LG Key Management Solution V1.0 for LG Shield Security Target 1.1



Document History

Version	Description of Change	Date
V1.0	Initial Creation	April 20, 2026
V1.1	Applied to EOR-01	July 30, 2026

Contents

1 INTRODUCTION OF SECURITY TARGET 5

1.1 Security Target Reference 5
1.2 TOE Reference 5
1.3 TOE Overview 6
1.4 TOE Description 9
1.4.1 Physical scope of TOE 9
1.4.2 Logical Scope of TOE 11
1.5 Conventions 14
1.6 Terms and Definition 16

2 CONFORMANCE CLAIMS 19

2.1 CC Conformance Claim 19
2.2 Protection Profiles Claim 20
2.3 Packages Claim 20
2.4 Conformance Rationale 20

3 SECURITY PROBLEM DEFINITION 21

3.1 Threats 21
3.2 Organizational Security Policy (OSP) 21
3.3 Assumptions 22

4 SECURITY OBJECTIVES 24

4.1 Security Objectives for the TOE 24
4.2 Security Objectives for the Operational Environment 25
4.3 Security Objectives Rationale 26
4.3.1 Rationale for Security Objectives for the TOE 28
4.3.2 Rational for Security Objectives for the Operational Environment 29

5 EXTENDED COMPONENTS DEFINITION 30

6 SECURITY REQUIREMENTS 31

6.1 Security Functional Requirements 31

6.1.1 Security audit 31

6.1.2 Security management 32

6.1.3 User data protection 34

6.2 Security Assurance Requirements 35

6.2.1 Development 36

6.2.2 Guidance documents 39

6.2.3 Life-cycle support 41

6.2.4 Security Target evaluation 43

6.2.5 Tests 50

6.2.6 Vulnerability assessment 53

6.3 Security Requirements Rationale 55

6.3.1 Security Functional Requirements Rationale 55

6.3.2 Security Assurance Requirements Rationale 57

6.3.3 Dependencies Rationale 58

7 TOE SUMMARY SPECIFICATION 60

7.1 Key Access Control 60

1 Introduction of Security Target

1.1 Security Target Reference

This section provides reference information to identify the Security Target (ST) in the following table. ST is identified by their ST title and ST version.

ST Title	LG Key Management Solution V1.0 for LG Shield Security Target
ST Version	V1.1
Date	July 30, 2026
Creator	LG Electronics, Inc.
Common Evaluation Criteria	Common Criteria for Information Technology Security (CC:2022 Revision 1)
Evaluation Assurance Level	EAL2

1.2 TOE Reference

This section provides reference information to identify the TOE in the following table.

TOE	LG Key Management Solution V1.0 for LG Shield
Detailed Version of TOE	WO4S.r01
Developer	LG Electronics, Inc.

1.3 TOE Overview

LG Key Management Solution V1.0 for LG Shield (hereinafter referred to as the "TOE") is a software-based key management solution that delivers public key to apps or services that have requested the public keys on webOS-based Smart TVs, Hotel TVs, Smart Monitors, and Projectors (hereinafter referred to as "smart devices").

The TOE requests DB8 to store symmetric keys and private keys so that smart device apps or service can securely store the symmetric keys and private keys used for encryption or decryption, and for generating or verifying digital signatures. When an app or service that requested the storage of a symmetric key and private key requests a public key response, the TOE verifies the information of the app or service, requests the KeyMaster TA to generate the public key, and provides a function that returns the public key to the app or service. In addition, when a smart device developer requests the deletion of symmetric keys or private keys, the TOE requests DB8 to delete the symmetric keys or private keys stored in DB8. webOS is a Linux-based smart device operating system developed by LG, which enables control of and access to advanced functions of smart devices and connected devices through a graphical user interface (GUI).

When a request is made to store symmetric keys or private keys, the TOE delivers the owner information of the corresponding key to DB8. When apps or services requests a public key response, the TOE searches for the key using the key owner information and determines whether to return the public key. The owner information includes the App ID that requested key storage and the Key name information provided by the corresponding app or service.

Major Security Features Provided by the TOE:

- Key Access Control

  The TOE determines whether to respond to a key request based on access control rules. The access control rules include the App ID of the app or service that stores or requests a key,

and the Key name of each key.

When an app or service requests a public key from the TOE, the TOE checks whether the corresponding symmetric key and private key exist by using the App ID of the requesting app or service and the Key name. If the corresponding key exists, the TOE requests the KeyMaster TA in the TEE to decrypt the private key and generate a public key and then returns the generated public key to the requesting app or service. If the public key generation request or return fails, the TOE generates a log indicating the reason for the failure.

The TOE is provided to smart device developers in the form of software and is not responsible for all security features provided by the smart device. The TOE provides only the security feature defined above.

Figure 1 shows the operational environment of the TOE. This environment consists of the hardware presented in Table 1 and the software presented in Table 2. The software consists of the operating system on which the TOE runs (webOS26) and the apps or services that perform key requests to the TOE.

