

## Mobile Code Authentication Protection Profile

---

Issue : 0.7

Issue date : 14 May 2001

Status : Draft

Project/document reference : 363.20754.83.1

Authors : Steve H Hill  
Kevin Appleford

Reviewed by : Andy Webber

Distribution : CESG  
Logica

## Table Of Contents

<b>1</b>	<b>Introduction .....</b>	<b>3</b>
1.1	PP Identification.....	3
1.2	PP Overview .....	3
1.3	Related PPs.....	4
1.4	CC Conformance Claim .....	4
1.5	Glossary of Terms.....	5
1.6	Document Structure .....	6
1.7	References .....	6
<b>2</b>	<b>TOE Description .....</b>	<b>8</b>
2.1	Intended Use .....	8
2.2	IT Features.....	10
<b>3</b>	<b>TOE Security Environment.....</b>	<b>12</b>
3.1	Environmental and Method of Use Assumptions .....	12
3.2	Assumed Threats.....	13
3.3	Organisational security policies.....	14
<b>4</b>	<b>Security Objectives .....</b>	<b>16</b>
4.1	TOE Security Objectives.....	16
4.2	Security Objectives for the Environment .....	16
<b>5</b>	<b>IT Security Requirements.....</b>	<b>18</b>
5.1	TOE Security Functional Requirements .....	18
5.2	TOE Security Assurance Requirements.....	22
5.3	SOF claim.....	22
5.4	Security requirements for the IT environment .....	23
<b>6</b>	<b>Protection Profile Rationale.....</b>	<b>25</b>
6.1	Security Objectives Rationale.....	25
6.2	Security Requirements Rationale.....	30

# 1 Introduction

## 1.1 PP Identification

Title:	Mobile Code Authentication Protection Profile
Issue:	0.7
Publication date:	14 May 2001
CC Version:	2.1 (also known as ISO standard 15408)
Sponsoring organisation:	CESG

Other PP identification information required by [CC] can be found on the front page of this document. Related PPs are referenced in section 1.3.

## 1.2 PP Overview

This protection profile has been developed to identify and describe the security requirements needed for providing security when downloading and executing mobile code, in particular, the authentication as to the source of such code. The Target of Evaluation (TOE) for this protection profile is a desktop application (typically a browser), or it may be a firewall or gateway proxy.

For the purposes of this PP **mobile code** is defined as code that is downloaded from a remote source over a network and run on a user's PC with little or no intervention from the user. The most common types of mobile code are:

- scripting languages such as JavaScript and VB Script;
- Java;
- ActiveX controls;
- Macros within office automation applications such as word processors, spreadsheets and databases.

This PP defines requirements to check the authenticity and integrity of downloaded mobile code before it is executed. This is achieved through validation of the digital signatures on the code, and of the public key certificates issued by a Certification Authority (CA) which provide proof that the code was indeed digitally signed by the identified signatory. The TOE communicates with the CA to obtain the digital certificate and certificate revocation list.

The security assurance requirements are specified as either EAL3 or EAL4. In this respect this (physical) PP document can be considered to contain two (logical) PPs that differ only in respect of the security assurance requirements.

Although this PP specifies constraints which make compliant and evaluated TOEs suitable for HMG installations, it does not require government only technologies, and may therefore be used in non HMG environments.

### 1.3 Related PPs

The Mobile Code Quarantine PP [QuarPP] and the Mobile Code Isolation PP [IsolPP] both also define security requirements to address the threat of malicious or unauthorised mobile code, and are closely related to this PP.

For example, a firewall or gateway proxy claiming conformance with the Mobile Code Quarantine PP may also claim conformance with this PP, since the information flow control decisions it enforces with respect to mobile code being downloaded may depend (in part) on the result of a check on the authenticity of the mobile code.

Similarly, a desktop application which is claimed to be conformant with the Mobile Code Isolation PP may also be claimed conformant with this PP, since the operations it permits the mobile code to perform may depend, in part, on the degree of trust in that code as established through a check on its authenticity.

### 1.4 CC Conformance Claim

This PP is Part 2 conformant and Part 3 conformant for EAL3 or EAL4, as appropriate.

