may support OTPs to store sensitive data, such as Water Meter ID or secrets.

2.2.1.2 Firmware

14. Firmware for a Water Meter Security Module is typically composed of a boot-loader, which is the first piece of code called by the ROM, an operating system for microcontroller and a Security Module Software running on top of this OS.

15. The Security Module Software is responsible for implementing TOE functionalities.

16. Firmware is usually stored on a flash memory to support upgrade.

2.2.2 TOE Operational Environment

17. The TOE operational environment is composed of the flow sensor, which measures water consumption and the backend servers used for uploading measurements and administrating the meters.

18. Additional water meter sensors such as the battery sensor, and pH, temperature, conductivity or other water quality sensors, and water meter components, such as a display, may also be part of the TOE operational environment.

19. In a network architecture based on gateways to aggregate communications between devices and backend-server, the gateway is also part of the operational environment. The gateway may be the endpoint of protected communication channel with the devices.

2.2.3 TOE Life Cycle

20. The TOE Life Cycle is as follows:

Phase	Actors
1 & 2: Firmware / Software / Hardware design	The water meter security module software developer is in charge of software development and testing. The device manufacturer may design additional software that will be linked with the water meter in phase 4. The water meter security module hardware designer is in charge of designing (part of) the processor(s) where the water meter software runs and designing (part of) the hardware security resources used by the water meter. The silicon vendor designs the ROM code and the secure portion of the water meter chipset.
3: Silicon/chip manufacturing	The silicon vendor produces the chipset for the water meter security module device.
4: Software manufacturing	The device manufacturer is responsible for the integration, validation, and preparation of the software to load in the water meter security module.
5: Device manufacturing and personalization	The device manufacturer is responsible for the device assembly and initialization and any other operation on the device before delivery to the end user. The water meter is personalized with credentials, in particular for network authentication.
6: Operational phase	The end user gets a device ready for use. The device may have to register to the network it uses. The water meter may be updated if it has not been designed to be immutable.
7: End-usage termination	The end user terminates their relationship to allow device reuse by performing a factory reset of the water meter.

21. Phases 1 to 5 are performed by trusted personnel in secure environments.

22. The TOE delivery point may occur at the end of phases 3, 4 or 5.

3 Conformance Claims

3.1 CC Conformance Claim

23. This Protection Profile is CC Part 2 [CC2] and CC Part 3 [CC3] conformant of Common Criteria version 3.1, revision 5.

3.2 Package Claim

24. The minimum assurance level for the evaluation of a Water Meter with a TOE conformant to this PP is EAL 2.

3.3 PP Claim

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

3.4 Conformance Claim to this PP

26. The conformance to this PP, required for the Security Targets and Protection Profiles claiming conformance to it, is demonstrable, as defined in CC Part 1 [CC1].

4 Security Problem Definition

4.1 Users and External Entities

27. The external entities that are considered in this PP are:

Remote Admin: This entity operates from backend servers and can configure the water meter remotely.
Local Admin: This entity operates locally and can configure the water meter and perform firmware update.
Attacker: This user can be the Customer, to modify its water consumption for instance, or any other attacker, for financial or malevolent reasons. He can operate remotely or locally.

28. Remote and Local Admin entities are not necessarily users but can be devices or systems controlled by trusted users.

4.2 Assets

4.2.1 TSF Data

29. The following assets contain data that belong to TSF.

4.2.1.1 Meter ID

30. A unique ID to identify the device on a network, which may be the MAC address of the device or the DevEUI unique identifier for LoRa devices.

31. Properties: Integrity

4.2.1.2 Firmware

32. The water meter's firmware.

33. Properties: Integrity, Authenticity

4.2.1.3 Firmware Certificate

34. The cryptographic certificate used to authenticate firmware and firmware updates.

35. Properties: Integrity

4.2.1.4 Logs

36. The event logs, that can be used to detect suspicious activities.

37. Properties: Integrity

4.2.2 User Data

4.2.2.1 Measurements

38. The water flow measurements produced by the meter, in several instantiations:

- The raw measurement data from the sensor that is used as input by the meter's software.
- The metering data and alerts sent over the network by the meter.
- The metering data and alerts stored temporarily (for aggregation and analysis) and persistently (for logging) on local storage.

