Embedded Firmware Security Solution of Connectivity Features V1.0 for Arçelik Bluetooth IoT Devices

SECURITY TARGET V0.9

ARÇELİK A.Ş.

Sensitivity: Public

Document History

Version	Description
0.1	First Draft
0.2	SFR Requirement Revision
0.3	SFR Rationale and SFR- Objective Rationale Table Revision
0.4	Typo Correction
0.5	Logical Scope of TOE Definition Update, SFR Revision, GR03 Solution
0.6	SFR Assignment Revision [FDP_IFC.1(1/2), FDP_IFF.1(1/2)]
0.7	SFR Content Revision, GR06 Solution
0.8	SFR Content Revision, GR06 Solution (v2.1.2)
0.9	Typo Correction

Table of Contents

1 INTRODUCTION

1.1 REFERENCES

This section provides information to refer to the Security Target (ST) and Target of Evaluation (TOE) as in the following Table. The ST is identified by ST Title (including the TOE identification) and ST Version. The TOE is identified by TOE Title and the TOE Version.

ST Title	Embedded Firmware Security Solution of Connectivity Features v1.0 for Arçelik Bluetooth IoT Devices Security Target
ST Version	V0.9
TOE Title	Embedded Firmware Security Solution of Connectivity Features v1.0 for Arçelik Bluetooth IoT Devices
TOE Version	v1.0
Assurance Level	EAL2
CC Identification	▪ Common Criteria Part 1 Version 3.1 Revision 5 ▪ Common Criteria Part 2 Version 3.1 Revision 5 ▪ Common Criteria Part 3 Version 3.1 Revision 5 ▪ Common Methodology for Information Technology Security Evaluation (CEM) version 3.1 Revision 5

Table 1 ST and TOE References

1.2 TOE OVERVIEW

Embedded Firmware Security Solution of Connectivity Features v1.0 for Arçelik Bluetooth IoT Devices (hereinafter TOE) is an IoT device security solution which provides security functions to implement secure OTA firmware update of Arcelik IoT Devices mainboard and secure log storage of Arcelik IoT Devices.

1.2.1 TOE USAGE AND SECURITY FEATURE

The TOE provides secure OTA firmware update feature to the device users. The user easily updates the device firmware by following the procedure demonstrated on the mobile application. During the OTA firmware update process the download and install phases are protected by several cryptographic processes which are stated below.

Also, the device periodically logs usage data like diagnosis, customer detailed usage, electrical and sensor data of the device etc. The Secure Log Storage feature provides the log data to be stored and transmitted securely inside and outside (to Arcelik Cloud Server) of the product.

TOE is an Arcelik IoT Devices Security Solution that provides security functions in the form of library by being embedded on Arcelik IoT Devices electronic boards.

The TOE's purpose and key security features are as follows.

- ✓ Secure OTA Firmware Download

  This function blocks the installation of unauthorized firmware by using digital signature verification. The digital signature process for the firmware takes place in the Arcelik Cloud server and this enables only the firmware that are downloaded from the Arcelik Cloud server to be installable on the IoT device.

- ✓ Secure OTA Firmware Installation of Mainboard

  The new mainboard firmware downloaded on the Arcelik IoT device is encrypted and stored in the external flash of display board. The TOE decrypts and verifies the mainboard firmware image. After the verification is successfully completed, the installation process starts. The mainboard firmware image is securely transferred from display board to mainboard via encrypted SPI line chunk by chunk. The mainboard microcontroller decrypts each OTA image chunks and the mainboard's bootloader programs its own flash accordingly. The firmware update of mainboard finished after all required packages arrived in mainboard.

- ✓ Secure Log Storage

  The Arcelik IoT Devices periodically logs usage data like diagnosis, customer detailed usage, electrical and sensor data of the appliance etc. The Secure Log Storage function provides the log data to be stored and transmitted securely inside and outside (to Arcelik Cloud Server) of the product. The logged data are generated by mainboard. The log data are stored in display board's external flash until sent to Cloud Server. The Secure Log Storage Function uses the same secure SPI connection between DB and MB as stated in Secure OTA Firmware Installation phase. The communication between DB_MCU and external flash is also secure.

1.2.2 TOE TYPE

TOE is an embedded firmware that consists of “Secure OTA Download and Installation” and “Secure Usage Log Storage” security features.

1.2.3 FIRMWARE/HARDWARE/SOFTWARE REQUIRED BY TOE

The TOE operates in electronic boards of IoT Device. The device user must use the Arcelik HomeWhiz mobile application for utilizing connectivity features of IoT device and activating the TOE. The HomeWhiz application acts like a gateway between Arcelik IoT Device and Arcelik Cloud Server. In addition to requiring services from the environment to achieve its main goal, the environment (Arcelik Cloud Server) also maintains a secure posture so that the application cannot be compromised by factors out of the TSF Scope of Control.

Table 2 identifies the hardware required by TOE.

Category		**Specifications**
Display Board	MPU	48-MHz ARM Cortex-M0 256KB Flash 32KB SRAM
	Bluetooth	BLE 4.2
	Flash Memory	2MB SPI Flash
	HSM	ECC508
Mainboard	MPU	48-MHz ARM Cortex-M0+ 128KB Flash 16KB RAM
Communication Interface	Mainboard MPU - Display MPU	SPI
	Display MPU - Flash Memory	SPI
	Display MPU - HSM	I2C
Mobile App/Device	HomeWhiz	Requires BLE4.2 or later mobile device Requires Android 5.0/iOS 9 or later mobile device

Table 2 Hardware Required by TOE

1.2.4 TOE OPERATIONAL ENVIRONMENT

The external entities of the TOE operational environment can be divided into three group: Arcelik Cloud Server, mobile device (mobile application) and the appliance user.

Figure 1 shows the external entities of the TOE operational environment.