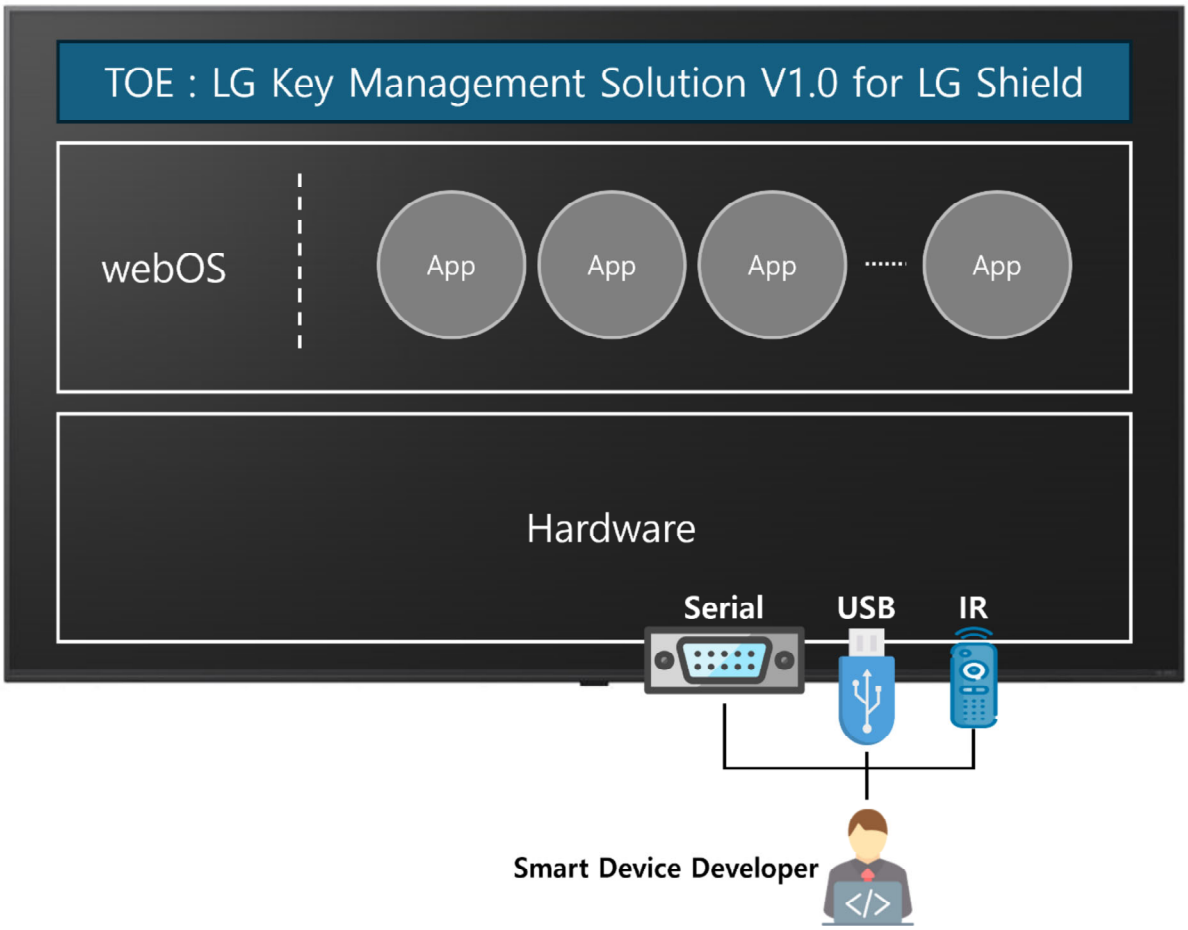


Figure 1. TOE Operational Environment

The non-TOE hardware and software required for the TOE to perform its functions are as follows:

Table 1. Non-TOE Hardware Required for the TOE

Category		**Minimum Specifications**
Smart Devices (Smart TV, Hotel TV, Smart Monitor, Projector)	CPU	ARM architecture (Cortex-A73 Quad) or higher
	DDR Memory	1.5GB or more
	Flash Memory	15MB or more (Required storage space for TOE installation)
	USB	USB 2.0 X 2
	Serial	RS-232 X 1
	IR Receiver	IR Receiver Module X 1
	Network	Ethernet / Wi-Fi

※ The serial interface is available only to smart device developers. Once smart device

development is completed, the serial interface is no longer available.

Table 2. Non-TOE Software Required for the TOE

Software	Description
webOS26	The operating system is based on Linux Kernel 6.12.105.
DB8	It is a storage service provided by webOS that stores and reads data in DB8 using LS2(webOS).

※ LS2 is included in the operating system (webOS26).

The TOE operates on, and is installed and executed in, a smart device developed based on webOS26. When an app or a service a symmetric key, a private key, and the identifier information for the corresponding key to the TOE using webOS(LS2), the TOE requests the KeyMaster TA in the TEE to encrypt the symmetric key and private key. The TOE then requests storage of the encrypted symmetric key and private key in DB8 using the App ID and Key name of the app or service that requested the storage.

A smart device developer must operate the TOE in an environment that supports both serial (RS-232) and USB interfaces. The developer can execute key storage and request commands and verify TOE reference information. To install the TOE on the smart device, the developer saves the TOE installation file distributed to a USB drive and connects it to the smart device. The developer also saves the key file to the USB, connects it, and transfers it to the smart device. Additionally, the developer can turn the power of the smart device on/off using an IR receiver.

1.4 TOE Description

1.4.1 Physical scope of TOE

The physical scope of the TOE includes the software and Developer Guidance.

The TOE, including the Developer Guidance, is delivered to smart device developers through a distribution site accessible only to LG Electronics smart device developers. In addition, the TOE is installed on the webOS operating system.

The physical scope of the TOE is limited to the software that requests DB8 to store symmetric keys and private keys, and that performs functions to return public keys and request deletion of symmetric keys and private keys according to requests from apps or services. Furthermore, the TOE does not provide all security features available in the smart device.

Figure 2 shows the physical scope of the TOE.

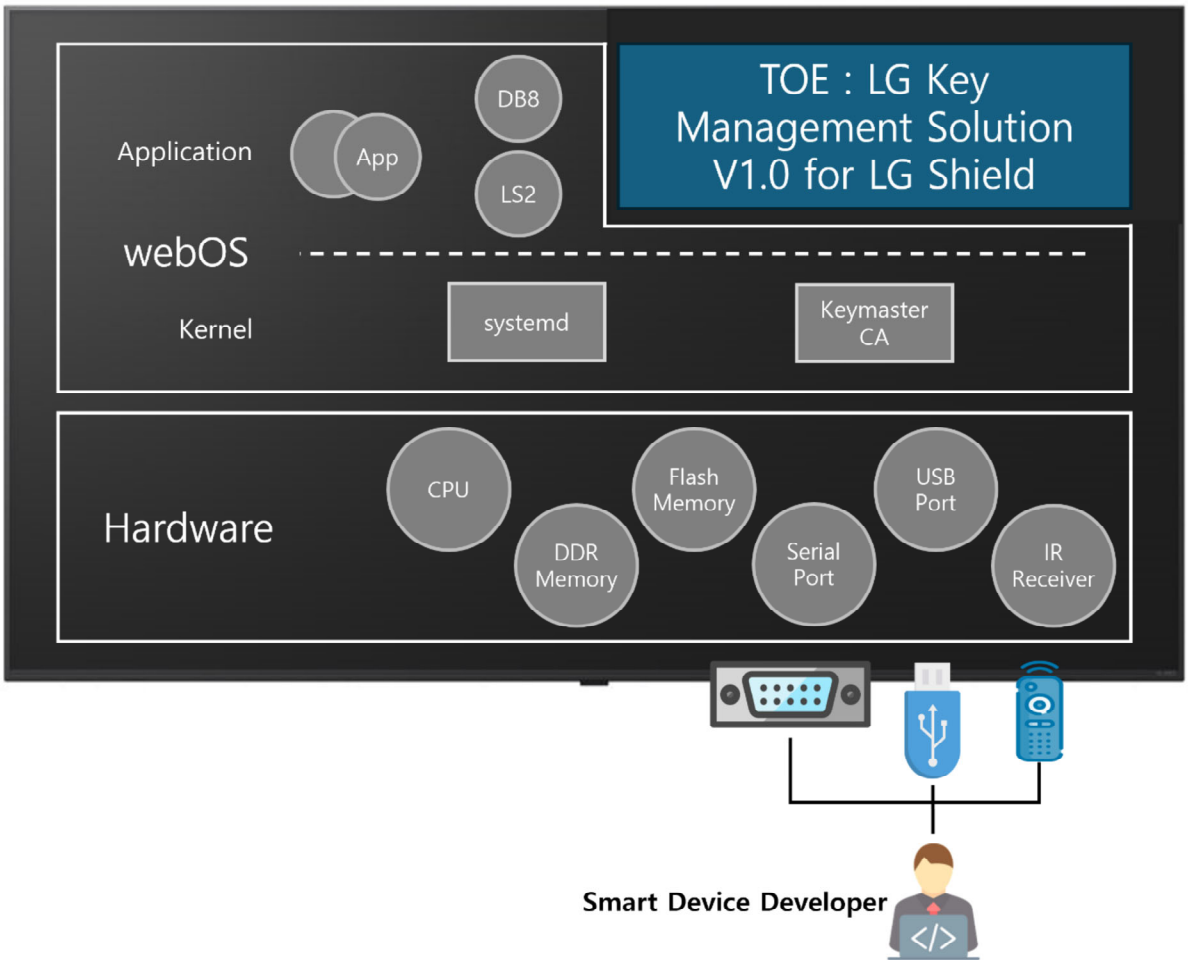


Figure 2. Physical scope of TOE

The following shows the physical scope of the TOE components.