39. Properties: Integrity, Confidentiality (for privacy reasons)

4.2.2.2 Configuration

40. The water meter's configuration, split into two components:

- The meter's software configuration, including the measurement patterns, the aggregation method, and the alert trigger configuration.
- The meter's network configuration, including IP address of backend servers and security settings.

41. Properties: Integrity

4.2.2.3 Credentials

42. Authentication credentials, used for local and remote authentication, and for data protection during communication. In the case of a LoRa network, they are very simple:

- A secret key to be used when the device joins a network.
- A secret key used to authenticate messages once a device has joined a network.
- A secret key used to encrypt messages once a device has joined a network.
- Device authentication credentials to authenticate with locally connected devices (if such a connection is available).
- Device authentication data, such as public key certificates.

43. Properties: Integrity, Confidentiality

4.3 Threats

44. An attacker is a threat agent (a person or a process acting on his/her behalf) trying to undermine the TOE security policy defined by the current ST and, hence, the TSF. The attacker especially tries to change properties of the assets defined in Section 4.2.

4.3.1 T.IMPERSONATION

45. An attacker impersonates a maintenance device on the local interface.

46. The credentials may be obtained through insecure communication protocols, or exposed through data disclosure.

47. They attacker may then modify configuration, firmware or logs.

48. Assets threatened directly: Credentials
Assets threatened indirectly: Firmware, Configuration, Logs.

4.3.2 T.MITM

49. An attacker performs a Man-In-The-Middle attack or impersonates a backend server.
50. The attacker may alter or modify messages exchanged with the device.
51. They attacker may then disclose and modify Measurements, Logs, Credentials, Configuration data.
52. Assets threatened directly: Credentials (Server), Logs, Measurements, Configuration

4.3.3 T.FIRMWARE_ABUSE

53. An attacker installs a flawed version of the firmware and obtains partial or total control of the meter. The firmware may have been modified prior to the attack to include a malware or consist of an outdated version of the original firmware.
54. The attacker may for instance modify on the device the value of the firmware certificate used to authenticate the installed firmware or firmware updates.
55. The attacker may exploit functionalities of the TOE, which should not be available at the current life-cycle state of the TOE.
56. Such an attack can allow for cloning the device, modifying the actual measurements or logs of the device, getting access to non-authorized features or performing a denial-of-service.
57. Assets threatened directly: Firmware, Firmware Certificate
Assets threatened indirectly: All.

4.3.4 T.REPUDIATION

58. A User of the water meter denies action performed on the TOE on its behalf.
59. This can be the local or remote administrator for configuration or firmware update.
60. Assets threatened directly: Logs, Measurements, Firmware.

4.3.5 T.TAMPER

61. An attacker tampers with the meter and tries to access or modify assets in persistent or volatile memory. The main targeted assets are Measurements, Logs, Credentials, Configuration data.
62. To perform this attack, the attacker may use debug functionalities or directly access memories.
63. Such an attack can for instance allow for cloning the device, modifying the actual measurements or logs of the device, getting access to non-authorized features or performing a denial-of-service.
64. Assets threatened directly: All.

4.4 Organisational Security Policies

65. The TOE and its environment shall comply with the following organizational security policies (OSP) as security rules, procedures, practices or guidelines imposed by an organization upon its operation.

4.4.1 P.KEYS_MANAGEMENT

66. The cryptographic keys, credentials and certificates used in the TOE shall be securely generated, provisioned on the TOE.

67. Additionally, they should be securely managed during the life-cycle of TOE when used outside of the TOE (such as in gateways, back-end servers or maintenance devices).