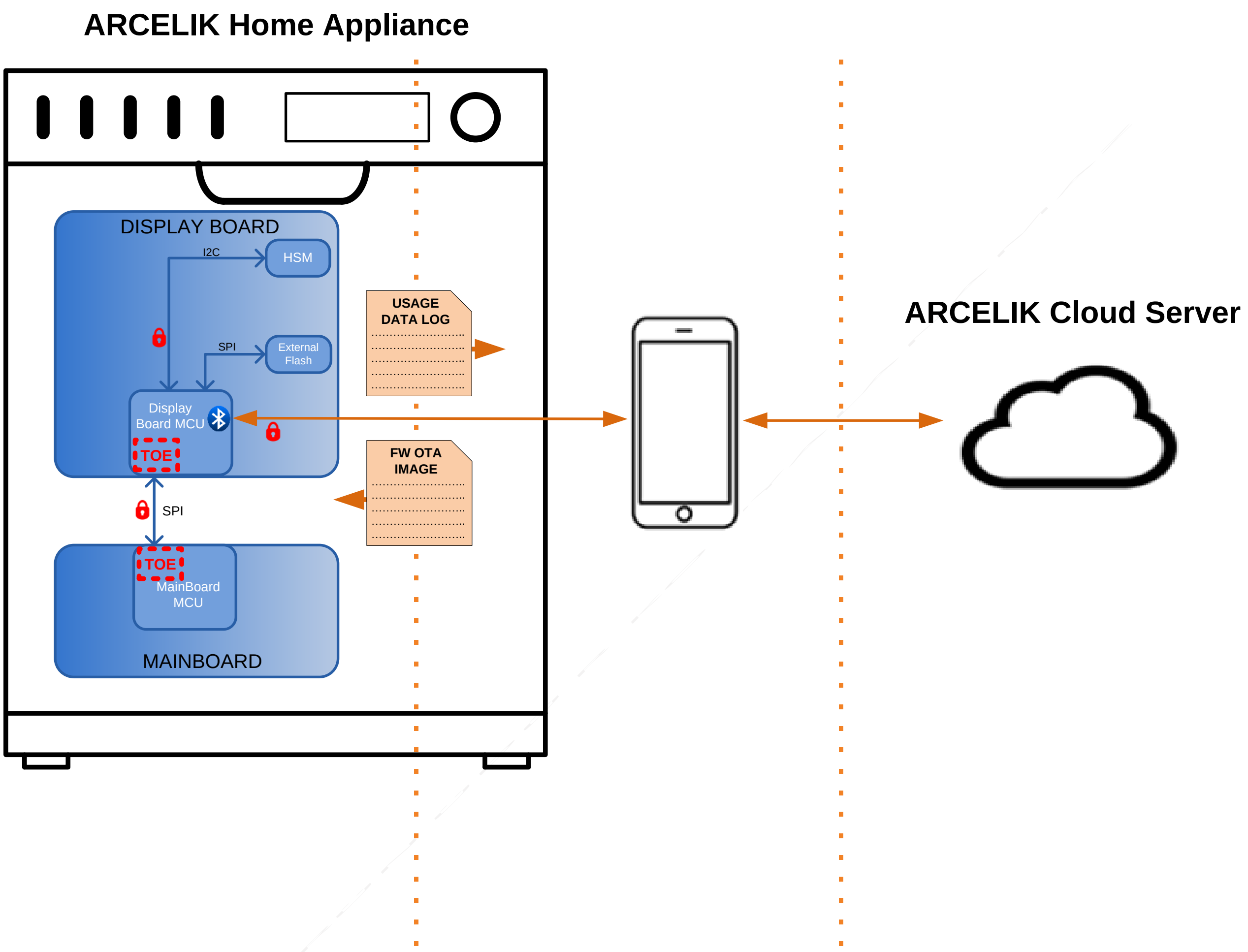

Figure 1 External Entities of the TOE Operational Environment

- ✓ Arcelik Cloud Server

Arcelik Cloud Server is implemented by Arcelik and running on AWS ec2 machines. It communicates with the mobile device via secure mqtt. It carries out an application layer TLS handshake process with the Arcelik IoT Device (exchanging the messages over mqtt) and transmits the “to be secured” messages as encrypted with that TLS session to make sure these messages are secured from the mobile device. Moreover, the usage log data are also encrypted with that TLS session and sent from the appliance to the cloud server. Mentioned TLS handshake process includes the mutual authentication of Arcelik IoT Device and Arcelik Cloud server, and that authentication is completed using public key certificates. (ECC certificate with prime256v1)

- ✓ Mobile Device

The mobile device establishes a link between an Arcelik IoT Device and the Arcelik Cloud Server. There is a mobile application (HomeWhiz) running on the mobile device. The mobile application acts like a gateway from security perspective. It transfers encrypted OTA and Log packages to appliance and cloud without knowing any valuable information like decryption key, private key etc.

- ✓ Appliance User

This refers to the user who uses an Arcelik IoT Device, connects it to the Arcelik Cloud Server using mobile application running on a mobile device. If necessary, upgrades the firmware of the appliance to take advantage of a variety of new features the appliance can provide. The users do not directly call the TOE but install new firmware to appliance and use mobile applications using IoT device functions.

1.3 TOE DESCRIPTION

This section describes the TOE's scope in terms of physical and logical scope of the TOE to describe the environment in which the TOE can be operated.

1.3.1 PHYSICAL SCOPE OF TOE

The physical scope of TOE includes software elements that are used for securing the OTA firmware update and implementing securely usage log storage. A figure of the TOE can be found in below (Figure 2) and identifies its components. Only authenticated and properly encrypted firmware images downloaded and installed to the product electronic boards. The usage log data is always stored and transferred encrypted inside the product. Also, the usage log data sent from product to server encrypted by using a secure authenticated communication channel.

The TOE has two firmware elements. Those are display board microcontroller firmware and mainboard microcontroller firmware. Those firmware's are installed to the electronic boards in the factory during the serial production phase of IoT devices. Upon completion of the production, the IoT device delivered and set up to the user's property by Arcelik Service Technician. Also, the Arcelik User Guide is delivered to the user during the delivery.

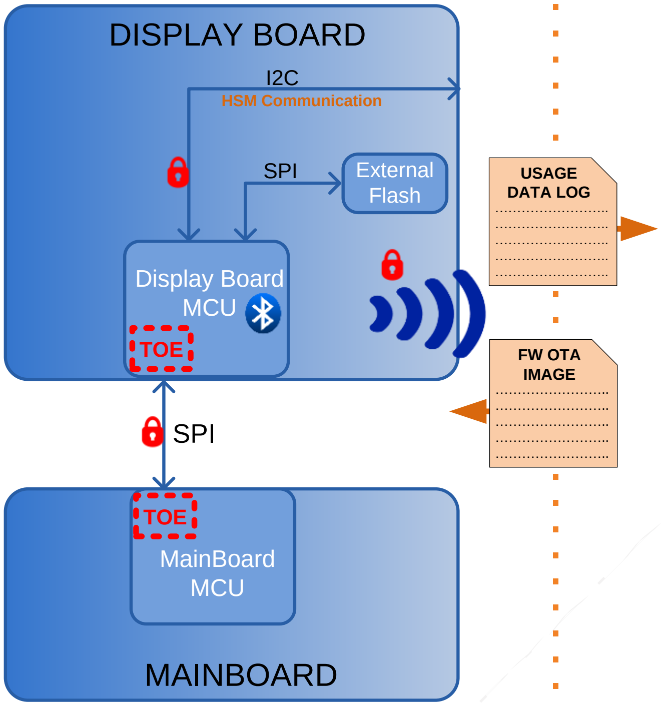

Figure 2 Infrastructure of TOE

1.3.2 LOGICAL SCOPE OF TOE

The logical scope of the TOE is described through the security functionality as follows;

- ✓ Secure OTA Firmware Download

  When a user connects a mobile device to Arcelik IoT Device via BLE, initially the firmware versions of the appliance and the latest firmware version published to Arcelik Cloud for respective appliance polled by the mobile device. After the comparison of firmware versions, the download request generated if needed.

