Common Criteria Protection Profile Configuration

Java Card System Multi-Assurance PP Configuration

Protection Profile Module for Firewall ADV_SPM

November 2025
Version 3.2M

Security Evaluations
Oracle Corporation
230 Oracle Way
Austin, TX 78741

Java Card System – Multi-Assurance PP Configuration and Firewall ADV_SPM PP Module
Version 3.2M

For any correspondence on this document please contact the following organisations:

- **Oracle Corporation,**
  2300 Oracle Way
  Austin, TX 78741,
  USA
  http://www.oracle.com
  seceval_us@oracle.com.

  Bundesamt für Sicherheit in der Informationstechnik
  Postfach 200363
  53133 Bonn, Germany
  https://www.bsi.bund.de/
  bsi@bsi.bund.de

TABLE OF CONTENTS

Tables

1 INTRODUCTION

This document consists of the following parts:

chapter 2 defines the Java Card System Multi-Assurance PP Configuration.

chapters 3-10 define the Protection Profile Module Firewall ADV_SPM.

2 JAVA CARD SYSTEM MULTI-ASSURANCE PP CONFIGURATION

2.1 REFERENCE

This PP-Configuration is identified as

Title: Java Card System Multi-Assurance PP Configuration

Version: 3.2M

Publication date: 13 November 2025

Registration: BSI-CC-PP-0123-2025

2.2 TOE OF THIS PP-CONFIGURATION

The TOE type in this PP-Configuration is the Java Card System (Java Card RE, Java Card VM and Java Card API) along with the additional native code embedded in a Smart Card Platform. The Java Card System is compliant with Java Card specifications versions 2.2.x or 3.x.x Classic Edition, including post-issuance installation facilities of applications verified off-card.

2.3 COMPONENTS STATEMENT

This PP-Configuration Java Card System Multi-Assurance PP Configuration has one single Base PP.

Title: Java Card System - Open Configuration Protection Profile

Version: 3.2

Publication date: July 2024

Registration: BSI-CC-PP-0099-V3-2024

This PP-Configuration consists of the Base PP together with the PP-Module:

Title: PP-Module for Firewall ADV_SPM

Version: 3.2M

Publication date: 13 November 2025

as described in chapters 3-10 of this document.

The organisation of the TSF is as follows: The base PP provides a sub-TSF of the entire Java Card System – Open Configuration Protection Profile (in that the base-PP can be used 'standalone' without a PP Configuration and PP-Module). The PP-Module for Firewall ADV_SPM provides a sub-TSF based on a sub-set around the base PP Firewall Module Security Objective. The SFRs of the PP-Module sub-TSF are duplicates of thirteen of the SFRs of the base PP.

2.4 CC CONFORMANCE CLAIMS FOR THE PP-CONFIGURATION

The PP-Configuration claims conformance to CC:2022 Revision 1.

This PP-Configuration is CC Part 2 conformant of Common Criteria 2022, Revision 1 [CC2].

This PP-Configuration is CC Part 3 conformant of Common Criteria 2022, Revision 1 [CC3].

The Common Methodology for Information Technology Security Evaluation, Evaluation Methodology 2022, Revision 1 [CEM] must be considered.

2.5 CONFORMANCE STATEMENT FOR THE PP-CONFIGURATION

The conformance to this PP-Configuration, required for the Security Targets claiming conformance to it, is demonstrable conformance, as defined in CC:2022 Part 1 [CC1].

2.6 CONFORMITY TO SECURITY ASSURANCE REQUIREMENTS

This PP-Configuration is a Multi-Assurance PP configuration, in that the conformity requirements of the Base PP are EAL4 augmented with AVA_VAN.5, ALC_DVS.2 and ALC_FLR.2, and the PP-Module conformity requirements are EAL4 augmented with AVA_VAN.5, ALC_DVS.2, ALC_FLR.2 and in addition ADV_SPM.1.

The global set of SARs are defined by the base PP, and are consistent with the threats as defined in the SPD of the base PP. The PP-Module does not define any additional threats to those defined in the base PP.

The global set of SARs as defined in the base PP are consistent with the additional SARs for the PP-Module, in that there is only one additional SAR for the PP module (ADV_SPM.1) that is applied only to the sub-TSF of the PP-Module. This application of ADV_SPM.1 to the PP-Module sub-TSF is the intent of the PP-Module and PP-Configuration.

3 PP-MODULE INTRODUCTION

The scope of this Protection Profile Module (PP-Module) is to describe a sub-set of the security functionality of the base PP based on the FIREWALL security objective, that can be evaluated at higher security assurance requirements, including ADV_SPM.1.

This PP-Module is intended for use with the following base PP:
Java Card System – Open Configuration Protection Profile, Version 3.2, See chapter 2 for the PP-Configuration.

This base PP is aligned and consistent with the PP-Module, as the PP-Module is entirely derived and dependent upon the existing functional and assurance requirements of this base PP. This PP-Module provides a method of evaluating security functional requirements that have already been evaluated in the base PP with the addition of ADV_SPM.

3.1 PP-MODULE REFERENCE

Title: Common Criteria Protection Profile Module for FIREWALL ADV_SPM

Version: 3.2M

Publication date: 13 November 2025

Certified by: Bundesamt für Sicherheit in der Informationstechnik

Registration: BSI-CC-PP-0123-2025

Sponsor: Oracle Corporation, 2300 Oracle Way, Austin, TX 78741, USA.

Review Committee: Java Card Forum – Common Criteria Subgroup

This PP-Module is conformant to the Common Criteria 2022 revision 1.

The minimum assurance level for this PP-Module is EAL 4 augmented with AVA_VAN.5 "Advanced methodical vulnerability analysis", ALC_DVS.2 "Sufficiency of security measures", ALC_FLR.2 "Flaw reporting procedures", and ADV_SPM.1 "Formal TOE Security Policy Model".

3.2 BASE PP IDENTIFICATION

This PP-Module requires the Java Card System - Open Configuration Protection Profile, version 3.2, July 2024, BSI-CC-PP-0099-V3-2024 to be selected as the base PP.

3.3 REFERENCES

Please refer to the base PP for references.

4 CONSISTENCY RATIONALE

This section analyses the consistency of the TOE type, the security problem definition (SPD), security objectives and security functional requirements (SFR) of the base PP with those of this PP-Module.

4.1 CONSISTENCY RATIONALE WITH BASE PP

4.1.1 TOE TYPE

The TOE type is exactly the same as the TOE type in the base PP. The TOE provides no additional services to those of the TSF in the base PP.

4.1.2 SECURITY PROBLEM DEFINITION

This PP-Module does not add any additional threats, organizational security policies or assumptions to those in the base PP Security Problem Definition.

4.1.3 SECURITY OBJECTIVES

This PP-Module adds the security objectives O.FIREWALL_MOD, O.GLOBAL_ARRAYS_INTEG_MOD, O.ARRAY_VIEWS_CONFID_MOD, O.ARRAY_VIEWS_INTEG_MOD and O.NATIVE_MOD which are reproductions of O.FIREWALL, O.GLOBAL_ARRAYS_INTEG, O.ARRAY_VIEWS_CONFID, O.ARRAY_VIEWS_INTEG and O.NATIVE respectively in the base PP to allow the additional Security Assurance Requirements to be applied to the SFRs in this PP-Module.