4.5 Assumptions

68. This section describes the assumptions about the operational environment of the TOE.

4.5.1 A.TRUSTED_ADMIN

69. Admin of the TOE are assumed to follow and apply administrative guidance in a trusted manner.

5 Security Objectives

5.1 Security Objectives for the TOE

5.1.1 OT.ACCESS_CONTROL

70. The TOE shall authenticate Remote and Local Admin entities before granting access the water meter configuration and logs and before performing firmware update.

5.1.2 OT.SECURE_STORAGE

71. The TOE shall protect integrity and confidentiality of Credentials when stored, and protect integrity of Firmware Certificate, Configuration and Logs when stored.

5.1.3 OT.FIRMWARE_AUTHENTICITY

72. The TOE shall authenticate and verify integrity of firmware image during boot and of new firmware versions prior upgrade.

73. The TOE shall also reject attempts of firmware downgrade.

5.1.4 OT.COMMUNICATION

74. The TOE shall only accept remote connections from configured back-end servers and be able to authenticate these servers.

75. The TOE shall also provide authenticity, confidentiality and replay protection for export outside of the TOE.

5.1.5 OT.AUDIT

76. The TOE shall maintain log of all significant events and allow access and analysis of these logs to authorized users only.

5.1.6 OT.SECURE_STATE

77. The TOE shall maintain a secure state even in case of failures, for instance failure of verification of firmware integrity.

5.1.7 OT.TAMPER

78. The TOE shall protect and react against physical tampering attempts.

5.2 Security Objectives for the Operational Environment

5.2.1 OE.CREDENTIALS_MANAGEMENT

79. Identical toP.KEYS_MANAGEMENT (p. 16).

5.2.2 OE.TRUSTED_ADMIN

80. The Admin of the TOE is not careless, wilfully negligent or hostile.

5.3 Security Objectives Rationale

81. The following table provides an overview for security objectives coverage (TOE and its environment) and also gives an evidence for sufficiency and necessity of the defined objectives. It shows that all threats and OSPs are addressed by the security objectives and it also shows that all assumptions are addressed by the security objectives for the TOE operational environment.

	OT.ACCESS_CONTROL	OT.SECURE_STORAGE	OT.FIRMWARE_AUTHENTICITY	OT.COMMUNICATION	OT.AUDIT	OT.SECURE_STATE	OT.TAMPER	OE.CREDENTIALS_MANAGEMENT	OE.TRUSTED_ADMIN
T.IMPERSONATION	X				X			X	
T.MITM				X					
T.FIRMWARE_ABUSE	X		X			X			
T.REPUDIATION	X			X	X				
T.TAMPER		X				X	X		
P.KEYS_MANAGEMENT	X							X	
A.TRUSTED_ADMIN									X

Table 1: Security Objectives Rationale

82. A justification required for suitability of the security objectives to cope with the security problem definition is given below.

5.3.1 Security Objective Rationales: Threats

5.3.1.1 Threat: T.IMPERSONATION

83. This threat assumes that the TOE can be attacked by impersonating of a legitimate user. This threat is countered by the security objectives OT.ACCESS_CONTROL that ensures authentication of users to access TOE functionalities and OT.AUDIT that allows for audit of TOE users activities and by the security objective on the operational environment OE.CREDENTIALS_MANAGEMENT that ensures that no default password can be used on operational usage.

5.3.1.2 Threat: T.MITM

84. This threat assumes that the TOE can be attacked by intercepting or spying communications with remote servers. This threat is countered by the security objective OT.COMMUNICATION that ensures authentication of remote servers and protection in confidentiality and integrity of exchanged data.

5.3.1.3 Threat: T.FIRMWARE_ABUSE

85. This threat assumes that the TOE can be attacked by modifying the firmware or installing and outdated flawed version. This threat is countered by the security objectives OT.ACCESS_CONTROL that ensures that

only Admin can initiate firmware upgrade, OT.FIRMWARE_AUTHENTICITY that ensures verification of firmware authenticity prior use and prior upgrade and OT.SECURE_STATE that ensures that the TOE maintains a secure state even in case of failure of verification of firmware integrity.