  The new OTA firmware update image is placed on Arcelik Cloud Server. The image on Cloud Server is signed and encrypted. Before the download process starts, the Appliance to Arcelik Cloud authentication and digital signature verification must be fulfilled. Appliance to Arcelik Cloud authentication is established using TLS 1.2 protocol and digital signature material downloaded through TLS channel from Cloud Server to Appliance. After successful completion of authentication and verification processes, OTA firmware update image starts download from Cloud Server to Appliance. This means that apart from authenticated and verified OTA firmware update images download process are blocked by TOE.

- ✓ Secure OTA Firmware Installation of Mainboard

  The downloaded OTA firmware image on the Arcelik IoT Device is encrypted and stored in display board. Before passing to installation phase, the display board must implement verification and decryption to OTA image. The verification and decryption keys which was downloaded through TLS channel are used for fulfilling this step.

  After verifying and decrypting the OTA image, the sending and installation process starts.

  The DB transfers the OTA image through secure channel between DB and MB. The secure channel implemented by encryption and decryption of OTA image packages using DB-MB communication encryption key. After that the mainboard microcontroller gets the OTA package securely, the mainboard's bootloader programs its own flash accordingly. The firmware update of mainboard finished after all required packages arrived in mainboard.

- ✓ Secure Log Storage

  The Arcelik IoT Device periodically logs usage data like diagnosis, customer detailed usage, electrical and sensor data of the appliance etc. The Secure Log Storage function provides that the log data is stored and transmitted securely. The logged data are generated by mainboard, sent to display board and then sent to Cloud Server if all security conditions are fulfilled. The Secure Log Storage Function uses the same secure channels between DB-MB and Appliance-Cloud Server which is implemented in OTA download and installation phases. The log data is stored in DB until the connection occurs. If there is a secure connection between appliance and Cloud Server, the log data is sent over this channel.

2 CONFORMANCE CLAIM

2.1 CC CONFORMANCE CLAIM

This ST claims conformance to

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; CCMB-2012-09-001, Version 3.1, Revision 5, April 2017, [1]
- Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components; CCMB-2012-09-002, Version 3.1, Revision 5, April 2017, [2], *Conformant*
- Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; CCMB-2012-09-003, Version 3.1, Revision 5, April 2017, [3], *Conformant*

The

- Common Methodology for Information Technology Security Evaluation, Evaluation Methodology; CCMB-2012-09-004, Version 3.1, Revision 5, April 2017, [4]

has to be taken into account.

2.2 PP CLAIM

This ST does not claim conformance to any protection profile

2.3 PACKAGE CLAIM

Evaluation Assurance Level is EAL2-conformant.

3 SECURITY PROBLEM DEFINITION

This chapter defines the threats, OSPs (Organizational Security Policies) and assumptions which are intended to be addressed by the TOE and its operational environment.

The assets covered in the Connected Product are as follows.

- OTA Firmware image of mainboard.
- Keys for secure communication
- Usage log data (lifetime, detailed usage, electrical, sensor data, etc.)

3.1 THREATS

The threat agents are described below;

- Attackers who have knowledge of how the TOE operates and are assumed to possess a basic skill level, and intend to alter TOE configuration settings/parameters and no physical access to the TOE.

The TOE addresses the following threats are applicable listed in table below;

T.UnverifiedOtaDownload

Attacker could gain unauthorized access to the TOE data by bypassing the verification requirements and download OTA package to the TOE.

T.ModifyingOtaImage

Attacker may install a malicious OTA image by intercepting OTA image installation process which is sent over the network and modifying the OTA image data.

T.StealModifyUsageLogData

Attacker may steal or modify the usage log data as it is sent outside of the chip or to the Cloud Server.

T.UnauthorizedKeyAccess

Attacker may try to access to the cryptographic keys which are used in authentication, encryption/decryption and verification functions of TOE.

T.FirmwareCopyrightInfringement

Infringement of product firmware copyrights may occur by illegally copying the firmware contents like source codes and assets illegally.

3.2 ORGANIZATIONAL SECURITY POLICY

The organizational security policies are described in below,

P.FirmwareUpdateFileGenerationStorage

For ensuring the secure firmware update procedure, the firmware update image file must be generated and stored securely. The firmware update image file is encrypted using ChaCha20Poly1305 cryptography algorithm and digitally signed using ECDSA algorithm, so that it is protected during being transferred via network. The encrypted and signed firmware update image file is stored in Arcelik Cloud Server. Also, the verification and decryption material -named Key Transfer package- exists on Cloud Server. The KT is produced in Arcelik Secure Room at the end of signing and encryption process of new OTA firmware image. All firmware update images shall be signed and encrypted for the target appliance, it shall not be possible for an incorrectly signed and encrypted image to execute, for example firmware signed and encrypted for a different target appliance. When the OTA request come from end-user, the TOE step in to the process and provide Secure Firmware OTA to the products.

P.CloudSecureKeyManagement

The private key, which is used for authentication process of Arcelik Cloud Server and connected product, is generated by Arcelik based on the ECC standard, and stored securely on the Arcelik Cloud Server. The keys used for encrypting OTA firmware package is generated based on the ChaCha20Poly1305 standard by Arcelik Cloud Server and stored securely on the Arcelik Cloud Server. The generation, storage, access control, and destruction of the cryptographic keys, which are managed from the Arcelik Cloud Server are performed securely in accordance with Arcelik regulations and policies.

3.3 ASSUMPTIONS

The assumptions are described in below;

A.SecureCloudServer

For secure operation of TOE, The Arçelik cloud server which exists in the operating environment is operated securely.

A.ProductUniqueIDRegistration

Arcelik must assign unique ID's to each product to manage them securely. This ID used to identify the product uniquely in Arcelik Cloud Server.

A.ProductDisassemblyAuthorization

The user of the Arcelik IoT Device hasn't got authorization for disassemble the product and access the TOE physically.

4 SECURITY OBJECTIVES

This Security Target classifies security objectives into 2 groups: security objectives for the TOE and security objectives for the operational environment. The security objectives for the TOE are those that are directly handled by the TOE, and the security objectives for the operational environment are those that must be addressed through the technical and procedural measures which are supported by the operational environment for the TOE to provide security functionality.

4.1 SECURITY OBJECTIVES FOR TOE

The security objectives for the TOE are described in below;

O.OtaPackageVerification

The TOE verifies that the OTA firmware package to be installed on the product is an authorized package through digital signature verification and there is no unauthorized modification during downloading process.

O.OtaPackageContentsProtection

The TOE stores content files of downloaded OTA package in an encrypted form on the product, and before the OTA package is installed, the TOE verifies its integrity and decrypts the encrypted OTA package content files.

O.LogDataProtection

The TOE stores user log data encrypted and when it’s needed to transfer the log data inside the product, it transferred through secured channels.

O.CryptographicKeyProtection

The TOE stores all private cryptographic keys in secure storage hardware element (HSM). The HSM prevents unauthorized physical access to the private keys stored in its slots. All private cryptographic keys may only be read by encrypted way.