4.1.4 SECURITY FUNCTIONAL REQUIREMENTS

This PP-Module adds the following additional SFRs to those of the base PP:
FDP_ACC.2/FIREWALL_MOD, FDP_ACF.1/FIREWALL_MOD, FDP_IFC.1/JCVM_MOD,
FDP_IFF.1/JCVM_MOD,
FIA_UID.2/AID_MOD,
FMT_MSA.1/JCRE_MOD, FMT_MSA.1/JCVM_MOD, FMT_MSA.2/FIREWALL_JCVM_MOD,
FMT_MSA.3/FIREWALL_MOD, FMT_MSA.3/JCVM_MOD, FMT_SMF.1/MOD, FMT_SMR.1/MOD,
FMT_MTD.1/JCRE_MOD.

These SFRs are reproductions from the base PP to allow the additional Security Assurance Requirements to be applied to the SFRs in this PP-Module. The content of the SFRs are identical to those in the base PP (without the '_MOD' name suffix).

4.1.5 SECURITY ASSURANCE REQUIREMENTS.

This PP-Module adds the Security Assurance Requirement of ADV_SPM.1 "Formal TOE Security Policy Model" to be applied only to the PP-Module, resulting in a Multi-Assurance PP Configuration.

4.1.6 CONCLUSION.

In summary, the PP-Module adds the SAR ADV_SPM.1 to be applied to the SFRs within the PP-Module, which are identical to content to those in the base PP.

5 TOE OVERVIEW

This chapter defines the Target of Evaluation (TOE) type and describes the main security features of the TOE, the components of the TOE environment, the TOE life-cycle and TOE intended usage.

5.1 TOE TYPE

5.1.1 TOE OF THIS PP-MODULE

The TOE type in this PP-Module is the Java Card System (Java Card RE, Java Card VM and Java Card API) along with the additional native code embedded in a Smart Card Platform. The Java Card System is compliant with Java Card specifications versions 2.2.x or 3.x.x Classic Edition, including post-issuance installation facilities of applications verified off-card. Native code post-issuance downloading is out of the scope in this PP.

This TOE constitutes the target of the security requirements stated in this PP-Module, in addition to those of the base PP. The perimeter of the security requirements stated in this PP-Module are the same as those for the base PP, in that they do not define the perimeter of an actual evaluated product (TOE of the ST) that must include the Smart Card Platform.

5.1.2 TOE OF THE ST

The TOE type of the Security Target (ST) that declares conformity to this PP-Module and base PP is the Smart Card Platform (IC and OS) along with the native applications (if any), pre-issuance applets (if any) and the Java Card System. See base PP for additional Application note.

5.2 TOE SECURITY FUNCTIONS

Please refer to the base PP regarding TOE Security Functions. This PP-Module does not add any additional security functionality.

5.3 NON-TOE HW/SW/FW AVAILABLE TO THE TOE

Please refer to the base PP regarding TOE Security Functions. This PP-Module does not make any changes to this section and inherits entirely from the base PP.

5.4 TOE LIFE CYCLE

Please refer to the base PP regarding TOE Life Cycle. This PP-Module does not make any changes to this section and inherits entirely from the base PP.

5.5 TOE USAGE

Please refer to the base PP regarding TOE usage. This PP-Module does not make any changes to this section and inherits entirely from the base PP.

6 CONFORMANCE CLAIMS

6.1 CC CONFORMANCE CLAIMS FOR THE PP-MODULE

The PP-Module claims conformance to CC:2022 Revision 1.

This PP-Module is CC Part 2 conformant of Common Criteria 2022, Revision 1 [CC2].

This PP-Module is CC Part 3 conformant of Common Criteria 2022, Revision 1 [CC3].

The Common Methodology for Information Technology Security Evaluation, Evaluation methodology 2022, Revision 1 [CEM] must be considered.

6.2 CONFORMANCE CLAIM TO A PACKAGE

The minimum assurance level for this PP-Module is EAL4 augmented with AVA_VAN.5 "Advanced methodical vulnerability analysis", ALC_DVS.2 "Sufficiency of security measures", ALC_FLR.2 "Flaw reporting procedures" and ADV_SPM.1 "Formal TOE security policy model". This package is CC Part 5 conformant of Common Criteria 2022, Revision 1 [CC5] .

6.3 PP-MODULE CONFORMANCE CLAIMS

This PP-Module does not claim conformance to any other Protection Profile or PP-Module.

6.4 CONFORMANCE STATEMENT REGARDING CLAIMS TO THIS PP-MODULE

The conformance to this PP-Module, required for the Security Targets claiming conformance to it, inherited from the base PP, is demonstrable conformance, as defined in CC:2022 Part 1 [CC1].

7 SECURITY ASPECTS

Please refer to the base PP regarding Security aspects. This PP-Module does not make any changes to this section and inherits entirely from the base PP.

8 SECURITY PROBLEM DEFINITION

Please refer to the base PP regarding Security Problem Definition. This PP-Module does not make any changes to this section and does not add additional threats, assumptions or security objectives for the operating environment.

The additional Security Objectives defined in section 9 map to the threats defined in the base PP as detailed in Table 1 Threats and Security Objectives – Coverage, and Table 2 Security Objectives and Threats - Coverage in section 9.3.3.

The rationale between the Organisational Security Policies and Security objectives detailed in Section 6.3.2 and Table 3 of the base PP are not impacted or amended by this PP-Module.

The additional Security Objectives defined in section 9 do not map to Organisational Security Policies as shown in Table 3. As seen in both this table and table 4 of the base PP, only O.LOAD, OE.VERIFICATION and OE.CODE-EVIDENCE map to Organisational Security Policies, and these are not impacted or amended by this PP-Module.

Section 6.3.3 and Table 5 and 6 in the base PP provide the mapping and rationale between assumptions and security objectives for the operational environment. Since the PP-Module does not add assumptions or security objectives for the operating environment, these are not impacted or amended by this PP-Module.

9 SECURITY OBJECTIVES

9.1 SECURITY OBJECTIVES FOR THE TOE

This section defines the security objectives to be achieved by the TOE. The only Security Objectives within this PP-Module are O.FIREWALL_MOD, O.GLOBAL_ARRAYS_INTEG_MOD, O.ARRAY_VIEWS_CONFID_MOD, O.ARRAY_VIEWS_INTEG_MOD and O.NATIVE_MOD which are reproductions of O.FIREWALL, O.GLOBAL_ARRAYS_INTEG, O.ARRAY_VIEWS_CONFID, O.ARRAY_VIEWS_INTEG and O.NATIVE respectively in the base PP to allow the additional Security Assurance Requirements for the SFRs in this PP-Module. O.SID in the base PP is also referenced but not duplicated in this PP-Module.

9.1.1 IDENTIFICATION

O.SID is referenced from the base PP. See section 6.1.1 of the base PP.

9.1.2 EXECUTION

O.FIREWALL_MOD

The TOE shall ensure controlled sharing of data containers owned by applets of different CAP files or the JCRE and between applets and the TSFs. See #.FIREWALL for details in the base PP (section 4.3). O.FIREWALL_MOD has reduced functional requirements compared to O.FIREWALL in the base PP and are met by a smaller set of SFRs (the purpose of the PP Module being to allow those SFRs to be additional evaluated against ADV_SPM.1). O.FIREWALL_MOD should be considered in the context that O.FIREWALL in the base PP is already fully satisfied by the TOE.

O.FIREWALL_MOD focusses on the firewall mechanism core functionality as described in [JCRE3], section 6.1; specifically Applet separation and access control with regards to data, memories and operations according to the firewall rules.

O.GLOBAL_ARRAYS_INTEG_MOD

The TOE shall ensure that no application can store a reference to the APDU buffer, a global byte array created by the user through makeGlobalArray method and the byte array used for invocation of the install method of the selected applet. O. GLOBAL_ARRAYS_INTEG _MOD has the same functional requirements as O. GLOBAL_ARRAYS_INTEG in the base PP.