5.3.1.4 Threat: T.REPUDIATION

86. This threat assumes that TOE users can deny their actions on the TOE. This threat is countered by the security objectives OT.ACCESS_CONTROL that ensures authentication of users to access TOE functionalities, OT.COMMUNICATION that ensures protection in authenticity of exported TOE data and OT.AUDIT that allows for audit of TOE users activities

5.3.1.5 Threat: T.TAMPER

87. This threat assumes that the TOE can be attacked by physical tampering. This threat is countered by the security objectives OT.SECURE_STORAGE that ensures a secure storage for TOE assets, by OT.SECURE_STATE that ensures that the TOE maintains a secure state in case of failure and by OT.TAMPER that ensures protection and reaction to physical tampering attempts.

5.3.2 Security Objective Rationales: Security Policies

88. Each identified security policy in this Security Target is addressed by at least one security objective for the TOE or security objective for the operational environment. This section provides a mapping from each security policy to the security objectives and provides a rationale how the security policy is fulfilled.

5.3.2.1 Policy: P.KEYS_MANAGEMENT

89. This security policy is directly upheld by the security objective on the operational environment OE.CREDENTIALS_MANAGEMENT.

5.3.3 Security Objective Rationales: Assumptions

90. Each security assumption in this Security Target is addressed by at least one security objective for the operational environment. This section maps assumptions to environmental security objectives and provides a rationale how the assumption is fulfilled.

5.3.3.1 Assumption: A.TRUSTED_ADMIN

91. This security policy is directly upheld by the security objective on the operational environment OE.TRUSTED_ADMIN.

6 Security Requirements

6.1 Security Functional Requirements

92. This part of the ST defines the detailed security functional requirements that are satisfied by the TOE.

93. These requirements are derived from the Security Objectives for the TOE (Section 5.1). Each sub-section is labelled with a security objective and provides the corresponding requirements.

94. As defined in Section1.7, “shall” represent mandatory requirements, while “should” denotes requirements for which there may exists valid reasons to ignore them. However, if such a requirement is ignored, the full implications must be understood and the ST shall justify any removal of such requirements.

6.1.1 OT.ACCESS_CONTROL

95. The TOE shall maintain the roles Local Admin and Remote Admin.

96. The TOE shall allow authentication of entities according to these roles through user-initiated interactive sessions.

97. **Note 1**: Depending of the implementation, Remote and Local Admin entities are either local system users or external devices or systems controlled by trusted users.

98. **Note 2**: The ST writer shall explicit how credentials for entities authentication are managed on the TOE. For local users, these credentials may consist of passwords, stored locally as salted hashes and diversified from one device to another. For external devices, this may be through certificate-based authentication or also for backend systems, this may rely on the remote entity authentication performed during communication establishment.

99. The TOE shall manage a threshold for unsuccessful authentication attempts. The ST writer shall precise the actions taken is this threshold is reached.

100. The TOE shall require each entity to be successfully authenticated before allowing any other actions on behalf of that user.

101. The TOE shall allow termination of user’s own interactive session and automatically terminate a remote interactive session after session inactivity.

102. The TOE shall enforce an access control policy on TOE assets and operations based on the identity of the user requesting access. The ST writer shall define rules of this policy.

103. **Note 3**: This policy will typically include rules such as:

Access to Configuration, Logs is only allowed to authenticated users with role Remote Admin.

Access to Credentials assets, Firmware upgrade operation is only allowed to authenticated users with role Local Admin.

104. The TOE shall prevent unauthorized uses of all assets. In particular, the TOE shall prevent reading of all Credentials and shall not provide an interface to do so.

6.1.2 OT.SECURE_STORAGE

105. The TOE shall monitor for integrity errors assets with a security need for integrity (Meter ID, Firmware, Firmware Certificate, Logs, Configuration, Credentials).

106. **Note 4**: The TOE will typically ensure integrity either with hardware based write-once mechanisms, such as OTP, or through cryptographic hash functions. In the latter case, the ST writer shall explicit the

cryptographic algorithms used for secure storage and related key characteristics and random generation methods.