4.2 SECURITY OBJECTIVES FOR OPERATIONAL ENVIRONMENT

OE.CloudSecureKeyManagement

The private key, which is used for authentication process of Arcelik Cloud Server and connected product, is generated by Arcelik based on the ECC standard, and stored securely on the Arcelik Cloud Server. The keys used for encrypting OTA firmware package is generated based on the ChaCha20Poly1305 standard by Arcelik Cloud Server and stored securely on the Arcelik Cloud Server. The generation, storage, access control, and destruction of the cryptographic keys, which are managed from the Arcelik Cloud Server are performed securely in accordance with Arcelik regulations and policies.

OE.FirmwareUpdateFileGenerationStorage

After shipment of connected products, the product firmware which includes the TOE is updated through mobile app(HomeWhiz) via OTA by end-user. The firmware update file (product firmware image) is encrypted using ChaCha20Poly1305 cryptography algorithm and digitally signed using ECDSA algorithm, stored in Arcelik Cloud Server. When the OTA request come from end-user, the TOE step in to the process and provide Secure Firmware OTA to the products.

OE.SecureCloudServer

For secure operation of the TOE, Arcelik Cloud Server which exists in the operating environment is operated securely.

OE.SecureCommunicationChannel

A secure communication channel is provided for communication between the Connected Product and the Arcelik Cloud Server based on TLS v1.2.

OE.ProductUniqueIDRegistration

Arcelik must assign unique ID’s to each product to manage them securely. This ID used to identify the product uniquely in Arcelik Cloud Server.

OE.ProductDisassemblyAuthorization

The user of the Arcelik IoT Device hasn’t got authorization for disassemble the product and access the electronic boards (TOE) physically. This information is given to the user in user manual of appliance.

4.3 SECURITY OBJECTIVE RATIONALE

The security objectives rationale demonstrates the following:

- Each threat, organizational security policies, and assumption is addressed by at least one security objective.

- Each security objective addresses at least one threat, organizational security policies, or assumption.

The following table demonstrates that all security objectives trace back to the threats, OSPs and assumptions in the security problem definition.

		THREATS					OSPs		ASSUMPTIONS		
		T.UnverifiedOtaDownload	T.ModifyingOtaImage	T.StealModifyUsageLogData	T.UnauthorizedKeyAccess	T.FirmwareCopyrightInfringement	P.FirmwareUpdateFileGenerationStorage	P.CloudSecureKeyManagement	A.SecureCloudServer	A.ProductUniqueIDRegistration	A.ProductDisassemblyAuthorization
SECURITY OBJECTIVES FOR TOE	O.OtaPackageVerification	X									
	O.OtaPackageContentsProtection		X			X					
	O.LogDataProtection			X							
	O.CryptographicKeyProtection				X						
SECURITY OBJECTIVES FOR OPERATIONAL ENVIRONMENT	OE.CloudSecureKeyManagement							X			
	OE.FirmwareUpdateFileGenerationStorage						X				
	OE.SecureCloudServer								X		
	OE.SecureCommunicationChannel		X								
	OE.ProductUniqueIDRegistration									X	
	OE.ProductDisassemblyAuthorization										X

Table 3 Security Objective Rationale

O.OtaPackageVerification

This security objective ensures that only the authenticated OTA package downloaded to the product via digital signature verification by the TOE. This security objective enables preventing the threat T.UnverifiedOtaDownload.

O.OtaPackageContentsProtection

This security objective addresses the threat T.FirmwareCopyrightInfringement by preventing unauthorized use such as illegal copying of product firmware source codes, etc. by encrypting files of the OTA package downloaded on the product. As it also addresses T.ModifyingOtaImage by checking the integrity of the downloaded OTA package. By means of this, corrupted or modified OTA packages detected.

O.LogDataProtection

This security objective enables preventing the threat T.StealModifyUsageLogData by storing the user log data encrypted and when its needed to transfer, it transfers encrypted data through secured channels.

O.CryptographicKeyProtection

This security objective addresses the threat T.UnauthorizedKeyAccess by stored all private cryptographic keys in HSM. Unauthorized physical and open text access to the private keys blocked by means of HSM. All private cryptographic keys can only be read by encrypted way.

OE.CloudSecureKeyManagement

This security objective for operational environment executes OSP (organization security policy), P.CloudSecureKeyManagement by performing following. The private key, which is used for authentication process of Arcelik Cloud Server and connected product, is generated by Arcelik based on the ECC standard, and stored securely on the Arcelik Cloud Server. The keys used for encrypting OTA firmware package is generated based on the ChaCha20Poly1305 standard by Arcelik Cloud Server and stored securely on the Arcelik Cloud Server. The generation, storage, access control, and destruction of the cryptographic keys, which are managed from the Arcelik Cloud Server are performed securely in accordance with Arcelik regulations and policies.

OE.FirmwareUpdateFileGenerationStorage

This security objective for operational environment executes organizational security policy, P.FirmwareUpdateFileGenerationStorage by executing the following. After shipment of connected products, the product firmware which includes the TOE is updated through mobile app(HomeWhiz) via OTA by end-user. The firmware update file (product firmware image) is encrypted using ChaCha20Poly1305 cryptography algorithm and digitally signed using

ECDSA algorithm, stored in Arcelik Cloud Server. When the OTA request come from end-user, the TOE step in to the process and provide Secure Firmware OTA to the products.

OE.SecureCloudServer

This security objective for operational environment supports the assumption A.SecureCloudServer by executing the following. For secure operation of the TOE, Arcelik Cloud Server which exists in the operating environment is operated securely.

OE.SecureCommunicationChannel

This security objective for operational environment addresses Threat T.ModifyingOtaImage by executing the following. A secure communication channel is provided for communication between the Connected Product and the Arcelik Cloud Server based on TLS v1.2.

OE.ProductUniqueIDRegistration

This security objective for operational environment supports the assumption A.ProductUniqueIDRegistration by executing the following. Arcelik must assign unique ID's to each product in order to manage them securely. This ID used to identify the product uniquely in Arcelik Cloud Server.

OE.ProductDisassemblyAuthorization

This security objective for operational environment supports the assumption A.ProductDisassemblyAuthorization by executing the following. The user of the Arcelik IoT Device hasn't got authorization for disassemble the product and access the electronic boards (TOE) physically. This information is given to the user in user manual of appliance.

5 EXTENDED COMPONENT DEFINITION

This Security Target does not include any extended component.

6 SECURITY REQUIREMENTS

SFR Formatting

This section defines the security requirements satisfied by the TOE. Each requirement has been extracted from version 3.1 of the Common Criteria, part 2 providing functional requirements and part 3 providing assurance requirements.

Part 2 of the Common Criteria defines an approved set of operations that may be applied to security functional requirements. Following are the approved operations and the document conventions that are used within this ST to depict their application.