Table 3. Physical scope of the TOE component

TOE Component	Description	Distribution Form	Distribution Method
Keymanager-1.0.WO4S.r01 (Keymanager-1.0.WO4S.r01.ipk)	Execution Binary of Keymanager	S/W	It is distributed in file form through a distribution site accessible only to LG Electronics smart device developers.
LG Key Management Solution V1.0 for LG Shield Developer Guidance V1.1 (LG Key Management Solution V1.0 for LG Shield Developer Guidance V1.1.pdf)	Developer Guidance	PDF File	

1.4.2 Logical Scope of TOE

The logical scope of the TOE consists of security features(key storage requests and key deletion requests, key access control) provided by the software included in the physical scope of the TOE. To accurately perform the key access control provided by the TOE, the TOE must support the features provided by webOS (LS2, DB8). webOS (LS2) is responsible for delivering commands entered by a smart device developer through the serial interface to the TOE, such as checking TOE reference information, requesting the storage/deletion of private keys and symmetric keys, and requesting the return of public keys. webOS (DB8) is responsible for storing the keys delivered through webOS (LS2).

When a power-on signal is input through the IR receiver, systemd automatically executes webOS (LS2), and webOS (LS2) is responsible for executing the TOE.

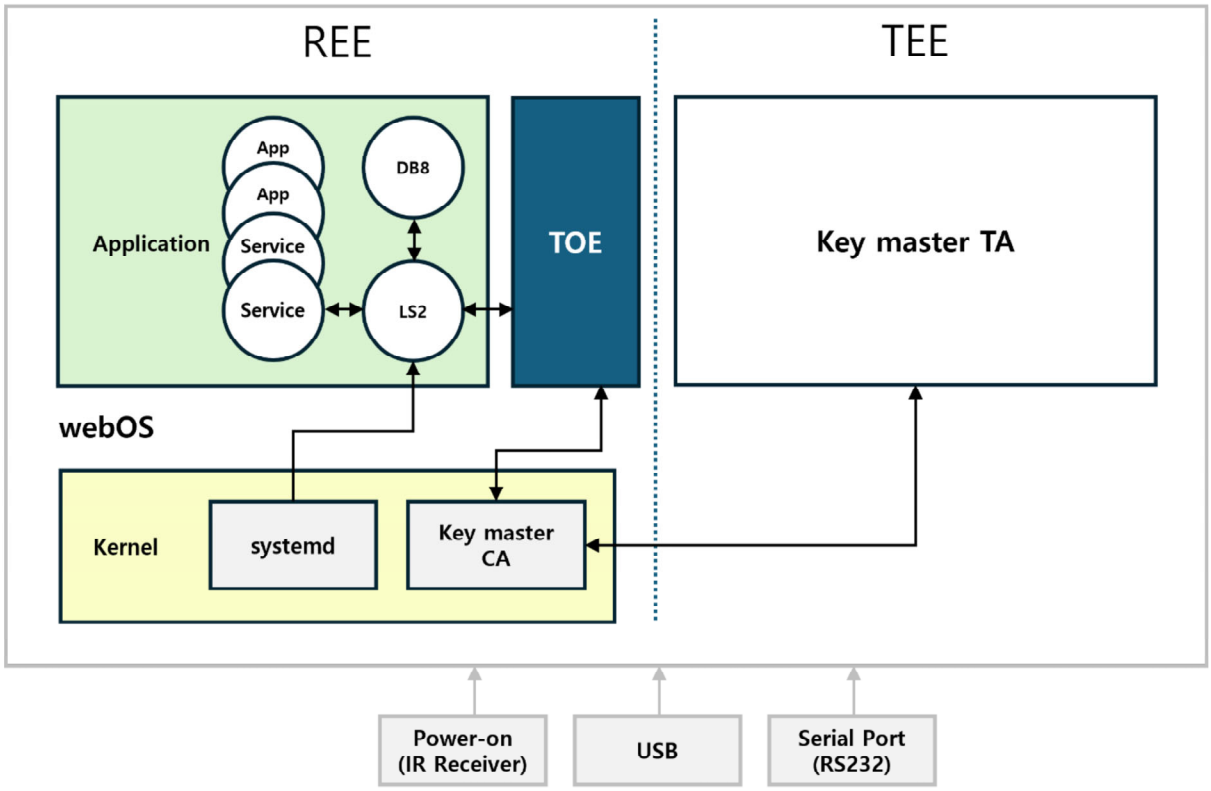


Figure 3. Logical Scope of TOE

The detailed security features included in the logical scope of the TOE are as follows:

Key Access Control

This feature determines whether to return a key in response to a key request.

To enable smart device apps or services to securely store symmetric keys and private keys used for encryption or decryption, and for digital signature generation or verification, the TOE requests DB8 to store the symmetric keys and private keys. When the apps or services that requested the storage of symmetric keys or private keys deliver a public key request command to the TOE using webOS (LS2), the TOE checks the App ID and Key name delivered in the command and searches webOS (DB8) for the private key that matches the corresponding information. If the comparison results match, the TOE requests the TEE's webOS (KeyMaster TA) to generate a public key and then returns the public key to the apps or services using webOS (LS2).

The smart device developer delivers a command requesting the deletion of related information, including the key stored in webOS (DB8), such as the App ID and Key name.

The TOE provides a function that allows the smart device developer to check TOE information. webOS (LS2) is responsible for delivering commands entered by the smart device developer through the serial interface, such as checking TOE reference information, to the TOE. When the smart device developer checks the TOE information, the TOE returns version information and detailed information.

The TOE generates logs and records them in the file system. Audit logs are generated using the timestamp of the webOS operating system and contain information on error types related to audit start and termination, storage/deletion of symmetric keys and private keys, and transmission/blocking of public keys.

1.5 Conventions

This section describes the conventions used to denote Common Criteria (CC) operations on security functional components and to distinguish text with special meaning. The notation, formatting, and conventions used in this ST are largely consistent with those used in the CC. Four presentation choices are discussed here.

Refinement

The refinement operation is used to add detail to a requirement, and, thus, further restricts a requirement. Refinement of security requirements is denoted by **bold text**.

Selection

The selection operation is used to select one or more options provided by the CC in stating a requirement. Selections are denoted by *underlined italicized text*.

Assignment

The assignment operation is used to assign a specific value to an unspecified parameter such as the length of a password. Showing the value in square brackets [assignment_value] indicates an assignment.