107. Upon detection of a data integrity error, the TOE shall maintain a secure state. The ST writer shall specify reaction of the TOE in this case.

108. **Note 5**: For assets with a security need for confidentiality (Credentials), protection of relies on access control measures (OT.ACCESS_CONTROL). However the TOE may offer additional protection by encryption of persistent memory. The ST writer shall specify the mechanism used and related encryption techniques.

6.1.3 OT.FIRMWARE_AUTHENTICITY

109. The TOE shall rely on a secure boot mechanism to authenticate and verify integrity of firmware prior transferring control to the firmware.

110. **Note 6**: A secure boot will typically rely on a multi-stage boot process where the authenticity of the first stage is assumed from read-only memory and other stages with verification of cryptographic signatures with asymmetric keys. The ST writer shall explicit which signature schemes are used at the various stages, including the hash algorithm, and the size of the various parameters (e.g., modulus of 2048 bits and exponent of 32 bits for RSASSA-PSS with SHA-512). He shall also specify the list of standards that are met by the chosen schemes or none.

111. If the firmware is loaded from a removable media, the TOE shall use a persistent storage to store the version of the last installed firmware and compare this version to the version from the loaded firmware to prevent loading of an out-dated firmware.

112. Upon detection of a firmware authenticity error, the TOE shall maintain a secure state. The ST writer shall specify the action to be taken if the verification fails (cf. OT.SECURE_STATE).

113. **Note 7**: The TOE may enter a maintenance mode where the ability to return a secure state is provided.

114. On firmware upgrade requests, the TOE shall first authenticate the upgrade binary based on digital signature and verify its integrity. The TOE shall also check that version of the firmware for upgrade is more recent than the firmware currently installed.

115. **Note 8**: The ST writer shall explicit which signature scheme is used.

116. Upon detection of an error during upgrade, the TOE shall revert to the version of the firmware prior the upgrade request.

117. The TOE should provide the ability to check availability of firmware upgrade and notify Admin.

6.1.4 OT.COMMUNICATION

118. The TOE shall establish a trusted communication channel with remote servers prior any exchange of TSF data or User data and verify if the peer certificate is valid.

119. The TOE shall prevent the disclosure and modification of user data when exporting user data outside of the TOE.

120. **Note 9**: Protection of user data relies on the encryption techniques provided with the trusted communication channel. The ST writer shall explicit which message integrity protection and encryption algorithms are used and related key sizes.

121. **Note 10**: In LoRa based network, an application session key (AppSKey) is used to protect the confidentiality of messages and a network session key (NwkSKey) to protect the integrity of the message. Related encryption algorithm is AES with a key size 128 bit used to encrypt message payload and to generate a message integrity code.

122. The TOE shall prevent replay of messages exchanged with the TOE.

123. **Note 11**: In LoRa based network, an incrementing frame counter is used to identify each received of sent message. Message with a frame value lower than the current value are discarded.

124. When the TOE is activated on the field and must request to join the network, the ST writer shall explicit which authentication and session keys derivation algorithms are used and related key sizes.

125. **Note 12**: In LoRa based network, devices can either be activated in factory with the application and network session keys or activated over-the-air. In the latter case, an AES-128 key (AppKey) personalized in factory is used to authenticate and derive the session keys with network equipment.

6.1.5 OT.AUDIT

126. The TOE shall maintain an audit trail of security events. Each record shall mention the nature of the event, date and time of the event and the user, if any, responsible for the event.

127. **Note 13**: The ST writer shall explicit which events are logged. This will include at least failed and successful authentication attempts, firmware upgrade requests and progress, integrity errors, cryptographic errors.

128. The TOE shall prevent users from deleting entries from the audit trail.

129. **Note 14**: The only audit trail operations and interfaces that should be available on the TOE are appending a line to the audit trail and export outside of the TOE.