- **Assignment:** The assignment operation provides the ability to specify an identified parameter within a requirement. Assignments are depicted using ***italic and bolded text*** and are surrounded by square brackets as follows **[*assignment*]**.
- **Selection:** The selection operation allows the specification of one or more items from a list. Selections are depicted using *italics text* and are surrounded by square brackets as follows [*selection*].
- **Refinement:** The refinement operation allows the addition of extra detail to a requirement. Refinements are indicated using **bolded text**, for additions, and ~~strike-through~~, for deletions.
- **Iteration:** The iteration operation allows a component to be used more than once with varying operations. Iterations are depicted by an identifier at the end of the component identifier as follows FDP_ACC.1 – IDENTIFIER

6.1 SECURITY FUNCTIONAL REQUIREMENTS (SFR)

This section specifies the security functional requirements for the TOE. It organizes the SFRs by the CC classes.

Requirement Class	Requirement Component
FCS: Cryptographic Support	FCS_CKM.1: Cryptographic Key Generation – Session Key
	FCS_CKM.3(1): Cryptographic Key Access – CK
	FCS_CKM.3(2): Cryptographic Key Access – Auth Key/Trnsprt Key
	FCS_CKM.3(3): Cryptographic Key Access – Flash Encryption Key
	FCS_CKM.4(1): Cryptographic Key Destruction – CK
	FCS_CKM.4(2): Cryptographic Key Destruction – Session Key
	FCS_CKM.4(3): Cryptographic Key Destruction – OTAC_DEC_KEY
	FCS_CKM.4(4): Cryptographic Key Destruction – Flash Encryption Key
	FCS_COP.1(1): Cryptographic Operation – Enc./Dec. of Db-Mb Link
	FCS_COP.1(2): Cryptographic Operation – FW Dec.
	FCS_COP.1(3): Cryptographic Operation – Log Data Enc./Dec.
FDP: User Data Protection	FDP_ETC.1: Export of User Data Without Security Attributes – Auth Key/Trnsprt Key/Flash Encryption Key
	FDP_ETC.2(1): Export of User Data With Security Attributes – Cloud
	FDP_ETC.2(2): Export of User Data With Security Attributes – HSM
	FDP_IFC.1(1): Subset Information Flow Control – Cloud
	FDP_IFC.1(2): Subset Information Flow Control – HSM
	FDP_IFF.1(1): Simple Security Attributes – Cloud
	FDP_IFF.1(2): Simple Security Attributes – HSM
	FDP_ITC.1: Import of User Data Without Security Attributes
	FDP_ITC.2(1): Import of User Data With Security Attributes – KT over Cloud
	FDP_ITC.2(2): Import of User Data With Security Attributes – OTA over Cloud
	FDP_ITC.2(3): Import of User Data With Security Attributes – HSM
FMT: Security Management	FMT_MSA.3(1): Static Attribute Initialization – Cloud
	FMT_MSA.3(2): Static Attribute Initialization – HSM
FTP: Trusted Paths/Channels	FTP_ITC.1: Inter-TSF Trusted Channel
	FTP_TRP.1: Trusted Path

Table 4 Security Functional Requirements

6.1.1 Cryptographic Support (FCS)

FCS_CKM.1 Cryptographic Key Generation – Session Key

Hierarchical to: No other components.

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or
FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **[*HMAC and SHA-256*]** and specified cryptographic key sizes **[*32B*]** that meet the following: **[*FIPS 198a*]**.

FCS_CKM.3(1) Cryptographic Key Access – CK

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.3.1 The TSF shall perform **[*Encrypted Read of other Cryptographic Keys*]** in accordance with a specified cryptographic key access method **[*Internal Flash Read*]** that meets the following: **[*None*]**.

FCS_CKM.3(2) Cryptographic Key Access – Auth Key/Trsprt Key

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.3.1 The TSF shall perform **[*HSM Authorization and Transportation*]** in accordance with a specified cryptographic key access method **[*Internal Flash Read*]** that meets the following: **[*None*]**.

FCS_CKM.3(3) Cryptographic Key Access – Flash Encryption Key

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.3.1 The TSF shall perform **[*Storage Encryption/Decryption of Log Data*]** in accordance with a specified cryptographic key access method **[*Internal Flash Read*]** that meets the following: **[*None*]**.

FCS_CKM.4(1) Cryptographic Key Destruction – CK

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **[*Recycling The RAM*]** that meets the following: **[*None*]**.

FCS_CKM.4(2) Cryptographic Key Destruction – Session Key

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **[*Recycling the RAM, Erasing and Rewriting to the Same Location of RAM*]** that meets the following: **[*None*]**.

FCS_CKM.4(3) Cryptographic Key Destruction – OTAC_DEC_KEY

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **[*Recycling the RAM, Erasing and Rewriting the Same Location of RAM*]** that meets the following: **[*None*]**.

FCS_CKM.4(4) Cryptographic Key Destruction – Flash Encryption Key

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **[*Recycling the RAM*]** that meets the following: **[*None*]**.

FCS_COP.1(1) Cryptographic Operation – Enc./Dec. of Db-Mb Link

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or

FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1 The TSF shall perform **[*Encryption and Decryption of DB_MCU - MB_MCU Data Link*]** in accordance with a specified cryptographic algorithm **[*HMAC and SHA-256*]** and cryptographic key sizes **[*32B*]** that meet the following: **[*FIPS 198a*]**.

FCS_COP.1(2) Cryptographic Operation – FW Dec

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or

FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1 The TSF shall perform **[*OTA firmware Image Decryption*]** in accordance with a specified cryptographic algorithm **[*ChaCha20Poly1305*]** and cryptographic key sizes **[*32B*]** that meet the following: **[*FIPS 180-4*]**.

FCS_COP.1(3) Cryptographic Operation – Log Data Enc./Dec.

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or

FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1 The TSF shall perform **[*storage encryption and decryption of log data*]** in accordance with a specified cryptographic algorithm **[*Chacha20poly1305*]** and cryptographic key sizes **[*32B*]** that meet the following: **[*FIPS 180-4*]**.

6.1.2 User Data Protection (FDP)

FDP_ETC.1 Export of user data without security attributes – Auth Key/Trnsprt Key/Flash Encryption Key

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]

~~FDP_ETC.1.1 The TSF shall enforce the **[assignment: access control SFP(s) and/or information flow control SFP(s)]** when exporting user data, controlled under the SFP(s), outside of the TOE.~~

FDP_ETC.1.2 The TSF shall export the user data without the user data's associated security attributes

FDP_ETC.2(1) Export of User Data with Security Attributes – Cloud

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]

FDP_ETC.2.1 The TSF shall enforce the **[*Cloud Flow Control SFP*]** when exporting user data, controlled under the SFP(s), outside of the TOE.

FDP_ETC.2.2 The TSF shall export the user data with the user data's associated security attributes.

FDP_ETC.2.3 The TSF shall ensure that the security attributes, when exported outside the TOE, are unambiguously associated with the exported user data.

FDP_ETC.2.4 The TSF shall enforce the following rules when user data is exported from the TOE: **[*None*]**.

FDP_ETC.2(2) Export of User Data with Security Attributes – HSM

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]

FDP_ETC.2.1 The TSF shall enforce the **[*HSM Flow Control SFP*]** when exporting user data, controlled under the SFP(s), outside of the TOE.