Iteration

Iteration is used when a component is repeated with varying operations. The result of iteration is marked with an iteration number in parenthesis following the component identifier, i.e., denoted as (iteration No.)

Application Note

Application Notes that introduced by the ST author have the detailed additional description.

1.6 Terms and Definition

App

App refers to application software installed on webOS that performs various functions, and apps are divided into native apps and web apps.

App ID

The App ID is a unique identifier assigned to a system or app. Apps or services installed on a smart device have a unique App id, and no duplicates are allowed.

DB8

DB8 is a storage system for webOS and provides a built-in JSON database that can be used through LS2.

Key name

It is the key name delivered to the TOE when an app or service requests storage of a symmetric key and a private key. The TOE checks the key name when the app or service requests a public key and then returns the public key. Each app or service has a unique key name, and no duplicates are allowed.

KeyMaster CA(Client Application)

The TOE serves as an Interface for using the KeyMaster TA.

KeyMaster TA(Trusted Application)

It provides encryption and decryption functions to protect symmetric keys and private keys. To store symmetric keys and private keys in DB8, it encrypts the symmetric keys and private keys delivered by the TOE and returns it to the TOE. When an app or service requests a public key, it decrypts the encrypted private key, generates a public key using the corresponding key, and returns it to the TOE.

LS2

LS2 refers to the APIs provided by webOS service daemons based on Luna Bus, the bus system used in webOS. Luna Bus and the corresponding APIs are implemented based on the Luna-Service2 library, and these are collectively referred to as LS2.

Smart Device

A smart device is a display-based electronic device that provides network connectivity and application execution functions. In this ST, smart devices are limited to Smart TVs, Hotel TVs, Smart Monitors, and Smart Projectors.

Smart Device Developer

A smart device developer refers to a developer who implements services that interact with the TOE during the software image development process for smart devices. The smart device developer performs installation, configuration, and testing of the TOE.

systemd

systemd is an initialization system that replaces the existing init process to bootstrap user space

and manages all processes within the system.

TSF

TOE Security Functionality

webOS

webOS is a Linux-based smart device operating system developed by LG Electronics. It is a software platform that provides an application development environment based on web standard technologies and supports content services, network connectivity, user interfaces, and control of device functions. webOS is applied to Smart TVs, Hotel TVs, Smart Monitors, Smart Projectors, and other devices. It also consists of major functions such as LS2 and DB8, as well as apps essential for operation.

2 Conformance Claims

This chapter describes how the Security Target conforms to the Common Criteria, Protection Profile and Package.

2.1 CC Conformance Claim

This Security Target conforms to the following Common Criteria.

Common Criteria Identification

Common Criteria for Information Technology Security Evaluation CC:2022 Revision 1

- Common Criteria for information Technology Security Evaluation, Part1: Introduction and general model, November 2022, CC:2022 Revision 1, CCMB-2022-11-001
- Common Criteria for Information Technology Security Evaluation, Part2: Security Functional components, November 2022, CC:2022 Revision 1, CCMB-2022-11-002
- Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, November 2022, CC:2022 Revision 1, CCMB-2022-11-003
- Common Criteria for Information Technology Security Evaluation, Part 4: Framework for the specification of evaluation methods and activities, November 2022, CC:2022 Revision 1, CCMB-2022-11-004
- Common Criteria for Information Technology Security Evaluation, Part 5: Pre-defined packages of security requirements, November 2022, CC:2022 Revision 1, CCMB-2022-11-005
- Errata and Interpretation for CC:2022 (Release 1) and CEM:2022 (Release 1), Version 1.2, CCMB-2025-001, October 2025

Conformance Claim

- Common Criteria for Information Technology Security Evaluation, Part 2 conformant

- Common Criteria for Information Technology Security Evaluation, Part 3 conformant.

2.2 Protection Profiles Claim

This Security Target does not claim conformance to PP.

2.3 Packages Claim

This Security Target claims conformance to assurance package EAL2.

2.4 Conformance Rationale

Since this Security Target does not declare that it conforms to other Protection Profiles, the conformance rationale is not provided.

3 Security Problem Definition

This chapter describes how the Security Target (ST) conforms to the Common Criteria (CC), Protection Profiles (PP), and packages.

The assets handled by smart devices are as follows.

Assets of webOS Smart Devices

Symmetric keys and private keys

Symmetric keys and private keys that the TOE requests to be stored for encryption and decryption, digital signatures, and public key generation are stored in text format.

3.1 Threats

A threat is an IT entity or user that can negatively affect assets through unauthorized access or abnormal ways and may cause the following threats. The threat agents against the TOE possess basic attack potential in terms of expertise, resources, and motivation.

T.UnauthorizedKeyAccess

A threat agent may attempt to access symmetric keys and private keys without proper authorization.

3.2 Organizational Security Policy (OSP)

The following describes the organizational security policies enforced by the TOE, the TOE operational environment, and both.

P.AuditLog

Security-related events shall be recorded to trace accountability for security-related actions.

3.3 Assumptions

The following describes the assumptions made about the TOE operational environment to provide security functions.

A.TrustedDeveloper

The TOE developers do not have malicious intent, have received appropriate training on the use of TOE, and properly perform their duties in accordance with the developer guidance. In addition, smart device developers should not intentionally provide LG Smart Device Services in a malicious manner.

A.webOSFunctionSupport

The system's reliability and security shall be ensured by addressing the latest vulnerabilities in webOS 26, on which the TOE is installed and operated. The TOE uses the functions provided by webOS 26. webOS 26 supports LS2, which delivers commands entered by Smart Device developers, such as requests to review TOE reference information. In addition, LS2 receives and processes requests to allow or block access to keys through the TOE.

A.KeymasterCAFunctionSupport

The TOE relies on the reliability of KeyMaster CA. The KeyMaster CA is responsible for forwarding key-related requests and keys received from the TOE to the KeyMaster TA, and for returning the processing results received from the KeyMaster TA back to the TOE.

A.KeymasterTAFunctionSupport

The TOE relies on the reliability of the KeyMaster TA. The KeyMaster TA encrypts, and decrypts symmetric keys and private keys received from the KeyMaster CA, or generates public keys using private keys received from the KeyMaster CA and returns them to the KeyMaster CA.

4 Security Objectives

This Security Target classifies the security objectives into two groups: security objectives for the TOE and security objectives for the operational environment.

The security objectives for the TOE refer to the security objectives directly performed by the TOE. The security objectives for the operational environment refer to the security objectives that shall be satisfied through technical and procedural measures supported by the operational environment so that the TOE can provide its security features.

4.1 Security Objectives for the TOE

The following are the security objectives directly performed by the TOE.

O.AuditLog

The TOE correctly generates security-related events to ensure accountability for security-related actions.

O.KeyAccessControl

The TOE determines whether an app or service is authorized when it requests access to symmetric keys and private keys. If the app or service is authorized, the TOE permits access to the symmetric keys and private keys.

The TOE denies access to symmetric keys and private keys from unauthorized apps or services.

4.2 Security Objectives for the Operational Environment

This section describes the security objectives that are satisfied through non-technical/procedural measures supported by the operational environment, enabling TOE to accurately provide its security features.