6.1.6 OT.SECURE_STATE

130. The TOE shall ensure residual information protection for credentials and session keys after they are being used.

131. Debug features of the TOE shall be deactivated or protected by a mechanism with the same level of security assurance as the PP.

132. The TOE shall maintain a secure state in case of failures, such as firmware integrity error, firmware upgrade error, RNG error, failure to establish a trusted communication channel.

133. **Note 15**: If the TOE should encounter a failure in the middle of a critical operation, the TOE should not just quit operating, leaving key material and user data unprotected. The ST writer shall specify

134. **Note 16**: In case of critical security event, the TOE may for instance notify backend system, securely erase credentials, switch to a maintenance mode.

135. The TOE shall periodically perform self-tests to check the correct operation of the security functions.

6.1.7 OT.TAMPER

136. The TOE shall detect physical tampering attempts and maintain a secure state (cf. OT.SECURE_STATE). The ST writer shall explicit which attacks can be detected.

137. **Note 17**: Typical detected attacks include environmental stress such as power glitch, damaged mesh lines, physical access to the TOE (use of sensors).

6.2 Security Assurance Requirements

138. The current assurance package was chosen based on the pre-defined assurance packet EAL 2. EAL 2 is chosen because the threats that were chosen are consistent with an attacker of basic attack potential.

7 Acknowledgements

139. This document was prepared for Arm by Prove & Run
http://www.provenrun.com

Appendix A Support of SFRs by Arm CryptoIsland IP

140. This appendix explains how SFRs of this PP can be implemented using an Arm Cortex-M microcontroller embedding Arm CryptoIsland IP.

PP Requirement	Support from CryptoIsland IP
OT.ACCESS_CONTROL	
Authentication of Admins	Secure cryptographic and RNG support. This feature can be used to support cryptographic algorithms used for authentication.
Access control policy on assets	Data protection functionalities, in particular support for asset use policy. This feature can be used to implement an access control policy on TOE assets based on the identity of the requester and additionally on the lifecycle state, the intended usage, and HW interface used for the request
OT.SECURE_STORAGE	
Integrity and confidentiality protection for stored assets	Persistent trusted storage based on OTP and local storage protected by an encryption key (AES-256 key). This feature, that offers integrity and confidentiality protection, can be used to store assets. OTP will be reserved for immutable assets, such as the Meter ID, and local storage for other assets.
OT.FIRMWARE_AUTHENTICITY	
Verification of firmware authenticity prior boot	Loaded SW validation functionality that authenticates loaded images based on a hardware root of trust. This feature can be used as part of the secure boot process to verify firmware during device start-up.
Verification of firmware authenticity prior update	SW update validation. This feature can be used to verify integrity and authenticity of firmware update image. The firmware authenticate is based on a cryptographic signature with PKI. It reports failures during the update process and fails back on the last valid image.
Anti-rollback for firmware update	SW update validation. This feature can also verify freshness of firmware update image.
OT.COMMUNICATION	
Authentication of remote servers	Secure cryptographic and RNG support. This feature can be used to implement and support cryptographic protocols for communication. In particular, the AES algorithm used to protect LoRa communication is supported. Related cryptographic keys can be stored in the persistent trusted storage provided by CryptoIsland IP.
Integrity and confidentiality protection for exchanged assed	
Replay protection	No direct support.
OT.AUDIT	
Audit trail of security events	No direct support.

Protection of audit trail	Persistent trusted storage functionality can be used to security store and control accesses to audit trails.
OT.SECURE_STATE	
Residual information protection for confidential assets	No direct support.
Protection of debug features	Authenticated debug functionality. Debug certificates can be used to protect and activate debug features of the processor.
Secure state in case of failure	Alarm signals handling. Possible reactions include for instance aborting current operation, resetting the processor, deactivating the device, zeroizing keys.
Self-tests	No direct support.
OT.TAMPER	
Detect physical tampering attempts	Alarm signals handling functionality. This feature can be used to trigger trusted response to alarm signals provided by external sensors/detectors.