FDP_ETC.2.2 The TSF shall export the user data with the user data's associated security attributes.

FDP_ETC.2.3 The TSF shall ensure that the security attributes, when exported outside the TOE, are unambiguously associated with the exported user data.

FDP_ETC.2.4 The TSF shall enforce the following rules when user data is exported from the TOE: **[*None*]**.

FDP_IFC.1(1) Subset Information Flow Control – Cloud

Hierarchical to: No other components.

Dependencies: FDP_IFF.1 Simple security attributes

- FDP_IFC.1.1 The TSF shall enforce the **[*Cloud Flow Control SFP*]** on **[*Subject: Display Board, Information: OTA Image, Stored Usage Log Operation: Send, Receive*]**.

FDP_IFC.1(2) Subset Information Flow Control – HSM

Hierarchical to: No other components.

Dependencies: FDP_IFF.1 Simple security attributes

- FDP_IFC.1.1 The TSF shall enforce the **[*HSM Flow Control SFP*]** on **[*Subject: Display Board, Information: Cryptographic Keys Operation: Send, Receive].***

FDP_IFF.1(1) Simple Security Attributes – Cloud

Hierarchical to: No other components.

Dependencies: FDP_IFC.1 Subset information flow control

FMT_MSA.3 Static attribute initialisation

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

[*Subject: Display Board, Information: OTA Image, Stored Usage Log,*

***Subject security atribute: None, Information security attribute: Signature of FW OTA Image, Security Session Existence*]**.

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: **[*TSF shall permit Display Board to receive OTA Image via Bluetooth Data Communication from Cloud, if signature of FW OTA Image is verified successfully, TSF shall permit Display Board to send Stored Usage Log via Bluetooth Data Communication to Cloud, if Security Session exists*]**.

FDP_IFF.1.3 The TSF shall enforce the **[*None*]**.

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

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

FDP_IFF.1(2) Simple Security Attributes – HSM

Hierarchical to: No other components.

Dependencies: FDP_IFC.1 Subset information flow control

FMT_MSA.3 Static attribute initialisation

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

[*Subject: Display Board, Information: Cryptographic Keys,*

***Subject security attribute: None, Information security attribute: Authorization Key*]**.

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: **[*TSF shall permit Display Board to send or receive cryptographic keys to/from HSM, if the Authorization Key verified*]**.

FDP_IFF.1.3 The TSF shall enforce the **[*None*]**.

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

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

FDP_ITC.1 Import of User Data without Security Attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_MSA.3 Static attribute initialisation

~~FDP_ITC.1.1 The TSF shall enforce the **[*assignment: access control SFP(s) and/or information flow control SFP(s)*]** when importing user data, controlled under the SFP, from outside of the TOE.~~

FDP_ITC.1.2 The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.

~~FDP_ITC.1.3 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: **[*assignment: additional importation control rules*]**.~~

FDP_ITC.2(1) Import of User Data with Security Attributes – KT over Cloud

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
[FTP_ITC.1 Inter-TSF trusted channel, or
FTP_TRP.1 Trusted path]
FPT_TDC.1 Inter-TSF basic TSF data consistency

FDP_ITC.2.1 The TSF shall enforce the **[*Cloud Flow Control SFP*]** when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.2.2 The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3 The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

FDP_ITC.2.4 The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: **[*None*]**.

FDP_ITC.2(2) Import of User Data with Security Attributes – OTA over Cloud

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
[FTP_ITC.1 Inter-TSF trusted channel, or
FTP_TRP.1 Trusted path]
FPT_TDC.1 Inter-TSF basic TSF data consistency

FDP_ITC.2.1 The TSF shall enforce the **[*Cloud Flow Control SFP*]** when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.2.2 The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3 The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

FDP_ITC.2.4 The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: **[*None*]**.

FDP_ITC.2(3) Import of User Data with Security Attributes – HSM

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
[FTP_ITC.1 Inter-TSF trusted channel, or
FTP_TRP.1 Trusted path]
FPT_TDC.1 Inter-TSF basic TSF data consistency

FDP_ITC.2.1 The TSF shall enforce the **[*HSM Flow Control SFP*]** when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.2.2 The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3 The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

FDP_ITC.2.4 The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: **[*None*]**.

6.1.3 Security Management (FMT)

FMT_MSA.3(1) Static Attribute Initialization – Cloud

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MSA.3.1 The TSF shall enforce the **[*Cloud Flow Control SFP*]** to provide [*Restrictive*] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the **[*None*]** to specify alternative initial values to override the default values when an object or information is created.

FMT_MSA.3(2) Static Attribute Initialization – HSM

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

FMT_MSA.3.1 The TSF shall enforce the **[*HSM Flow Control SFP*]** to provide [*Restrictive*] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the **[*None*]** to specify alternative initial values to override the default values when an object or information is created.

6.1.4 Trusted Path/Channels (FTP)

FTP_ITC.1 Inter-TSF trusted channel

Hierarchical to: No other components.

Dependencies: No dependencies.

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

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

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for **[*X.509 Certificate, Ephemeral ECDH Key Pair*]**.

FTP_TRP.1 Trusted path

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_TRP.1.1 The TSF shall provide a communication path between itself and [*Remote*] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [*Modification, Disclosure*]

FTP_TRP.1.2 The TSF shall permit [*the TSF, Remote Users*] to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for [***[Secure OTA Firmware Download and Secure Log Storage]***].

6.2 SECURITY ASSURANCE REQUIREMENTS (SAR)

The TOE meets the security assurance requirements for EAL2. The following table is the summary for the requirements.

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

Table 6 Security Assurance Requirements

6.3 SECURITY REQUIREMENTS RATIONALE

Security Requirements Rationale demonstrates that the described security requirements are suitable to satisfy security objectives and, as a result, appropriate to address security problems.

6.3.1 SFR RATIONALE

SFR Dependency Rationale

The table below shows dependencies of security functional requirements.

No	SFR	Dependency	Dependency Met?

1	FCS_CKM.1	[FCS_CKM.2 or FCS_COP.1]
		FCS_CKM.4
			FCS_COP.1(1)
			FCS_CKM.4(2)

2	FCS_CKM.3(1)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	The related key(CK) was loaded to the TOE in the production phase. It wasn't imported or created. FCS_CKM.4(1)
4	FCS_CKM.3(2)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	FDP_ITC.1 The related keys(Auth/Trsprt) located in Db and Mb internal flash and they are continuously used by TOE. Thus they are not destructed.
5	FCS_CKM.3(3)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	FDP_ITC.1 FCS_CKM.4(4)
6	FCS_CKM.4(1)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	The related key(CK) was loaded to the TOE in the production phase. It wasn't imported or created.
7	FCS_CKM.4(2)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1

8	FCS_CKM.4(3)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FDP_ITC.2(1)
9	FCS_CKM.4(4)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FDP_ITC.2(3)
10	FCS_COP.1(1)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	FCS_CKM.1 FCS_CKM.4(2)