O.ARRAY_VIEWS_CONFID_MOD

The TOE shall ensure that no application can read elements of an array view not having array view security attribute ATTR_READABLE_VIEW.

The TOE shall ensure that an application can only read the elements of the array view within the bounds of the array view.

O.ARRAY_VIEWS_CONFID_MOD has the same functional requirements as O. ARRAY_VIEWS_CONFID in the base PP.

O.ARRAY_VIEWS_INTEG_MOD

The TOE shall ensure that no application can write to an array view not having array view security attribute ATTR_WRITABLE_VIEW.

The TOE shall ensure that an application can only write within the bounds of the array view.

O.ARRAY_VIEWS_ INTEG _MOD has the same functional requirements as O. ARRAY_VIEWS_ INTEG in the base PP.

O.NATIVE_MOD

The only means that the Java Card VM shall provide for an application to execute native code is the invocation of a method of the Java Card API, or any additional API. See #.NATIVE for details.

O.NATIVE_MOD has the same functional requirements as O.NATIVE in the base PP.

9.2 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT

Please refer to the base PP regarding Security Objectives for the Operational Environment. The PP-Module does not make any changes to this section and inherits entirely from the base PP.

9.3 SECURITY OBJECTIVES RATIONALE

9.3.1 THREATS

Please refer to the base PP regarding Threats. The PP-Module does not make any changes to this section and inherits entirely from the base PP.

9.3.2 ASSUMPTIONS

Please refer to the Base PP regarding Assumptions. The PP-Module does not make any changes to this section and inherits entirely from the base PP.

9.3.3 SPD AND SECURITY OBJECTIVES

Note that the following tables are identical to those in the base PP, but the additional Security Objectives **O.FIREWALL_MOD, O.GLOBAL_ARRAYS_INTEG_MOD, O.ARRAY_VIEWS_CONFID_MOD, O.ARRAY_VIEWS_INTEG_MOD** and **O.NATIVE_MOD** are also mapped here.

Threats	Security Objectives	base PP Rationale
T.CONFID-APPLI-DATA	OE.SCP.RECOVERY, OE.SCP.SUPPORT, OE.CARD-MANAGEMENT, OE.VERIFICATION, O.SID, O.OPERATE, O.FIREWALL, **O.FIREWALL_MOD**, O.GLOBAL_ARRAYS_CONFID, O.ARRAY_VIEWS_CONFID, **O.ARRAY_VIEWS_CONFID_MOD,** O.ALARM, O.TRANSACTION, O.CIPHER, O.RNG, O.PIN-MNGT, O.KEY-MNGT, O.REALLOCATION	Section 6.3.1
T.CONFID-JCS-CODE	OE.VERIFICATION, OE.CARD-MANAGEMENT, O.NATIVE, **O.NATIVE_MOD**	Section 6.3.1
T.CONFID-JCS-DATA	OE.SCP.RECOVERY, OE.SCP.SUPPORT, OE.CARD-MANAGEMENT, OE.VERIFICATION, O.SID, O.OPERATE, O.FIREWALL, **O.FIREWALL_MOD**, O.ALARM	Section 6.3.1
T.INTEG-APPLI-CODE	OE.CARD-MANAGEMENT, OE.VERIFICATION, O.NATIVE, **O.NATIVE_MOD**, OE.CODE-EVIDENCE	Section 6.3.1
T.INTEG-APPLI-CODE.LOAD	O.LOAD, OE.CARD-MANAGEMENT, OE.CODE-EVIDENCE	Section 6.3.1
T.INTEG-APPLI-DATA	OE.SCP.RECOVERY, OE.SCP.SUPPORT, OE.CARD-MANAGEMENT, OE.VERIFICATION, O.SID, O.OPERATE, O.FIREWALL, **O.FIREWALL_MOD**, O.GLOBAL_ARRAYS_INTEG, **O.GLOBAL_ARRAYS_INTEG_MOD**, O.ARRAY_VIEWS_INTEG, **O.ARRAY_VIEWS_INTEG_MOD,** O.ALARM, O.TRANSACTION, O.CIPHER, O.RNG, O.PIN-MNGT, O.KEY-MNGT, O.REALLOCATION, OE.CODE-EVIDENCE,	Section 6.3.1
T.INTEG-APPLI-DATA.LOAD	O.LOAD, OE.CARD-MANAGEMENT, OE.CODE-EVIDENCE	Section 6.3.1
T.INTEG-JCS-CODE	OE.CARD-MANAGEMENT, OE.VERIFICATION, O.NATIVE, **O.NATIVE_MOD**, OE.CODE-EVIDENCE	Section 6.3.1
T.INTEG-JCS-DATA	OE.SCP.RECOVERY, OE.SCP.SUPPORT, OE.CARD-MANAGEMENT, OE.VERIFICATION, O.SID, O.OPERATE, O.FIREWALL, **O.FIREWALL_MOD**, O.ALARM, OE.CODE-EVIDENCE	Section 6.3.1

T.SID.1	OE.CARD-MANAGEMENT, O.FIREWALL, **O.FIREWALL_MOD**, O.GLOBAL_ARRAYS_CONFID, O.GLOBAL_ARRAYS_INTEG, **O.GLOBAL_ARRAYS_INTEG_MOD**, O.INSTALL, O.SID	Section 6.3.1
T.SID.2	OE.SCP.RECOVERY, OE.SCP.SUPPORT, O.SID, O.OPERATE, O.FIREWALL, **O.FIREWALL_MOD**, O.INSTALL	Section 6.3.1
T.EXE-CODE.1	OE.VERIFICATION, O.FIREWALL, **O.FIREWALL_MOD**	Section 6.3.1
T.EXE-CODE.2	OE.VERIFICATION	Section 6.3.1
T.NATIVE	OE.VERIFICATION, OE.CAP_FILE , O.NATIVE, **O.NATIVE_MOD**	Section 6.3.1
T.RESOURCES	O.INSTALL, O.OPERATE, O.RESOURCES, OE.SCP.RECOVERY, OE.SCP.SUPPORT	Section 6.3.1
T.DELETION	O.DELETION, OE.CARD-MANAGEMENT	Section 6.3.1
T.INSTALL	O.INSTALL, O.LOAD, OE.CARD-MANAGEMENT	Section 6.3.1
T.OBJ-DELETION	O.OBJ-DELETION	Section 6.3.1
T.PHYSICAL	OE.SCP.IC	Section 6.3.1