OE.TrustedDeveloper

The TOE developer does not have malicious intent, has completed appropriate training on the use of the TOE, and properly performs their duties in accordance with the developer guidance. In addition, the smart device developer should not intentionally provide LG smart device services in a malicious manner.

OE.webOSFunctionSupport

The system's reliability and security shall be ensured by addressing the latest vulnerabilities in webOS 26, on which the TOE is installed and operated. The TOE uses the functions provided by webOS 26, which supports LS2.

LS2 performs the role of delivering commands entered by the Smart Device developer, such as reviewing TOE reference information. In addition, it receives and executes requests to allow or block key access through TOE.

OE.TimeStamp

The TOE shall correctly generate security-related events using trusted timestamps provided by the TOE operational environment.

OE.KeymasterCAFunctionSupport

The TOE relies on the reliability of KeyMaster CA. The KeyMaster CA performs the role of forwarding key-related requests and keys received from the TOE to the KeyMaster TA, and returning the processing results received from the KeyMaster TA back to the TOE.

OE.KeymasterTAFunctionSupport

The TOE relies on the reliability of the KeyMaster TA. The KeyMaster TA encrypts/decrypts symmetric keys and private keys received from the keyMaster CA, or generates a public key using a private key received from KeyMaster CA and returns it to KeyMaster CA.

4.3 Security Objectives Rationale

The Security Objectives Rationale demonstrates that the defined security objectives are sufficient and appropriate to address the security problem, are not excessive, and are strictly necessary.

The Security Objectives Rationale demonstrates the following:

- Each threat, organizational security policy, and assumption is addressed by one or more security objectives.

- Each security objective addresses one or more threats, organizational security policies, or assumptions.

Table 4. Security problems and Security Objectives

	TOE Security Objectives	Security Objectives for Operational Environment

	O.KeyAccessControl	O.AuditLog	OE.TrustedDeveloper	OE.webOSFunctionSupport	OE.TimeStamp	OE.KeymasterCAFunctionSupport	OE.KeymasterTAFunctionSupport
T.UnauthorizedKeyAccess	X						
P.AuditLog		X			X		
A.TrustedDeveloper			X				
A.webOSFunctionSupport				X			
A.KeymasterCAFunctionSupport						X	
A.KeymasterTAFunctionSupport							X

4.3.1 Rationale for Security Objectives for the TOE

O.AuditLog

This security objective ensures that the TOE correctly generates audit logs so that accountability for security-related events can be traced. Therefore, this security objective is necessary to implement P.AuditLog, which is an organizational security policy.

O.KeyAccessControl

The TOE prevents unauthorized access to symmetric keys and private keys by determining whether an app or a service is authorized when it requests access to those keys. Therefore, this security objective is necessary to address T.UnauthorizedKeyAccess, which is a threat.

4.3.2 Rational for Security Objectives for the Operational Environment

OE.TrustedDeveloper

Smart device developers are not malicious, are appropriately trained to use the TOE, and must correctly perform their responsibilities in accordance with the developer guidance. Therefore, this security objective is necessary to support A.TrustDeveloper.

OE.webOSFunctionSupport

By supporting LS2 on webOS 26 where the TOE is installed, it ensures that the TOE can perform its security functions safely and reliably. Therefore, this security objective is necessary to support A.webOSFunctionSupport. In addition, it ensures the reliability and security of the operational environment by addressing the latest vulnerabilities in webOS 26 where the TOE is installed and operated.

OE.TimeStamp

It ensures that the TOE accurately records security-related events using trusted timestamps provided by the operational environment. Therefore, this security objective is necessary to implement P.AuditLog, the organizational security policy.

OE.KeymasterCAFunctionSupport

It supports A.KeymasterCAFunctionSupport by providing the KeyMaster CA used to deliver the key requested by the TOE to KeyMaster TA.

OE.KeymasterTAFunctionSupport

It is necessary to support A.KeymasterTAFunctionSupport by providing the KeyMaster TA within the TEE that securely performs the cryptographic operations required by the TOE.

5 Extended Components Definition

This Security Target does not include any extended components beyond those defined in Common Criteria (CC) Part 2 or Part 3.

6 Security Requirements

This chapter describes security functional requirements and security assurance requirements which should be satisfied in the TOE.

6.1 Security Functional Requirements

The security functional requirements defined in this Security Target are based on the functional requirements in Part 2 of the Common Criteria.

Following table shows the security functional requirements that are used in this Security Target document

Table 5. Security Functional Requirements

Security Functional Class	Security Functional Components	
Security Audit	FAU_GEN.1	Audit data generation
Security Management	FMT_MSA.1	Management of security attributes
	FMT_MSA.3	Static attribute initialization
	FMT_SMF.1	Specification of Management Functions
	FMT_SMR.1	Security roles
User Data Protection	FDP_IFC.1	Subset information flow control
	FDP_IFF.1	Simple security attributes

6.1.1 Security audit

FAU_GEN.1 Audit data generation

Component relationships

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;
b) All auditable events for the *not specified* level of audit;
c) [None]

FAU_GEN .1.2 The TSF shall record within the audit data at least the following information:

a) Date and time of the auditable event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event;
b) For each auditable event type, based on the auditable event definitions of the functional components included in the PP, PP-Module, functional package or ST, [None].

6.1.2 Security management

FMT_MSA.1 Management of security attributes

Component relationships

Hierarchical to: No other components

Dependencies: [FDP_ACC.1 Subset access control, or

FDP_IFC.1 Subset information flow control]

FMT_SMF.1 Specification of Management Functions

FMT_SMR.1 Security roles

FMT_MSA.1.1 The TSF shall enforce the [Key Access Control] to restrict the ability to *request storage and deletion* the security attributes [App ID, Key name, key] to [Smart Device Developer].

FMT_MSA.3 Static attribute initialization

Component relationships

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 [Key Access Control] to provide [*allow access only when the App ID and Key Name match*] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [Smart Device Developer] to specify alternative initial values to override the default values when an object or information is created.

FMT_SMF.1 Specification of Management Functions

Component relationships

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [query the TOE reference information, request storage and deletion of App ID and Key Name].

FMT_SMR.1 Security roles

Component relationships

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.1.1 The TSF shall maintain the roles [Smart Device Developer].

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

6.1.3 User data protection

FDP_IFC.1 Subset information flow control

Component relationships

Hierarchical to: No other components.

Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.1.1 The TSF shall enforce the [Key Access Control] on [the follow list of subjects, information, and operations].

- List of subjects : App
- List of information : App ID, Key name
- List of operations : deliver key to App

FDP_IFF.1 Simple security attributes

Component relationships

Hierarchical to: No other components.

Dependencies: FDP_IFC.1 Subset information flow control

FMT_MSA.3 Static attribute initialization

FDP_IFF.1.1 The TSF shall enforce the [Key Access Control] based on the following types of subject and information security attributes: [the follow list of subjects, information, Security attributes of subjects, Security attributes of informations].