Appendix B Support of SFRs by Arm TrustZone PSA IP

141. This appendix explains how SFRs of this PP can be implemented using an Arm Cortex-M microcontroller embedding Arm TrustZone-based PSA Secure Processing Environment IP.

PP Requirement	Support from TrustZone-based PSA IP
OT.ACCESS_CONTROL	
Authentication of Admins	Cryptographic Operations Trusted Functions. Related functions can be used to support cryptographic algorithms used for authentication. Trusted Device Initialization can be used to provision related secrets to the device.
Access control policy on assets	Related assets can be controlled and isolated from the Non Secure Processing Environment by a Secure Partition.
OT.SECURE_STORAGE	
Integrity and confidentiality protection for stored assets	Secure Storage/Data sealing Trusted Functions.
OT.FIRMWARE_AUTHENTICITY	
Verification of firmware authenticity prior boot	Trusted Boot features can be used for an authenticated boot process.
Verification of firmware authenticity prior update	Firmware Update features and related firmware update agent can be used to authenticate and authorize firmware updates.
Anti-rollback for firmware update	No direct support.
OT.COMMUNICATION	
Authentication of remote servers	Cryptographic Operations and RNG Trusted Functions. Related functions. In particular, the AES algorithm used to protect LoRa communication is supported. Trusted Device Initialization can be used to provision related secrets to the device.
Integrity and confidentiality protection for exchanged assed	Cryptographic keys for authentication can be stored in the persistent trusted storage provided by Secure Storage Trusted Functions.
Replay protection	No direct support.
OT.AUDIT	
Audit trail of security events	Audit Logs Trusted Functions.
Protection of audit trail	Audit Logs Trusted Functions.
OT.SECURE_STATE	
Residual information protection for confidential assets	No direct support.
Protection of debug features	Secure Debug.
Secure state in case of failure	Secure functions are isolated from failure from the Non Secure Processing Environment.
Self-tests	No direct support.
OT.TAMPER	

Detect physical tampering attempts	No direct support.

Appendix C Compatibility with Root-of-Trust PP

142. The Root of Trust Protection Profile targets platforms that provide a set of trusted and basic functions or services from which an initial chain or trust can be derived. It is based on the GlobalPlatform *Root of Trust Definitions and Requirements* document [GPRoT]. The PP is a modular-PP, organized as a base-PP corresponding to the Root of Trust platform and PP-modules corresponding to optional security services based on top of this platform, such as authentication, confidentiality, authorization or update services.

143. This appendix explains how SFRs of this PP can inherit from the requirements set in the Root of Trust PP and related PP-modules.

PP Requirement	Support from a Root of Trust
OT.ACCESS_CONTROL	
Authentication of Admins	Root of Trust with an Authentication Service allows authenticating users.
Access control policy on assets	Root of Trust with an Authorization Service allows enforcing an access control policy on TOE assets.
OT.SECURE_STORAGE	
Integrity and confidentiality protection for stored assets	A Root of Trust with a Confidentiality and Integrity Services allows enforcing confidentiality and integrity of storage for TOE assets.
OT.FIRMWARE_AUTHENTICITY	
Verification of firmware authenticity prior boot	A Root of Trust with a Verification Service allows verifying the authenticity of firmware.
Verification of firmware authenticity prior update	A Root of Trust with an Update Service allows enforcing integrity and authenticity of firmware update.
Anti-rollback for firmware update	
OT.COMMUNICATION	
Authentication of remote servers	Root of Trust with an Authentication Service allows authenticating remote entities.
Integrity and confidentiality protection for exchanged assed	No direct support.
Replay protection	No direct support.
OT.AUDIT	
Audit trail of security events	No direct support.
Protection of audit trail	No direct support.
OT.SECURE_STATE	
Residual information protection for confidential assets	No direct support.
Protection of debug features	No direct support.
Secure state in case of failure	No direct support.
Self-tests	No direct support.
OT.TAMPER	
Detect physical tampering attempts	No direct support.