Table 1 Threats and Security Objectives - Coverage

Security Objectives	Threats
O.SID	T.CONFID-APPLI-DATA, T.CONFID-JCS-DATA, T.INTEG-APPLI-DATA, T.INTEG-JCS-DATA, T.SID.1, T.SID.2
O.FIREWALL	T.CONFID-APPLI-DATA, T.CONFID-JCS-DATA, T.INTEG-APPLI-DATA, T.INTEG-JCS-DATA, T.SID.1, T.SID.2, T.EXE-CODE.1
O.FIREWALL_MOD	T.CONFID-APPLI-DATA, T.CONFID-JCS-DATA, T.INTEG-APPLI-DATA, T.INTEG-JCS-DATA, T.SID.1, T.SID.2, T.EXE-CODE.1
O.GLOBAL_ARRAYS_CONFID	T.CONFID-APPLI-DATA, T.SID.1
O.GLOBAL_ARRAYS_INTEG	T.INTEG-APPLI-DATA, T.SID.1
O.GLOBAL_ARRAYS_INTEG_MOD	T.INTEG-APPLI-DATA, T.SID.1
O.ARRAY_VIEWS_CONFID	T.CONFID-APPLI-DATA
O.ARRAY_VIEWS_CONFID_MOD	T.CONFID-APPLI-DATA
O.ARRAY_VIEWS_INTEG	T.INTEG-APPLI-DATA
O.ARRAY_VIEWS_INTEG_MOD	T.INTEG-APPLI-DATA
O.NATIVE	T.CONFID-JCS-CODE, T.INTEG-APPLI-CODE, T.INTEG-JCS-CODE, T.NATIVE
O.NATIVE_MOD	T.CONFID-JCS-CODE, T.INTEG-APPLI-CODE, T.INTEG-JCS-CODE, T.NATIVE
O.OPERATE	T.CONFID-APPLI-DATA, T.CONFID-JCS-DATA, T.INTEG-APPLI-DATA, T.INTEG-JCS-DATA, T.SID.2, T.RESOURCES
O.REALLOCATION	T.CONFID-APPLI-DATA, T.INTEG-APPLI-DATA
O.RESOURCES	T.RESOURCES
O.ALARM	T.CONFID-APPLI-DATA, T.CONFID-JCS-DATA, T.INTEG-APPLI-DATA, T.INTEG-JCS-DATA
O.CIPHER	T.CONFID-APPLI-DATA, T.INTEG-APPLI-DATA
O.RNG	T.CONFID-APPLI-DATA, T.INTEG-APPLI-DATA
O.KEY-MNGT	T.CONFID-APPLI-DATA, T.INTEG-APPLI-DATA
O.PIN-MNGT	T.CONFID-APPLI-DATA, T.INTEG-APPLI-DATA
O.TRANSACTION	T.CONFID-APPLI-DATA, T.INTEG-APPLI-DATA
O.OBJ-DELETION	T.OBJ-DELETION
O.DELETION	T.DELETION
O.LOAD	T.INTEG-APPLI-CODE.LOAD, T.INTEG-APPLI-DATA.LOAD, T.INSTALL
O.INSTALL	T.SID.1, T.SID.2, T.RESOURCES, T.INSTALL
OE.CAP_FILE	T.NATIVE

OE.CARD-MANAGEMENT	T.CONFID-APPLI-DATA, T.CONFID-JCS-CODE, T.CONFID-JCS-DATA, T.INTEG-APPLI-CODE, T.INTEG-APPLI-CODE.LOAD, T.INTEG-APPLI-DATA, T.INTEG-APPLI-DATA.LOAD, T.INTEG-JCS-CODE, T.INTEG-JCS-DATA, T.SID.1, T.DELETION, T.INSTALL
OE.SCP.IC	T.PHYSICAL
OE.SCP.RECOVERY	T.CONFID-APPLI-DATA, T.CONFID-JCS-DATA, T.INTEG-APPLI-DATA, T.INTEG-JCS-DATA, T.SID.2, T.RESOURCES
OE.SCP.SUPPORT	T.CONFID-APPLI-DATA, T.CONFID-JCS-DATA, T.INTEG-APPLI-DATA, T.INTEG-JCS-DATA, T.SID.2, T.RESOURCES
OE.VERIFICATION	T.CONFID-APPLI-DATA, T.CONFID-JCS-CODE, T.CONFID-JCS-DATA, T.INTEG-APPLI-CODE, T.INTEG-APPLI-DATA, T.INTEG-JCS-CODE, T.INTEG-JCS-DATA, T.EXE-CODE.1, T.EXE-CODE.2, T.NATIVE
OE.CODE-EVIDENCE	T.INTEG-APPLI-CODE, T.INTEG-APPLI-CODE.LOAD, T.INTEG-APPLI-DATA, T.INTEG-APPLI-DATA.LOAD, T.INTEG-JCS-CODE, T.INTEG-JCS-DATA

Table 2 Security Objectives and Threats - Coverage

Please refer to the base PP for tables regarding **OSPs and Security Objectives - Coverage** which are not affected by this PP-Module.

Security Objectives	Organisational Security Policies
O.SID	
O.FIREWALL	
O.FIREWALL_MOD	
O.GLOBAL_ARRAYS_CONFID	
O.GLOBAL_ARRAYS_INTEG	
O.GLOBAL_ARRAYS_INTEG_MOD	
O.ARRAY_VIEWS_CONFID	
O.ARRAY_VIEWS_CONFID_MOD	
O.ARRAY_VIEWS_INTEG	
O.ARRAY_VIEWS_INTEG_MOD	
O.NATIVE	
O.NATIVE_MOD	
O.OPERATE	
O.REALLOCATION	
O.RESOURCES	
O.ALARM	
O.CIPHER	
O.RNG	
O.KEY-MNGT	
O.PIN-MNGT	
O.TRANSACTION	
O.OBJ-DELETION	
O.DELETION	
O.LOAD	OSP.VERIFICATION
O.INSTALL	
OE.CAP_FILE	
OE.CARD-MANAGEMENT	
OE.SCP.IC	
OE.SCP.RECOVERY	
OE.SCP.SUPPORT	
OE.VERIFICATION	OSP.VERIFICATION
OE.CODE-EVIDENCE	OSP.VERIFICATION

Table 3 Security Objectives and OSPs – Coverage

Please refer to the base PP for tables regarding **Assumptions and Security Objectives for the Operational Environment – Coverage** and **Security Objectives for the Operational Environment and Assumptions – Coverage** which are not affected by this PP-Module.

10 SECURITY REQUIREMENTS

10.1 SECURITY FUNCTIONAL REQUIREMENTS

This section states the security functional requirements for the PP-Module. The thirteen Security Functional Requirements within this PP-Module are identical in content to those found in the base PP.

10.1.1.1 FIREWALL POLICY

FDP_ACC.2/FIREWALL_MOD Complete access control

FDP_ACC.2.1/FIREWALL_MOD The TSF shall enforce the **FIREWALL access control SFP** on **S.CAP_FILE, S.JCRE, S.JCVM, O.JAVAOBJECT** and all operations among subjects and objects covered by the SFP.

Refinement:

The operations involved in the policy are:

- OP.CREATE,
- OP.INVK_INTERFACE,
- OP.INVK_VIRTUAL,
- OP.JAVA,
- OP.THROW,
- OP.TYPE_ACCESS.
- OP.ARRAY_LENGTH
- OP.ARRAY_T_ALOAD
- OP.ARRAY_T_ASTORE
- OP.ARRAY_AASTORE

FDP_ACC.2.2/FIREWALL_MOD The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

Application Note:

It should be noticed that accessing array's components of a static array, and more generally fields and methods of static objects, is an access to the corresponding O.JAVAOBJECT.