- List of subjects : App

- List of information : App ID, Key name
- Security attributes of subjects : No other components
- Security attributes of information : App ID, Key name

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold : [allow access only when the App ID and Key Name match].

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

FDP_IFF.1.4 The TSF shall explicitly authorize 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].

6.2 Security Assurance Requirements

Security assurance requirements (SAR) defined in this document consists of assurance component in Common Criteria for Information Technology Security Evaluation, Part 3. The Evaluation Assurance Levels (EALs) is EAL2. The following table shows the summary of the security assurance requirements.

Table 6. Security Assurance 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

6.2.1 Development

ADV_ARC.1 **Security architecture description**

Dependencies: ADV_FSP.1 Basic functional specification

ADV_TDS.1 Basic design

Developer action elements:

ADV_ARC.1.1D The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2D The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.

ADV_ARC.1.3D The developer shall provide a security architecture description of the TSF.

Content and presentation elements:

ADV_ARC.1.1C The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2C The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

ADV_ARC.1.3C The security architecture description shall describe how the TSF initialisation process is secure.

ADV_ARC.1.4C The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5C The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

Evaluator action elements:

ADV_ARC.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.2 Security-enforcing functional specification

Dependencies: ADV_TDS.1 Basic design

Developer action elements:

ADV_FSP.2.1D The developer shall provide a functional specification.

ADV_FSP.2.2D The developer shall provide a tracing from the functional specification to the SFRs.

Content and presentation elements:

ADV_FSP.2.1C The functional specification shall completely represent the TSF.

ADV_FSP.2.2C The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.2.3C The functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.2.4C For each SFR-enforcing TSFI, the functional specification shall describe the SFR-

enforcing actions associated with the TSFI.

ADV_FSP.2.5C For each SFR-enforcing TSFI, the functional specification shall describe direct error messages resulting from processing associated with the SFR-enforcing actions.

ADV_FSP.2.6C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

Evaluator action elements:

ADV_FSP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.2.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

ADV_TDS.1 Basic design

Dependencies: ADV_FSP.2 Security-enforcing functional specification

Developer action elements:

ADV_TDS.1.1D The developer shall provide the design of the TOE.

ADV_TDS.1.2D The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.

Content and presentation elements:

ADV_TDS.1.1C The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.1.2C The design shall identify all subsystems of the TSF.

ADV_TDS.1.3C The design shall provide the behaviour summary of each SFR-supporting or SFR-non-interfering TSF subsystem.

ADV_TDS.1.4C The design shall summarise the SFR-enforcing behaviour of the SFR-enforcing subsystems.

ADV_TDS.1.5C The design shall provide a description of the interactions among SFR-enforcing subsystems of the TSF, and between the SFR-enforcing subsystems of the TSF and other subsystems of the TSF.

ADV_TDS.1.6C The mapping shall demonstrate that all TSFIs trace to the behavior described in the TOE design that they invoke.

Evaluator action elements:

ADV_TDS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_TDS.1.2E The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

6.2.2 Guidance documents

AGD_OPE.1 Operational user guidance

Dependencies: ADV_FSP.1 Basic functional specification

Developer action elements:

AGD_OPE.1.1D The developer shall provide operational user guidance.

Content and presentation elements:

AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available

functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security controls to be followed in order to fulfil the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

Evaluator action elements:

AGD_OPE.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1 Preparative procedures

Dependencies: No dependencies.

Developer action elements:

AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.

Content and presentation elements:

AGD_PRE.1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

Evaluator action elements:

AGD_PRE.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1.2E The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

6.2.3 Life-cycle support

ALC_CMC.2 Use of a CM system

Dependencies: ALC_CMS.1 TOE CM coverage

Developer action elements:

ALC_CMC.2.1D The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.2.2D The developer shall provide the CM documentation.

ALC_CMC.2.3D The developer shall use a CM system.

Content and presentation elements:

ALC_CMC.2.1C The TOE shall be labelled with its unique reference.

ALC_CMC.2.2C The CM documentation shall describe the method used to uniquely identify the configuration items.

ALC_CMC.2.3C The CM system shall uniquely identify all configuration items.

Evaluator action elements:

ALC_CMC.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_CMS.2 Parts of the TOE CM coverage

Dependencies: No dependencies.

Developer action elements:

ALC_CMS.2.1D The developer shall provide a configuration list for the TOE.

Content and presentation elements:

ALC_CMS.2.1C The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; and the parts that comprise the TOE.

ALC_CMS.2.2C The configuration list shall uniquely identify the configuration items.

ALC_CMS.2.3C For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.

Evaluator action elements:

ALC_CMS.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DEL.1 Delivery procedures

Dependencies: No dependencies.

Developer action elements:

ALC_DEL.1.1D The developer shall document and provide procedures for delivery of the TOE or parts of it to the downstream user.

ALC_DEL.1.2D The developer shall use the delivery procedures.

Content and presentation elements:

ALC_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the downstream user.

Evaluator action elements:

ALC_DEL.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.2.4 Security Target evaluation

ASE_INT.1 ST introduction

Dependencies: No dependencies.

Developer action elements:

ASE_INT.1.1D The developer shall provide an ST introduction.

Content and presentation elements:

ASE_INT.1.1C The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.

ASE_INT.1.2C The ST reference shall uniquely identify the ST.

ASE_INT.1.3C The TOE reference shall uniquely identify the TOE.

ASE_INT.1.4C The TOE overview shall summarize the usage and major security features of the TOE.

ASE_INT.1.5C The TOE overview shall identify the TOE type.

ASE_INT.1.6C The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.

ASE_INT.1.7C For a multi-assurance ST, the TOE overview shall describe the TSF organization in terms of the sub-TSFs defined in the PP-Configuration the ST claims conformance to.

ASE_INT.1.8C The TOE description shall describe the physical scope of the TOE.

ASE_INT.1.9C The TOE description shall describe the logical scope of the TOE.

Evaluator action elements:

ASE_INT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_INT.1.2E The evaluator shall confirm that the TOE reference, the TOE overview, and the TOE description are consistent with each other.

ASE_CCL.1 Conformance claims

Dependencies: ASE_INT.1 ST introduction

ASE_ECD.1 Extended components definition

ASE_REQ.1 Direct rationale security requirements

Developer action elements:

ASE_CCL.1.1D The developer shall provide a conformance claim.

ASE_CCL.1.2D The developer shall provide a conformance claim rationale.

Content and presentation elements:

ASE_CCL.1.1C The conformance claim shall identify the edition of the CC to which the ST and the TOE claim conformance.

ASE_CCL.1.2C The conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.

ASE_CCL.1.3C The conformance claim shall describe the conformance of the ST to CC Part 3 as

either CC Part 3 conformant or CC Part 3 extended.

ASE_CCL.1.4C The conformance claim shall be consistent with the extended components definition.

ASE_CCL.1.5C The conformance claim shall identify a PP-Configuration, or all PPs and security requirement packages to which the ST claims conformance.

ASE_CCL.1.6C The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.

ASE_CCL.1.7C The conformance claim shall describe any conformance of the ST to a PP as PP-Conformant.