An ST claiming conformance with this PP shall clearly indicate at which level conformance is being claimed, e.g. by including a statement such as the following:

*This ST claims conformance with the Mobile Code Authentication Protection Profile at the EAL4 level of assurance.*

## 1.5 Glossary of Terms

This section contains definitions of technical terms that are used with a meaning specific to this document. Terms defined in the [CC] are not reiterated here, unless stated otherwise.

*Denial of service attacks:* These attacks aim to disrupt the normal operation of computer hardware or software to prevent its use. This commonly involves using so much of a particular resource that the response time becomes so slow as to make the resource effectively unusable, or that a critical component crashes.

*Firewall:* A firewall is a set of related programs and servers acting as a network gateway that protects the resources of a private network from users on other networks.

*Gateway:* A gateway is a network node that acts as a gateway to another network.

*Mobile Code:* Code that is downloaded and run with little or no user intervention. This definition excludes code manually downloaded from remote sites and run, e.g. via ftp. This is because, with manually downloaded code, the onus is on the user to ensure that the code is safe before running it, or that it runs in such a way as to minimise any damage it may cause.

*Private key:* A cryptographic key used with a public key cryptographic algorithm, uniquely associated with an entity, and not made public.

*Proxy:* A proxy is an application that acts as an intermediary between a workstation user application and the Internet to provide security.

*Public key:* A cryptographic key used with a public key cryptographic algorithm, uniquely associated with an entity, and which may be made public.

*Public Key Cryptographic algorithm:* A cryptographic algorithm that uses two related keys, a public key and a private key; the two keys have the property that, given the public key, it is computationally infeasible to derive the private key.

*Trusted networks:* Networks whose users are trusted not to:

- maliciously interfere with data travelling over the network,
- launch attacks on PCs on the network or
- passively offer malicious data to other PCs on the network.

Trusted networks are typically connected to one or more, larger networks, e.g. the Internet, in which there is less or no trust.

*Untrusted networks:* Networks that are not trusted.

## 1.6 Document Structure

The structure of this document is as defined by [CC] Part 1 Annex C.

- Section 1 (this section) is the PP Introduction
- Section 2 is the TOE Description.
- Section 3 provides the statement of TOE security environment.
- Section 4 provides the statement of security objectives.
- Section 5 provides the statement of IT security requirements.
- Section 6 provides the PP rationale.

## 1.7 References

CC	Common Criteria for Information Technology Security Evaluation (Comprising Parts 1-3, [CC1], [CC2], [CC3]).
CC1	Common Criteria for Information Technology Security Evaluation Part 1: Introduction and General Model CCIMB-99-031, Version 2.1, August 1999.
CC2	Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements CCIMB-99-032, Version 2.1, August 1999.
CC3	Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Requirements CCIMB-99-033, Version 2.1, August 1999.
IS1	HMG Infosec Standard No. 1 (IS1), April 1998
IsolPP	Mobile Code Isolation Protection Profile, 363.20754.84.1, Issue 0.4, April 2001

Memo21           CESG Infosec Memorandum No 21 Risk Management of Mobile Code,  
Issue 1.0, January 2001

QuarPP           Mobile Code Quarantine Protection Profile,  
363.20754.85.1, Issue 0.4, April 2001