FDP_ACF.1/FIREWALL_MOD Security attribute based access control

FDP_ACF.1.1/FIREWALL_MOD The TSF shall enforce the **FIREWALL access control SFP** to objects based on the following:

Subject/Object	Security attributes
S.CAP_FILE	LC Selection Status
S.JCVM	Active Applets, Currently Active Context
S.JCRE	Selected Applet Context
O.JAVAOBJECT	Sharing, Context, LifeTime

FDP_ACF.1.2/FIREWALL_MOD The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

- **R.JAVA.1 ([JCRE3], §6.2.8): S.CAP_FILE may freely perform, OP.INVK_VIRTUAL, OP.INVK_INTERFACE, OP.THROW or OP.TYPE_ACCESS upon any O.JAVAOBJECT whose Sharing attribute has value "JCRE entry point" or "global array".**
- **R.JAVA.2 ([JCRE3], §6.2.8): S.CAP_FILE may freely perform OP.ARRAY_ACCESS, OP.INSTANCE_FIELD, OP.INVK_VIRTUAL, OP.INVK_INTERFACE or OP.THROW upon any O.JAVAOBJECT whose Sharing attribute has value "Standard" and whose Lifetime attribute has value "PERSISTENT" only if O.JAVAOBJECT's Context attribute has the same value as the active context.**
- **R.JAVA.3 ([JCRE3], §6.2.8.10): S. CAP_FILE may perform OP.TYPE_ACCESS upon an O.JAVAOBJECT with Context attribute different from the currently active context, whose Sharing attribute has value "SIO" only if O.JAVAOBJECT is being cast into (checkcast) or is being verified as being an instance of (instanceof) an interface that extends the Shareable interface.**
- **R.JAVA.4 ([JCRE3], §6.2.8.6): S.CAP_FILE may perform OP.INVK_INTERFACE upon an O.JAVAOBJECT with Context attribute different from the currently active context, whose Sharing attribute has the value "SIO", and whose Context attribute has the value "CAP File AID", only if the invoked interface method extends the Shareable interface and one of the following conditions applies:**

a) The value of the attribute Selection Status of the CAP file whose AID is "CAP File AID" is "Multiselectable",

b) The value of the attribute Selection Status of the CAP file whose AID is "CAP File AID" is "Non-multiselectable", and either "CAP File AID" is the value of the currently selected applet or otherwise "CAP File AID" does not occur in the attribute Active Applets.

- **R.JAVA.5: S.CAP_FILE may perform OP.CREATE upon O.JAVAOBJECT only if the value of the Sharing parameter is "Standard" or "SIO".**

- **R.JAVA.6 ([JCRE3], §6.2.8): S.CAP_FILE may freely perform OP.ARRAY_ACCESS or OP.ARRAY_LENGTH upon any O.JAVAOBJECT whose Sharing attribute has value "global array".**

FDP_ACF.1.3/FIREWALL_MOD The TSF shall explicitly authorise access of subjects to objects based on the following additional rules:

- **1) The subject S.JCRE can freely perform OP.JAVA(") and OP.CREATE, with the exception given in FDP_ACF.1.4/FIREWALL_MOD, provided it is the Currently Active Context.**
- **2) The only means that the subject S.JCVM shall provide for an application to execute native code is the invocation of a Java Card API method (through OP.INVK_INTERFACE or OP.INVK_VIRTUAL)**.

FDP_ACF.1.4/FIREWALL_MOD The TSF shall explicitly deny access of subjects to objects based on the following additional rules:

- **1) Any subject with OP.JAVA upon an O.JAVAOBJECT whose LifeTime attribute has value "CLEAR_ON_DESELECT" if O.JAVAOBJECT's Context attribute is not the same as the Selected Applet Context.**
- **2) Any subject attempting to create an object by the means of OP.CREATE and a "CLEAR_ON_DESELECT" LifeTime parameter if the active context is not the same as the Selected Applet Context**.
- **3) S.CAP_FILE performing OP.ARRAY_AASTORE of the reference of an O.JAVAOBJECT whose sharing attribute has value "global array" or "Temporary".**
- **4) S.CAP_FILE performing OP.PUTFIELD or OP.PUTSTATIC of the reference of an O.JAVAOBJECT whose sharing attribute has value "global array" or "Temporary"**
- **5) R.JAVA.7 ([JCRE3], §6.2.8.2): S.CAP_FILE performing OP.ARRAY_T_ASTORE into an array view without ATTR_WRITABLE_VIEW access attribute.**
- **6) R.JAVA.8 ([JCRE3], §6.2.8.2):S.CAP_FILE performing OP.ARRAY_T_ALOAD into an array view without ATTR_READABLE_VIEW access attribute.**

Application Note: FDP_ACF.1.4/FIREWALL_MOD:

- The deletion of applets may render some O.JAVAOBJECT inaccessible, and the Java Card RE may be in charge of this aspect. This can be done, for instance, by ensuring that references to objects belonging to a deleted application are considered as a null reference. Such a mechanism is implementation-dependent.

In the case of an array type, fields are components of the array ([JVM], §2.14, §2.7.7), as well as the length; the only methods of an array object are those inherited from the Object class.

The Sharing attribute defines five categories of objects:

- Standard ones, whose both fields and methods are under the firewall policy,
- Shareable interface Objects (SIO), which provide a secure mechanism for inter-applet communication,
- JCRE entry points (Temporary or Permanent), who have freely accessible methods but protected fields,
- Global arrays, having both unprotected fields (including components; refer to JavaCardClass discussion above) and methods.
- Array Views, having fields/elements access controlled by access control attributes, ATTR_READABLE_VIEW and ATTR_WRITABLE_VIEW and methods.

When a new object is created, it is associated with the Currently Active Context. But the object is owned by the applet instance within the Currently Active Context when the object is instantiated ([JCRE3], §6.1.3). An object is owned by an applet instance, by the JCRE or by the library where it has been defined (these latter objects can only be arrays that initialize static fields of CAP files).

([JCRE3], Glossary) Selected Applet Context. The Java Card RE keeps track of the currently selected Java Card applet. Upon receiving a SELECT command with this applet's AID, the Java Card RE makes this applet the Selected Applet Context. The Java Card RE sends all APDU commands to the Selected Applet Context.

While the expression "Selected Applet Context" refers to a specific installed applet, the relevant aspect to the policy is the context (CAP file AID) of the selected applet. In this policy, the "Selected Applet Context" is the AID of the selected CAP file.

([JCRE3], §6.1.2.1) At any point in time, there is only one active context within the Java Card VM (this is called the Currently Active Context).

It should be noticed that the invocation of static methods (or access to a static field) is not considered by this policy, as there are no firewall rules. They have no effect on the active context as well and the "acting CAP File" is not the one to which the static method belongs to in this case.

It should be noticed that the Java Card platform, version 2.2.x and version 3.x.x Classic Edition, introduces the possibility for an applet instance to be selected on multiple logical channels at the same time, or accepting other applets belonging to the same CAP file being selected simultaneously. These applets are referred to as multiselectable applets. Applets that belong to a same CAP file are either all multiselectable or not ([JCVM3], §2.2.5). Therefore, the selection mode can be regarded as an attribute of CAP files. No selection mode is defined for a library CAP file.

An applet instance will be considered an active applet instance if it is currently selected in at least one logical channel. An applet instance is the currently selected applet instance only if it is processing the current command. There can only be one currently selected applet instance at a given time ([JCRE3], §4).

FDP_IFC.1/JCVM_MOD Subset information flow control

FDP_IFC.1.1/JCVM_MOD The TSF shall enforce the **JCVM information flow control SFP** on **S.JCVM, S.LOCAL, S.MEMBER, I.DATA and OP.PUT(S1, S2, I)**.

Application Note:

It should be noticed that references of temporary Java Card RE entry points, which cannot be stored in class variables, instance variables or array components, are transferred from the internal memory of the Java Card RE (TSF data) to some stack through specific APIs (Java Card RE owned exceptions) or Java Card RE invoked methods (such as the process (APDU apdu)); these are causes of OP.PUT(S1,S2,I) operations as well.

FDP_IFF.1/JCVM_MOD Simple security attributes

FDP_IFF.1.1/JCVM_MOD The TSF shall enforce the **JCVM information flow control SFP** based on the following types of subject and information security attributes:

Subjects	Security attributes
S.JCVM	Currently Active Context

FDP_IFF.1.2/JCVM_MOD The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

- **An operation OP.PUT(S1, S.MEMBER, I.DATA) is allowed if and only if the Currently Active Context is "Java Card RE";**
- **other OP.PUT operations are allowed regardless of the Currently Active Context's value**.