11	FCS_COP.1(2)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]
		FCS_CKM.4
			FDP_ITC.2(1)
			FCS_CKM.4(3)

12	FCS_COP.1(3)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]
		FCS_CKM.4
			FDP_ITC.2(3)
			FCS_CKM.4(4)

13	FDP_ETC.1	[FDP_ACC.1 or FDP_IFC.1]	NA. There isn't any SFP defined for exporting Auth Key/Trsprt Key and Flash Encryption Key to Internal Flash
14	FDP_ETC.2(1)	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.1(1)
15	FDP_ETC.2(2)	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.1(2)
16	FDP_IFC.1(1)	FDP_IFF.1	FDP_IFF.1(1)
17	FDP_IFC.1(2)	FDP_IFF.1	FDP_IFF.1(2)

18	FDP_IFF.1(1)	FDP_IFC.1
		FMT_MSA.3
			FDP_IFC.1(1)
			FMT_MSA.3(1)

19	FDP_IFF.1(2)	FDP_IFC.1
		FMT_MSA.3
			FDP_IFC.1(2)
			FMT_MSA.3(2)

20	FDP_ITC.1	[FDP_ACC.1 or FDP_IFC.1] FMT_MSA.3	NA. There isn't any SFP defined for importing Cryptographic Keys from internal flash to DB_MCU NA. There is no security attribute involve during Cryptographic Keys import from internal flash to DB_MCU
21	FDP_ITC.2(1)	[FDP_ACC.1 or FDP_IFC.1] [FTP_ITC.1 or FTP_TRP.1] FPT_TDC.1	FDP_IFC.1(1) FTR_TRP.1 The KT Package is verfied using HSM.
22	FDP_ITC.2(2)	[FDP_ACC.1 or FDP_IFC.1] [FTP_ITC.1 or FTP_TRP.1] FPT_TDC.1	FDP_IFC.1(1) FTR_TRP.1 The KT Package is verfied using HSM.
23	FDP_ITC.2(3)	[FDP_ACC.1 or FDP_IFC.1] [FTP_ITC.1 or FTP_TRP.1] FPT_TDC.1	FDP_IFC.1(2) FTP_ITC.1 The consistency check of data import from HSM is not needed because only the HSM's that configured properly can communicate with TOE. HSM configuration is done by 3rd party manufacturer secure infrastructure
24	FMT_MSA.3(1)	FMT_MSA.1 FMT_SMR.1	NA. Cloud Security Attributes are constant and unvariable. NA. TOE hasn't got security management roles
25	FMT_MSA.3(2)	FMT_MSA.1 FMT_SMR.1	NA. Cloud Security Attributes are constant and unvariable. NA. TOE hasn't got security management roles
26	FTP_ITC.1	-	-
27	FTP_TRP.1	-	-

Table 7 SFR Dependency Table

SFR- Objective Rationale

Rationale of security functional requirements demonstrates in the following table. Each TOE security objective has at least one security functional requirement corresponding to it. Each TOE security functional requirement corresponds back to at least one TOE security objectives.

		Security Objectives			
		O.OtaPackageVerification	O.OtaPackageContentsProtection	O.LogDataProtection	O.CryptographicKeyProtection
Security Functional Requirements	FCS_CKM.1		X	X	
	FCS_CKM.3(1)				X
	FCS_CKM.3(2)	X			X
	FCS_CKM.3(3)			X	
	FCS_CKM.4(1)				X
	FCS_CKM.4(2)		X	X	
	FCS_CKM.4(3)		X		
	FCS_CKM.4(4)			X	
	FCS_COP.1(1)		X	X	
	FCS_COP.1(2)		X		
	FCS_COP.1(3)			X	
	FDP_ETC.1				X
	FDP_ETC.2(1)			X	
	FDP_ETC.2(2)	X			
	FDP_IFC.1(1)	X	X	X	
	FDP_IFC.1(2)				X
	FDP_IFF.1(1)	X	X	X	
	FDP_IFF.1(2)				X
	FDP_ITC.1				X
	FDP_ITC.2(1)	X	X		
	FDP_ITC.2(2)		X		
	FDP_ITC.2(3)				X
	FMT_MSA.3(1)	X			
	FMT_MSA.3(2)				X
	FTP_ITC.1				X
	FTP_TRP.1	X	X		

Table 8 SFR - Objective Rationale Table

O.OtaPackageVerification

The Cloud Flow Control SFP corresponds with *FDP_IFF.1(1).* The TLS trusted path which is established between Cloud server and TOE meets by *FTP_TRP.1. FCS_CKM.3(2)* helps the objective by accessing the auth key which is used in establishing the TLS connection. *FDP_ETC.2(2)* helps the objective by exporting certificates with security attributes to HSM for sign and verification process which is used in establishing the TLS connection. Importing the KT Package from Cloud Server to TOE (using TLS trusted path) is provided by *FDP_ITC.2(1).* Cloud static attribute initialization corresponds with *FMT_MSA.3(1).* Subset information flow control between Cloud and TOE corresponds with *FDP_IFC.1(1).*

O.OtaPackageContentsProtection

The Cloud Flow Control SFP corresponds with *FDP_IFF.1(1).* The TLS trusted path which is established between Cloud server and TOE meets by *FTP_TRP.1.* Importing the KT Package from Cloud Server to TOE is provided by *FDP_ITC.2(1).* The content of KT package used for decryption process of OTA FW image. Importing the OTA FW image from Cloud Server to TOE is provided by *FDP_ITC.2(2). FCS_COP.1(1) ensures* Display Board-Mainboard secure communication by using the Session Key. The pre-shared keys(PSK) are used for generating the Session Key. The Session Key generation process done by *FCS_CKM.1.* The Session Key destruction process done by *FCS_CKM.4(2). FCS_COP.1(2)* helps the objective by decrypting the OTA FW image using OTAC_DEC_KEY. The OTAC_DEC_KEY which is used for decrypting the OTA image destructed by *FCS_CKM.4(3).* Subset information flow control between Cloud and TOE corresponds with *FDP_IFC.1(1).*

O.LogDataProtection

The device Log Data created in Mainboard and transferred to Display Board. *FCS_COP.1(1) ensures* Display Board-Mainboard secure communication by using the Session Key. The pre-shared keys(PSK) are used for generating the Session Key. The Session Key generation process done by *FCS_CKM.1.* The Session Key destruction process done by *FCS_CKM.4(2).* After the Log data transferred to Display Board, the Display Board transfer it to External Flash for storage purpose. Display Board-External Flash secure communication channel is implemented using the Flash Encryption Key. The Flash Encryption Key accessed from internal flash by *FCS_CKM.3(3). FCS_COP.1(3)* helps the objective by encrypting/decrypting the Log Data using Flash Encryption Key before transferring to External Flash. The Flash Encryption Key destructed by *FCS_CKM.4(4).* Subset information flow control between Cloud and TOE correspond with *FDP_IFC.1(1).* The Cloud Flow Control SFP corresponds with *FDP_IFF.1(1). FDP_ETC.2(1)* helps the objective by exporting Log Data with security attributes from TOE to Cloud Server.