ASE_CCL.1.8C The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PP-Configuration or PPs for which conformance is being claimed.

ASE_CCL.1.9C The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PP-Configuration, PPs and any functional packages for which conformance is being claimed.

ASE_CCL.1.10C The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PP-Configuration, PPs, and any functional package for which conformance is being claimed.

ASE_CCL.1.11C The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PP-Configuration, PPs, and any functional packages for which conformance is being claimed.

ASE_CCL.1.12C The conformance claim for PP(s) or a PP-Configuration shall be exact, strict, or demonstrable or a list of conformance types.

ASE_CCL.1.13C If the conformance claim identifies a set of Evaluation methods and Evaluation

activities derived from CEM work units that shall be used to evaluate the TOE then this set shall include all those that are included in any package, PP, or PP-Module in a PP-Configuration to which the ST claims conformance, and no others.

Evaluator action elements:

ASE_CCL.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_SPD.1 Security problem definition

Dependencies: No dependencies.

Developer action elements:

ASE_SPD.1.1D The developer shall provide a security problem definition.

Content and presentation elements:

ASE_SPD.1.1C The security problem definition shall describe the threats.

ASE_SPD.1.2C All threats shall be described in terms of a threat agent, an asset, and an adverse action.

ASE_SPD.1.3C The security problem definition shall describe the OSPs.

ASE_SPD.1.4C The security problem definition shall describe the assumptions about the operational environment of the TOE.

Evaluator action elements:

ASE_SPD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_OBJ.2 Security objectives

Dependencies: ASE_SPD.1 Security problem definition

Developer action elements:

ASE_OBJ.2.1D The developer shall provide a statement of security objectives.

ASE_OBJ.2.2D The developer shall provide a security objectives rationale.

Content and presentation elements:

ASE_OBJ.2.1C The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment.

ASE_OBJ.2.2C The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.

ASE_OBJ.2.3C The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.

ASE_OBJ.2.4C The security objectives rationale shall demonstrate that the security objectives counter all threats.

ASE_OBJ.2.5C The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.

ASE_OBJ.2.6C The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.

Evaluator action elements:

ASE_OBJ.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_ECD.1 Extended components definition

Dependencies: No dependencies.

Developer action elements:

ASE_ECD.1.1D The developer shall provide a statement of security requirements.

ASE_ECD.1.2D The developer shall provide an extended components definition.

Content and presentation elements:

ASE_ECD.1.1C The statement of security requirements shall identify all extended security requirements.

ASE_ECD.1.2C The extended components definition shall define an extended component for each extended security requirement.

ASE_ECD.1.3C The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.

ASE_ECD.1.4C The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.

ASE_ECD.1.5C The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements may be demonstrated.

Evaluator action elements:

ASE_ECD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_ECD.1.2E The evaluator shall confirm that no extended component may be clearly expressed using existing components.

ASE_REQ.2 Derived security requirements

Dependencies: ASE_OBJ.2 Security objectives

ASE_ECD.1 Extended components definition

Developer action elements:

ASE_REQ.2.1D The developer shall provide a statement of security requirements.

ASE_REQ.2.2D The developer shall provide a security requirements rationale.

Content and presentation elements:

ASE_REQ.2.1C The statement of security requirements shall describe the SFRs and the SARs.

ASE_REQ.2.2C For a single-assurance ST, the statement of security requirements shall define the global set of SARs that apply to the entire TOE. The sets of SARs shall be consistent with the PPs or PP-Configuration to which the ST claims conformance.

ASE_REQ.2.3C For a multi-assurance ST, the statement of security requirements shall define the global set of SARs that apply to the entire TOE and the sets of SARs that apply to each sub-TSF. The sets of SARs shall be consistent with the multi-assurance PP-Configuration to which the ST claims conformance.

ASE_REQ.2.4C All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.

ASE_REQ.2.5C The statement of security requirements shall identify all operations on the security requirements.

ASE_REQ.2.6C All operations shall be performed correctly.

ASE_REQ.2.7C Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.

ASE_REQ.2.8C The security requirements rationale shall trace each SFR back to the security objectives for the TOE. The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.

ASE_REQ.2.9C The security requirements rationale shall explain why the SARs were chosen.

ASE_REQ.2.10C The statement of security requirements shall be internally consistent.

ASE_REQ.2.11C If the ST defines sets of SARs that expand the sets of SARs of the PPs or PP-

Configuration it claims conformance to, the security requirements rationale shall include an assurance rationale that justifies the consistency of the extension and provides a rationale for the disposition of any Evaluation methods and Evaluation activities identified in the conformance statement that are affected by the extension of the sets of SARs.

ASE_REQ.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_TSS.1 TOE summary specification

Dependencies: ASE_INT.1 ST introduction

ASE_REQ.1 Direct rationale security requirements

ADV_FSP.1 Basic functional specification

Developer action elements:

ASE_TSS.1.1D The developer shall provide a TOE summary specification.

Content and presentation elements:

ASE_TSS.1.1C The TOE summary specification shall describe how the TOE meets each SFR.

Evaluator action elements:

ASE_TSS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_TSS.1.2E The evaluator shall confirm that the TOE summary specification is consistent with the TOE overview and the TOE description.

6.2.5 Tests

ATE_COV.1 Evidence of coverage

Dependencies: ADV_FSP.2 Security-enforcing functional specification

ATE_FUN.1 Functional testing

Developer action elements:

ATE_COV.1.1D The developer shall provide evidence of the test coverage.

Content and presentation elements:

ATE_COV.1.1C The evidence of the test coverage shall show the correspondence between the tests in the test documentation and the TSFIs in the functional specification.

Evaluator action elements:

ATE_COV.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_FUN.1 Functional testing

Dependencies: ATE_COV.1 Evidence of coverage

Developer action elements:

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation.

Content and presentation elements:

ATE_FUN.1.1C The test documentation shall consist of test plans, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.3C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.4C The actual test results shall be consistent with the expected test results.

Evaluator action elements:

ATE_FUN.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2 Independent testing - sample

Dependencies: ADV_FSP.2 Security-enforcing functional specification

AGD_OPE.1 Operational user guidance

AGD_PRE.1 Preparative procedures

ATE_COV.1 Evidence of coverage

ATE_FUN.1 Functional testing

Developer action elements:

ATE_IND.2.1D The developer shall provide the TOE for testing.

Content and presentation elements:

ATE_IND.2.1C The TOE shall be suitable for testing.

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

Evaluator action elements:

ATE_IND.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence

ATE_IND.2.2E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

ATE_IND.2.3E The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

6.2.6 Vulnerability assessment

AVA_VAN.2 Vulnerability analysis

Dependencies: ADV_ARC.1 Security architecture description

ADV_FSP.2 Security-enforcing functional specification

ADV_TDS.1 Basic design

AGD_OPE.1 Operational user guidance

AGD_PRE.1 Preparative procedures

Developer action elements:

AVA_VAN.2.1D The developer shall provide the TOE for testing.

AVA_VAN.2.2D The developer shall provide a list of third party components included in the TOE and the TOE delivery.

Content and presentation elements:

AVA_VAN.2.1C The TOE shall be suitable for testing.