FDP_IFF.1.3/JCVM_MOD The TSF shall enforce the **[assignment: additional information flow control SFP rules]**.

FDP_IFF.1.4/JCVM_MOD The TSF shall explicitly authorise an information flow based on the following rules: **[assignment: rules, based on security attributes, that explicitly authorise information flows]**.

FDP_IFF.1.5/JCVM_MOD The TSF shall explicitly deny an information flow based on the following rules: **[assignment: rules, based on security attributes, that explicitly deny information flows]**.

Application Note:

The storage of temporary Java Card RE-owned objects references is runtime-enforced ([JCRE3], §6.2.8.1-3).

It should be noticed that this policy essentially applies to the execution of bytecode. Native methods, the Java Card RE itself and possibly some API methods can be granted specific rights

or limitations through the FDP_IFF.1.3/JCVM_MOD to FDP_IFF.1.5/JCVM_MOD elements. The way the Java Card virtual machine manages the transfer of values on the stack and local variables (returned values, uncaught exceptions) from and to internal registers is implementation-dependent. For instance, a returned reference, depending on the implementation of the stack frame, may transit through an internal register prior to being pushed on the stack of the invoker. The returned bytecode would cause more than one OP.PUT operation under this scheme.

FMT_MSA.1/JCRE_MOD Management of security attributes

FMT_MSA.1.1/JCRE_MOD The TSF shall enforce the **FIREWALL access control SFP** to restrict the ability to **modify** the security attributes **Selected Applet Context** to **the Java Card RE**.

Application Note:

The modification of the Selected Applet Context should be performed in accordance with the rules given in [JCRE3], §4 and [JCVM3], §3.4.

FMT_MSA.1/JCVM_MOD Management of security attributes

FMT_MSA.1.1/JCVM_MOD The TSF shall enforce the **FIREWALL access control SFP and the JCVM information flow control SFP** to restrict the ability to **modify** the security attributes **Currently Active Context and Active Applets** to **the Java Card VM (S.JCVM)**.

Application Note:

The modification of the Currently Active Context should be performed in accordance with the rules given in [JCRE3], §4 and [JCVM3], §3.4.

FMT_MSA.2/FIREWALL_JCVM_MOD Secure security attributes

FMT_MSA.2.1/FIREWALL_JCVM_MOD The TSF shall ensure that only secure values are accepted for **all the security attributes of subjects and objects defined in the FIREWALL access control SFP and the JCVM information flow control SFP**.

Application Note:

The following rules are given as examples only. For instance, the last two rules are motivated by the fact that the Java Card API defines only transient arrays factory methods. Future versions may allow the creation of transient objects belonging to arbitrary classes; such evolution will naturally change the range of "secure values" for this component.

- The Context attribute of an O.JAVAOBJECT must correspond to that of an installed applet or be "Java Card RE".
- An O.JAVAOBJECT whose Sharing attribute is a Java Card RE entry point or a global array necessarily has "Java Card RE" as the value for its Context security attribute.
-
- Any O.JAVAOBJECT whose Sharing attribute value is not "Standard" has a PERSISTENT-LifeTime attribute's value.
- Any O.JAVAOBJECT whose LifeTime attribute value is not PERSISTENT has an array type as JavaCardClass attribute's value.

FMT_MSA.3/FIREWALL_MOD Static attribute initialisation

FMT_MSA.3.1/FIREWALL_MOD The TSF shall enforce the **FIREWALL access control SFP** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/FIREWALL_MOD [Editorially Refined] The TSF shall not allow **any role** to specify alternative initial values to override the default values when an object or information is created.

Application Note:

FMT_MSA.3.1/FIREWALL_MOD

- Objects' security attributes of the access control policy are created and initialized at the creation of the object or the subject. Afterwards, these attributes are no longer mutable (FMT_MSA.1/JCRE_MOD). At the creation of an object (OP.CREATE), the newly created object, assuming that the FIREWALL access control SFP permits the operation, gets its Lifetime and Sharing attributes from the parameters of the operation; on the contrary, its Context attribute has a default value, which is its creator's Context attribute and AID respectively ([JCRE3], §6.1.3). There is one default value for the Selected Applet Context that is the default applet identifier's Context, and one default value for the Currently Active Context that is "Java Card RE".
- The knowledge of which reference corresponds to a temporary entry point object or a global array and which does not is solely available to the Java Card RE (and the Java Card virtual machine).

FMT_MSA.3.2/FIREWALL_MOD

- The intent is that none of the identified roles has privileges with regard to the default values of the security attributes. It should be noticed that creation of objects is an operation controlled by the FIREWALL access control SFP. The operation shall fail anyway if the created object would have had security attributes whose value violates FMT_MSA.2.1/FIREWALL_JCVM_MOD.

FMT_MSA.3/JCVM_MOD Static attribute initialisation

FMT_MSA.3.1/JCVM_MOD The TSF shall enforce the **JCVM information flow control SFP** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/JCVM_MOD [Editorially Refined] The TSF shall not allow **any role** to specify alternative initial values to override the default values when an object or information is created.

FMT_SMF.1/MOD Specification of Management Functions

FMT_SMF.1.1/MOD The TSF shall be capable of performing the following management functions:

- **modify the Currently Active Context, the Selected Applet Context and the Active Applets**.

FMT_SMR.1/MOD Security roles

FMT_SMR.1.1/MOD The TSF shall maintain the roles**:**

- **Java Card RE (JCRE),**
- **Java Card VM (JCVM)**.

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

10.1.1.2 AID MANAGEMENT

FIA_UID.2/AID_MOD User identification before any action

FIA_UID.2.1/AID_MOD The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

Application Note:

- By users here it must be understood the ones associated to the CAP files (or applets) that act as subjects of policies. In the Java Card System, every action is always performed by an identified user interpreted here as the currently selected applet or the CAP file that is the subject's owner. Means of identification are provided during the loading procedure of the CAP file and the registration of applet instances.

- The role Java Card RE defined in FMT_SMR.1/MOD is attached to an IT security function rather than to a "user" of the CC terminology. The Java Card RE does not "identify" itself to the TOE, but it is part of it.

FMT_MTD.1/JCRE_MOD Management of TSF data

FMT_MTD.1.1/JCRE_MOD The TSF shall restrict the ability to **modify** the **list of registered applets' AIDs** to **the JCRE**.

Application Note:

- The installer and the Java Card RE manage other TSF data such as the applet life cycle or CAP files, but this management is implementation specific. Objects in the Java programming language may also try to query AIDs of installed applets through the lookupAID(...) API method.
- The installer, applet deletion manager or even the card manager may be granted the right to modify the list of registered applets' AIDs in specific implementations (possibly needed for installation and deletion; see #.DELETION and #.INSTALL).

10.2 SECURITY ASSURANCE REQUIREMENTS

The Evaluation Assurance Level is EAL4 augmented with ALC_DVS.2, ALC_FLR.2, AVA_VAN.5 and ADV_SPM.1.

10.3 SECURITY REQUIREMENTS RATIONALE

10.3.1 OBJECTIVES

10.3.1.1.1 IDENTIFICATION

O.SID is referenced from the base PP. See section 7.3.1.1.1 of the base PP.

O.SID This objective is met by the base PP SFRs detailed in section 7.3.1.1.1 of the base PP, and is additionally contributed to by the following SFRs in this PP-Module: FMT_MSA.1/JCRE_MOD, FMT_MSA.1/JCVM_MOD, FMT_MSA.3/FIREWALL_MOD, FMT_MSA.3/JCVM_MOD, FIA_UID.2/AID_MOD and FMT_MTD.1.1/JCRE_MOD.