O.CryptographicKeyProtection

The TOE uses several cryptographic keys and certificates for ensuring the security objectives. Due to secure storage of those cryptographic keys and certificates an HSM module exists in the system. During the initialization phase of TOE, a secure I2C communication channel established between HSM and Display Board (TOE). *FCS_CKM.3(1)* helps the objective by accessing the Common Key from internal flash. The Common Key used for accessing the Trsprt Key. The Common Key destructed by *FCS_CKM.4(1). FCS_CKM.3(2)* helps the objective by accessing the Trsprt Key. The bus between TOE to HSM is encrypted using the Trsprt Key. HSM static attribute initialization corresponds with *FMT_MSA.3(2).* The trusted I2C channel between TOE to HSM represented by *FTP_ITC.1*. Importing cryptographic keys and certificates from HSM to TOE is provided by *FDP_ITC.2(3).* After importing cryptographic keys, they are exported to internal flash with *FDP_ETC.1*. When the cryptographic keys needed to use, they are imported from internal flash with *FDP_ITC.1*. The HSM Flow Control SFP corresponds with *FDP_IFF.1(2).* Subset information flow control between HSM and TOE correspond with *FDP_IFC.1(2).* The pre-shared keys(PSK) are used for generating the Session Key. PSK’s are generated by HSM and unique for each device.

6.3.2 SAR RATIONALE

The chosen assurance level is appropriate with the threats defined for the environment. The threats that were chosen are consistent with attacker of low attack motivation, therefore EAL2 was chosen for this ST.

7 TOE SUMMARY SPECIFICATION

This section summarizes security functions provided by TOE in term of how they fulfill the related SFR's. The TOE security functions divided into "secure OTA firmware download", "secure OTA firmware Installation of Mainboard" and "Secure Log Storage".

7.1.1 Secure OTA Firmware Download

When a user connects a mobile device to Arcelik IoT Device via BLE, initially the firmware versions of the appliance and the latest firmware version published to Arcelik Cloud for respective appliance polled by the mobile device. After the comparison of firmware versions, the download process starts if needed. Appliance to Arcelik Cloud authentication is established using TLS 1.2 protocol. The new OTA firmware update image is placed on Arcelik Cloud Server. The image on Cloud Server is signed using ECDSA and encrypted using ChaCha20Poly1305. The verification and decryption material - named Key Transfer package- also exists on Cloud Server. The KT is produced in Arcelik Secure Room at the end of signing and encryption process of new OTA image. The KT (Key Transfer) package sent over TLS channel from Cloud to Appliance. The TOE verifies the KT package using ECDSA on HSM placed on Display board. If the verification is successfully completed, the encrypted OTA package download starts.

Functional Requirement Satisfied: *FCS_CKM.3(1), FCS_CKM.3(2), FCS_CKM.4(1), FDP_ETC.1, FDP_ETC.2(2), FDP_IFC.1(1), FDP_IFC.1(2), FDP_IFF.1(1), FDP_IFF.1(2), FDP_ITC.1, FDP_ITC.2(1), FDP_ITC.2(3), FMT_MSA.3(1), FMT_MSA.3(2), FTP_ITC.1, FTP_TRP.1*

7.1.2 Secure OTA Firmware Installation of Mainboard

The new mainboard firmware image downloaded on the Arcelik IoT Device is encrypted and it is stored in the external flash of display board. In order to verify the downloaded OTA image, the display board microcontroller reads and decrypts the OTA image from external flash chunk by chunk (128 byte per chunk). The decrypting/verifying keys included in Key Transfer Package are used for those processes. Those keys are located in KT package that downloaded from Cloud server during in download phase. The decrypted chunks of image are added into a running hash. Signing key, result of hash and signature which is send with KT is processed on DB_MCU and HSM. Key that is used for verification is in ECC key format. Key is produced on Arcelik Cloud Server and it is used to sign new OTA image.

After the verification is successfully completed, the installation process starts. The display board reads the mainboard firmware image from the external flash and decrypts it chunk by chunk by using decryption key included in KT. The DB_MCU re-decrypt the whole image chunk by chunk without verifying. This time it sends the chunks to the Main Board over secure SPI channel. The DB_MCU re-encrypt the package with using display-mainboard communication encryption key (SPI session key). That key will be generated by display board and mainboard handshake mechanism using HMAC and PRF. Each failure on connection or data sending process make the session key to be dropped out. The encrypted package transferred from display board to mainboard. After that the mainboard microcontroller decrypts the OTA package and the mainboard's bootloader programs its own flash accordingly. In order to do that decryption, mainboard will need the related key for decrypting the image. That key will already be generated by display board and mainboard handshake mechanism using HMAC and PRF. The firmware update of mainboard finished after all required packages arrived to mainboard.

Functional Requirement Satisfied: *FCS_CKM.1, FCS_CKM.3(1), FCS_CKM.3(2), FCS_CKM.4(1), FCS_CKM.4(2), FCS_CKM.4(3), FCS_COP.1(1), FCS_COP.1(2), FDP_ETC.1, FDP_IFC.1(1), FDP_IFC.1(2), FDP_IFF.1(1), FDP_IFF.1(2), FDP_ITC.1, FDP_ITC.2(1), FDP_ITC.2(2), FDP_ITC.2(3), FMT_MSA.3(2), FTP_ITC.1, FTP_TRP.1*

7.1.3 Secure Log Storage

The Arcelik IoT Device periodically logs usage data like diagnosis, customer detailed usage, electrical and sensor data of the appliance etc. The Secure Log Storage function provides that the log data is stored and is transmitted securely. The logged data are generated by mainboard. The Secure Log Storage function uses secure channels those are already established between the components of appliance and Cloud Server. The log data is stored until the connection occurs. If there is a secure connection between appliance and Cloud Server (via Mobile app), the log data is sent over this channel. The Secure Log Storage Function uses the same secure SPI connection between DB and MB. Then MB transfers the log data to DB_MCU. DB_MCU encrypts the log data which is decrypted on secure SPI channel's DB_MCU end. It writes them into external flash by using flash encryption key. In this process key that is used to encrypt the log data is read from HSM. This reading process is an encrypted read process. In this process, another key is used which is named transport key. This key is used in encryption of the link between HSM and DB_MCU. Every encrypted read operation uses this key. Transport key is written into HSM and DB_MCU while production process. When the DB_MCU sending process starts, DB_MCU reads the encrypted log data and sends it over the TLS to the Cloud Server.

Functional Requirement Satisfied: *FCS_CKM.1, FCS_CKM.3(1), FCS_CKM.3(2), FCS_CKM.3(3), FCS_CKM.4(1) FCS_CKM.4(2), FCS_CKM.4(4), FCS_COP.1(1), FCS_COP.1(3), FDP_ETC.1, FDP_ETC.2(1), FDP_IFC.1(1), FDP_IFC.1(2), FDP_IFF.1(1), FDP_IFF.1(2), FDP_ITC.1, FDP_ITC.2(3), FMT_MSA.3(2), FTP_ITC.1*