AVA_VAN.2.2C The list of third party components shall include components provided by third parties, and that are part of the TOE or otherwise part of the TOE delivery.

Evaluator action elements:

AVA_VAN.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.2.2E The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE the components in the list of third party components, and specific IT products in the environment that the TOE depends

on.

AVA_VAN.2.3E The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design and security architecture description to identify potential vulnerabilities in the TOE.

AVA_VAN.2.4E The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

6.3 Security Requirements Rationale

The security requirements rationale demonstrates that the specified security requirements are suitable to meet the security objectives and, as a result, are appropriate for addressing the security problem.

6.3.1 Security Functional Requirements Rationale

The security functional requirements rationale requirements demonstrates that at least one security functional requirement is traced to each TOE security objective, and that each TOE security functional requirement is traced to at least one TOE security objective.

Table 7. Summary of Mappings between Security Objectives and Security Functional

Security Objectives / Security Functional Requirements	O.AuditLog	O.KeyAccessControl
FAU_GEN.1	X	
FMT_MSA.1		X
FMT_MSA.3		X
FMT_SMF.1		X
FMT_SMR.1		X
FDP_IFC.1		X
FDP_IFF.1		X

FAU_GEN.1 Audit Data Generation

This component satisfies O.AuditLog, the TOE security objective, by enabling TOE to generate audit logs.

FMT_MSA.1 Security Attribute Management

This component satisfies O.KeyAccessControl, the TOE security objective, because the ability to request storage and deletion of the security attributes, namely App ID and Key Name, is restricted to the Smart Device Developer.

FMT_MSA.3 Static Property Initialization

This component satisfies O.KeyAccessControl, the TOE security objective, because it sets the default values of the security attributes to permit access only when the App ID and Key Name match, to enforce the key access control policy through Key Access Control.

FMT_SMF.1 Specification of Management Functions

This component satisfies O.KeyAccessControl, the TOE security objective, because it performs the following management functions: querying the TOE version information and making storage and deletion requests for App ID and Key Name.

FMT_SMR.1 Security Roles

This component satisfies O.KeyAccessControl, the TOE objective, because it defines the Smart Device Developer as the role responsible for security

management.

FDP_IFC.1 **Patial Information Flow Control**

This component satisfies O.KeyAccessControl, the TOE security objective, because the TOE delivers only authorized keys to authorized apps.

FDP_IFF.1 **Single-Level Security Attributes**

This component satisfies O.KeyAccessControl, the TOE security objective, because it enforces information flow control based on the security attributes App ID and Key name, ensuring that the key is delivered to the app only when an authorized app requests the encrypted key.

6.3.2 Security Assurance Requirements Rationale

The security assurance level for this ST was selected as EAL2, considering the value of the assets protected by the TOE and the level of threats. EAL2 requires only the level of effort necessary for the developer to provide design information and test results using sound commercial development practices. EAL2 is applicable when complete development records are not readily available, and developers or users require a low to moderate level of independently assured security.

EAL2 provides assurance by analyzing the security functional requirements included in a complete ST, using the functional specification, interface specification, guidance documentation, and a basic description of the TOE structure to understand the security operation. This analysis demonstrates independent testing of the TSF, evidence of developer testing based on the functional

specification, independent confirmation of a sample of the developer's test results, and resistance to penetration attacks by an attacker with basic attack potential, based on the functional specification, TOE design, architectural design, and documentation evidence.

EAL2 also provides assurance through evidence of a configuration management system and secure delivery procedures.

In addition, AVA_VAN.2 vulnerability analysis is required to address potential vulnerabilities, and the evaluator performs penetration testing to determine that the potential vulnerabilities cannot be exploited in the TOE operational environment. The evaluator performs penetration testing, assuming a basic attack potential.

6.3.3 Dependencies Rationale

The dependencies rationale dependencies of both the SFRs and the SARs.

The following table shows the dependencies of the security functional requirements.

Table 8. Dependence on the TOE Security Functional Components

No.	Functional Components	Dependencies	Reference No.
1	FAU_GEN.1	FPT_STM.1 Trusted Timestamp	Rationale (OE.TimeStamp)
2	FMT_MSA.1	[FDP_ACC.1 Partial Access Control or FDP_IFC.1 Partial Information Flow Control] FMT_SMF.1Specification of Management Functions FMT_SMR.1 Security Roles	4, 5, 6
3	FMT_MSA.3	FMT_MSA.1 Security Attribute Management FMT_SMR.1 Security Roles	2, 5
4	FMT_SMF.1	N/A	N/A

5	FMT_SMR.1	FIA_UID.1 Identification	Rationale
6	FDP_IFC.1	FDP_IFF.1 Single-Level Security Attributes	7
7	FDP_IFF.1	FDP_IFC.1 Partial Information Flow Control FMT_MSA.3 Static Property Initialization	3, 6

FAU_GEN.1 has a dependency on FPT_STM.1. This dependency is satisfied through OE.TimeStamp, which provides reliable time stamps.

FMT_SMR.1 has a dependency on FIA_UID.1. Smart devices with embedded TOE security functions are generally owned by individuals, and all permissions, including user permissions and developer permissions, are granted to individual users. Therefore, identification or authentication is not provided.

The dependencies of the assurance packages provided by CC are also satisfied.

7 TOE Summary Specification

This chapter summarizes how the security functions provided by the TOE satisfy the security functional requirements. The following sections explain how each TOE security function satisfies the security functional requirements described in Chapter 6.

7.1 Key Access Control

When an app or service developed by a smart device developer requests a public key using webOS (LS2), the TOE requests the KeyMaster TA to generate the public key using the private key stored in DB8.After the generated public key is delivered to the TOE through webOS, the TOE returns it to the app or service using webOS (LS2).

To return the public key, the TOE receives the App ID and Key name of the corresponding app from webOS (LS2) and uses them to request a search for the private key stored in DB8.

If matching information exists, the TOE delivers the private key to the KeyMaster TA in the TEE through the KeyMaster CA in the REE, and the KeyMaster TA decrypts the private key.

The KeyMaster TA generates the public key and delivers it to the TOE through the KeyMaster CA.

The TOE then delivers the public key to the requesting app or service using webOS (LS2).

Related SFRs: FDP_IFC.1, FDP_IFF.1, FMT_MSA3

The TOE provides the Smart Device Developer with the ability to query TOE reference information.

When a Smart Device Developer checks the TOE reference information, the developer can send a command to the TOE through webOS (LS2) to check the TOE reference information. At this point, the TOE returns the reference information, including the TOE name and version.

Smart Device Developers use TOE to request the storage and deletion of symmetric keys and private keys used for the Key Access Control in DB8.

Related SFRs: FMT_MSA.1, FMT_SMF.1, FMT_SMR.1

The TOE generates audit records for the startup and shutdown of the audit function, storage/deletion of symmetric keys and private keys, and whether public keys are returned, using the timestamp of webOS.

When generating an audit record, the TOE records the date and time of the event, the type of auditable event, and the event result, whether success or failure.

Related SFRs: FAU_GEN.1