10.3.1.1.2 EXECUTION

O.FIREWALL_MOD This objective is met by the FIREWALL access control policy FDP_ACC.2/FIREWALL_MOD and FDP_ACF.1/FIREWALL_MOD, and the JCVM information flow control policy (FDP_IFF.1/JCVM_MOD, FDP_IFC.1/JCVM_MOD). The functional requirements of the class FMT (FMT_MSA.1/JCRE_MOD, FMT_MSA.1/JCVM_MOD, FMT_MSA.2/FIREWALL_JCVM_MOD, FMT_MSA.3/FIREWALL_MOD, FMT_MSA.3/JCVM_MOD, FMT_MTD.1/JCRE_MOD, , FMT_SMR.1/MOD, FMT_SMF.1/MOD,) also indirectly contribute to

meet this objective. O.FIREWALL_MOD has reduced functional requirements compared to O.FIREWALL in the base PP and the objective is met by a smaller set of SFRs that contribute to meeting the objective. Not all SFRs that only indirectly contribute to meeting O.FIREWALL of the base PP are duplicated.

O.GLOBAL_ARRAYS_INTEG_MOD This objective is met by the JCVM information flow control policy (FDP_IFF.1/JCVM_MOD , FDP_IFC.1/JCVM_MOD), which prevents an application from keeping a pointer to the APDU buffer of the card, to the global byte array of the applet's install method or to the global arrays created by the JCSystem.makeGlobalArray(…) method. Such a pointer could be used to access and modify it when the buffer is being used by another application.

O.GLOBAL_ARRAYS_INTEG_MOD is a duplication of O.GLOBAL_ARRAYS_INTEG in the base PP without changes.

O.ARRAY_VIEWS_CONFID_MOD Array views have security attributes of temporary objects where the JCVM information flow control policy (FDP_IFF.1/JCVM_MOD , FDP_IFC.1/JCVM_MOD) prevents an application from storing a reference to the array view. Furthermore, array views may not have ATTR_READABLE_VIEW security attribute which ensures that no application can read the contents of the array view.

O. ARRAY_VIEWS_CONFID _MOD is a duplication of O. ARRAY_VIEWS_CONFID in the base PP without changes.

O.ARRAY_VIEWS_INTEG_MOD Array views have security attributes of temporary objects where the JCVM information flow control policy (FDP_IFF.1/JCVM_MOD , FDP_IFC.1/JCVM_MOD) prevents an application from storing a reference to the array view. Furthermore, array views may not have ATTR_WRITABLE_VIEW security attribute which ensures that no application can alter the contents of the array view.

O. ARRAY_VIEWS_INTEG _MOD is a duplication of O. ARRAY_VIEWS_INTEG in the base PP without changes.

O.NATIVE_MOD This security objective is covered by FDP_ACF.1/FIREWALL_MOD : the only means to execute native code is the invocation of a Java Card API method. This objective mainly relies on the environmental objective **OE.CAP_FILE**, which uphold the assumption A.CAP_FILE.

O. NATIVE_MOD is a duplication of O. NATIVE in the base PP without changes.

10.3.2 Rationale tables of Security Objectives and SFRs

Security Objectives	Security Functional Requirements	Rationale
O.FIREWALL_MOD	FDP_IFC.1/JCVM_MOD, FDP_IFF.1/JCVM_MOD, FDP_ACC.2/FIREWALL_MOD, FDP_ACF.1/FIREWALL_MOD, FMT_MTD.1/JCRE_MOD, FMT_SMR.1/MOD, FMT_SMF.1/MOD, FMT_MSA.2/FIREWALL_JCVM_MOD, FMT_MSA.3/FIREWALL_MOD, FMT_MSA.3/JCVM_MOD, FMT_MSA.1/JCRE_MOD, FMT_MSA.1/JCVM_MOD	Section 10.3.1.1.2
O.GLOBAL_ARRAYS_INTEG_MOD	FDP_IFC.1/JCVM_MOD, FDP_IFF.1/JCVM_MOD	Section 10.3.1.1.2
O.ARRAY_VIEWS_CONFID_MOD	FDP_IFC.1/JCVM_MOD, FDP_IFF.1/JCVM_MOD, FDP_ACC.2/Firewall_MOD, FDP_ACF.1/Firewall_MOD	Section 10.3.1.1.2
O.ARRAY_VIEWS_INTEG_MOD	FDP_IFC.1/JCVM_MOD, FDP_IFF.1/JCVM_MOD, FDP_ACC.2/Firewall_MOD, FDP_ACF.1/Firewall_MOD	Section 10.3.1.1.2
O.NATIVE_MOD	FDP_ACF.1/FIREWALL_MOD	Section 10.3.1.1.2
O.SID	FMT_MSA.1/JCRE_MOD, FMT_MSA.1/JCVM_MOD, FMT_MSA.3/FIREWALL_MOD, FMT_MSA.3/JCVM_MOD, FMT_MTD.1/JCRE_MOD FIA_UID.2/AID_MOD	Section 10.3.1.1.1

Table 4 Security Objectives and SFRs - Coverage

Security Functional Requirements	Security Objectives
FDP_ACC.2/FIREWALL_MOD	O.FIREWALL_MOD, O.ARRAY_VIEWS_CONFID_MOD, O.ARRAY_VIEWS_INTEG_MOD
FDP_ACF.1/FIREWALL_MOD	O.FIREWALL_MOD, O.ARRAY_VIEWS_CONFID_MOD, O.ARRAY_VIEWS_INTEG_MOD
FDP_IFC.1/JCVM_MOD	O.FIREWALL_MOD, O.GLOBAL_ARRAYS_INTEG_MOD, O.ARRAY_VIEWS_CONFID_MOD, O.ARRAY_VIEWS_INTEG_MOD
FDP_IFF.1/JCVM_MOD	O.FIREWALL_MOD, O.GLOBAL_ARRAYS_INTEG_MOD, O.ARRAY_VIEWS_CONFID_MOD, O.ARRAY_VIEWS_INTEG_MOD
FMT_MSA.1/JCRE_MOD	O.FIREWALL_MOD, O.SID
FMT_MSA.1/JCVM_MOD	O.FIREWALL_MOD, O.SID
FMT_MSA.2/FIREWALL_JCVM_MOD	O.FIREWALL_MOD
FMT_MSA.3/FIREWALL_MOD	O.FIREWALL_MOD, O.SID
FMT_MSA.3/JCVM_MOD	O.FIREWALL_MOD, O.SID
FMT_SMF.1/MOD	O.FIREWALL_MOD
FMT_SMR.1/MOD	O.FIREWALL_MOD
FIA_UID.2/AID_MOD	O.SID
FMT_MTD.1/JCRE_MOD	O.FIREWALL_MOD, O.SID

Table 5 SFRs and Security Objectives

10.3.3 DEPENDENCIES

10.3.3.1 SFRS DEPENDENCIES

Requirements	CC Dependencies	Satisfied Dependencies in the PP-Module
FDP_ACC.2/FIREWALL_MOD	(FDP_ACF.1)	FDP_ACF.1/FIREWALL_MOD
FDP_ACF.1/FIREWALL_MOD	(FDP_ACC.1) and (FMT_MSA.3)	FDP_ACC.2/FIREWALL_MOD, FMT_MSA.3/FIREWALL_MOD
FDP_IFC.1/JCVM_MOD	(FDP_IFF.1)	FDP_IFF.1/JCVM_MOD
FDP_IFF.1/JCVM_MOD	(FDP_IFC.1) and (FMT_MSA.3)	FDP_IFC.1/JCVM_MOD, FMT_MSA.3/JCVM_MOD
FMT_MSA.1/JCRE_MOD	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_ACC.2/FIREWALL_MOD, FMT_SMF.1/MOD, FMT_SMR.1/MOD
FMT_MSA.1/JCVM_MOD	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_ACC.2/FIREWALL_MOD, FDP_IFC.1/JCVM_MOD, FMT_SMF.1/MOD, FMT_SMR.1/MOD
FMT_MSA.2/FIREWALL_JCVM_MOD	(FDP_ACC.1 or FDP_IFC.1) and (FMT_MSA.1) and (FMT_SMR.1)	FDP_ACC.2/FIREWALL_MOD, FDP_IFC.1/JCVM_MOD, FMT_MSA.1/JCRE_MOD, FMT_MSA.1/JCVM_MOD, FMT_SMR.1/MOD
FMT_MSA.3/FIREWALL_MOD	(FMT_MSA.1) and (FMT_SMR.1)	FMT_MSA.1/JCRE_MOD, FMT_MSA.1/JCVM_MOD, FMT_SMR.1/MOD
FMT_MSA.3/JCVM_MOD	(FMT_MSA.1) and (FMT_SMR.1)	FMT_MSA.1/JCVM_MOD, FMT_SMR.1/MOD
FMT_SMF.1/MOD	No Dependencies	
FMT_SMR.1/MOD	(FIA_UID.1)	FIA_UID.2/AID_MOD
FIA_UID.2/AID_MOD	No Dependencies	
FMT_MTD.1/JCRE_MOD	(FMT_SMF.1) and (FMT_SMR.1)	FMT_SMF.1/MOD, FMT_SMR.1/MOD

Table 6 SFRs Dependencies

10.3.3.2 SARs Dependencies

Requirements	CC Dependencies	Satisfied Dependencies
ADV_ARC.1	(ADV_FSP.1) and (ADV_TDS.1)	ADV_FSP.4, ADV_TDS.3
ADV_FSP.4	(ADV_TDS.1)	ADV_TDS.3
ADV_IMP.1	(ADV_TDS.3) and (ALC_TAT.1)	ADV_TDS.3, ALC_TAT.1
ADV_SPM.1	(ADV_FSP.4)	ADV_FSP.4
ADV_TDS.3	(ADV_FSP.4)	ADV_FSP.4
AGD_OPE.1	(ADV_FSP.1)	ADV_FSP.4
AGD_PRE.1	No Dependencies	
ALC_CMC.4	(ALC_CMS.1) and (ALC_DVS.1) and (ALC_LCD.1)	ALC_CMS.4, ALC_DVS.2, ALC_LCD.1
ALC_CMS.4	No Dependencies	
ALC_DEL.1	No Dependencies	
ALC_DVS.2	No Dependencies	
ALC_FLR.2	No Dependencies	
ALC_LCD.1	No Dependencies	
ALC_TAT.1	(ADV_IMP.1)	ADV_IMP.1
ASE_CCL.1	(ASE_ECD.1) and (ASE_INT.1) and (ASE_REQ.1)	ASE_ECD.1, ASE_INT.1, ASE_REQ.2
ASE_ECD.1	No Dependencies	
ASE_INT.1	No Dependencies	
ASE_OBJ.2	(ASE_SPD.1)	ASE_SPD.1
ASE_REQ.2	(ASE_ECD.1) and (ASE_OBJ.2)	ASE_ECD.1, ASE_OBJ.2
ASE_SPD.1	No Dependencies	
ASE_TSS.1	(ADV_FSP.1) and (ASE_INT.1) and (ASE_REQ.1)	ADV_FSP.4, ASE_INT.1, ASE_REQ.2
ATE_COV.2	(ADV_FSP.2) and (ATE_FUN.1)	ADV_FSP.4, ATE_FUN.1
ATE_DPT.1	(ADV_ARC.1) and (ADV_TDS.2) and (ATE_FUN.1)	ADV_ARC.1, ADV_TDS.3, ATE_FUN.1
ATE_FUN.1	(ATE_COV.1)	ATE_COV.2

ATE_IND.2	(ADV_FSP.2) and (AGD_OPE.1) and (AGD_PRE.1) and (ATE_COV.1) and (ATE_FUN.1)	ADV_FSP.4, AGD_OPE.1, AGD_PRE.1, ATE_COV.2, ATE_FUN.1
AVA_VAN.5	(ADV_ARC.1) and (ADV_FSP.4) and (ADV_IMP.1) and (ADV_TDS.3) and (AGD_OPE.1) and (AGD_PRE.1) and (ATE_DPT.1)	ADV_ARC.1, ADV_FSP.4, ADV_IMP.1, ADV_TDS.3, AGD_OPE.1, AGD_PRE.1, ATE_DPT.1

Table 7 SARs Dependencies

10.3.4 Rationale for the Security Assurance Requirements

EAL4 is required for this type of TOE and product since it is intended to defend against sophisticated attacks. This evaluation assurance level allows a developer to gain maximum assurance from positive security engineering based on good practices. EAL4 represents the highest practical level of assurance expected for a commercial grade product. In order to provide a meaningful level of assurance that the TOE and its embedding product provide an adequate level of defense against such attacks: the evaluators should have access to the low level design and source code. The lowest level for which such access is required is EAL4.

10.3.5 ADV_SPM.1 Formal TOE security policy model

For certain users, this PP-Module can be selected to provide additional assurance through the development of a formal representation of the TSF and its properties, as defined by the SFRs and the security objectives of this PP-Module, further referred to as the formal model and the formal properties, respectively.

It is expected to establish by means of a formal proof that these formal properties hold in the formal model and to establish by means of a correspondence rationale that the TOE functional specification preserves the formal properties proven for the formal model. ADV_SPM.1 has dependencies on ADV_FSP.4, which is satisfied by EAL4.

10.3.6 ALC_DVS.2 Sufficiency of security measures

Development security is concerned with physical, procedural, personnel and other technical measures that may be used in the development environment to protect the TOE and the embedding product. The standard ALC_DVS.1 requirement mandated by EAL4 is not enough. Due to the nature of the TOE and embedding product, it is necessary to justify the sufficiency of these procedures to protect their confidentiality and integrity. ALC_DVS.2 has no dependencies.

10.3.7 AVA_VAN.5 ADVANCED METHODICAL VULNERABILITY ANALYSIS

The TOE is intended to operate in hostile environments. AVA_VAN.5 "Advanced methodical vulnerability analysis" is considered as the expected level for Java Card technology-based products hosting sensitive applications, in particular for payment and identity areas. AVA_VAN.5 has dependencies on ADV_ARC.1, ADV_FSP.4, ADV_TDS.3, ADV_IMP.1, AGD_OPE.1,AGD_PRE.1 and ATE_DPT.1. All of them are satisfied by EAL4.

10.3.8 ALC_FLR.2 FLAW REPORTING PROCEDURES

Due to the nature of the TOE and embedding product, it is necessary to provide flaw reporting procedures to track all reported security flaws in each release of the TOE.

In order for the developer (of Java Card technology-based product) to be able to act appropriately upon security flaw reports from TOE users, and to know to whom to send corrective fixes, TOE users need to understand how to submit security flaw reports to the developer. Flaw remediation guidance from the developer to the TOE user is necessary to ensure that TOE users are aware of this important information.

ALC_FLR.2 has no dependencies.

End of Document