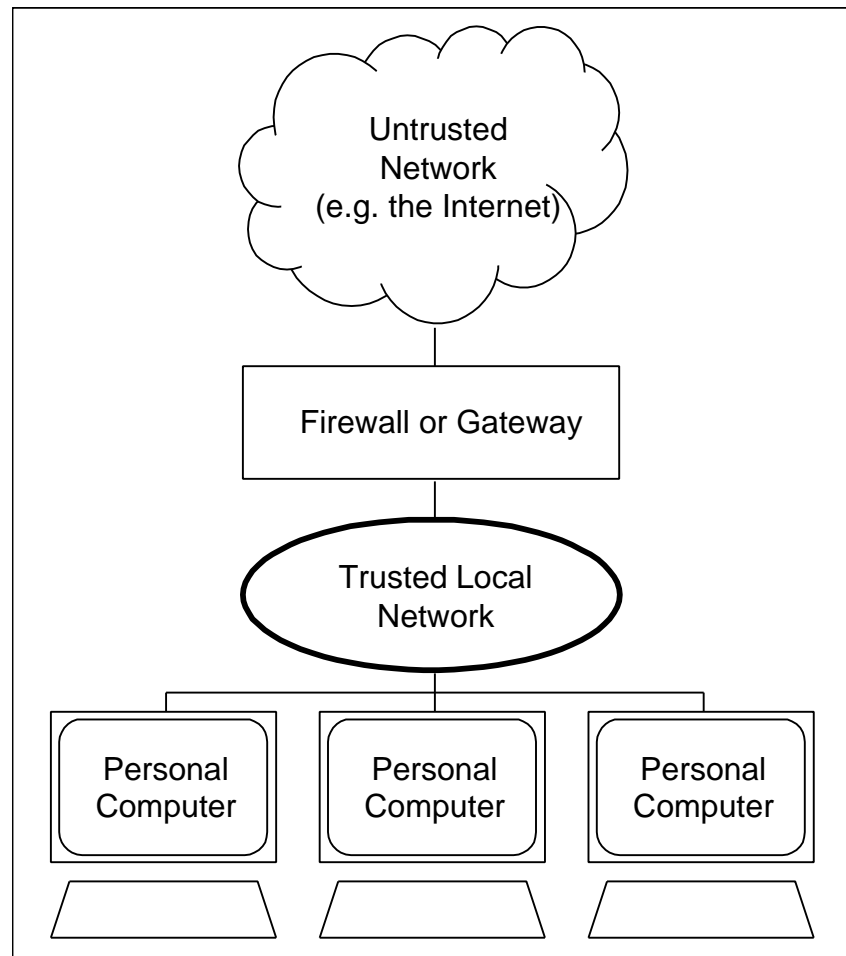
## 2 TOE Description

The TOE is either a proxy application on a firewall or gateway server between two networks, or a desktop application (typically a browser) on a host node of a network.

### 2.1 Intended Use

There are a number of different approaches to securing end users against malicious mobile code, and it is likely that individuals, organisations and applications will implement a combination of these.

For organisations, the boundary between a trusted and untrusted network is typically controlled with a firewall or other filtering device. The aim is to prevent attackers on the untrusted network gaining access to information or resources on the trusted network by gaining control of a trusted PC or snooping on trusted network traffic. Figure 1 illustrates a typical network infrastructure. Firewalls or filtering devices may also be required to create boundaries between divisions within an organisation.



**Figure 1: Typical Network Infrastructure**

A correctly implemented and configured firewall will prevent most active attacks originating from the untrusted network. However, the assumed environment for the TOE described by this PP is one in which there is a business need to allow some mobile code through. This presents an opportunity to the attacker who may subvert the mobile code, or lure the unsuspecting user into executing malicious mobile code, so that it performs an attack when run on a workstation on the trusted network. The assets on the trusted network may also be placed at risk as a result of the execution of mobile code which has design or implementation flaws.

The boundary between trusted and untrusted networks is context dependent. For many organisations, it will be the connection between the organisation's network and the Internet. But for some organisations, boundaries must be protected between organisational divisions, e.g. to control access between research and development department and personnel departments. Whilst in some situations, there is no local network, e.g. a user dialling into the Internet via an ISP where the PC connects directly to the untrusted network.

In this simpler situation where the user at their workstation attaches directly to an untrusted network, perhaps the Internet, protection can be provided by ensuring that the workstation application itself filters mobile code, passing only code from trusted sites for execution to the workstation.

This PP addresses these mobile code aware applications, where code authentication is used to allow applications and their users a finer granularity of control over the execution of mobile code, whether provided by applications at the firewall or at the desktop.

## 2.2 IT Features

As stated above, the TOE type is either a firewall or gateway proxy, or is a desktop application (typically, a browser). Common to both types is the provision of functionality which enables a user on the trusted network to download and/or execute mobile code coming from the untrusted network.

The principal security feature of the TOE (insofar as this PP is concerned) is to control the downloading or execution of mobile code from the untrusted network on the basis of code authentication checks. Code authentication allows a person or organisation to vouch for a piece of code. It relies upon digital signatures, which in turn rely on public-private key pairs and certificates.

Typically an individual will create a public-private key pair and a commercial Certification Authority (CA) will generate a certificate binding the relationship between attributes of the individual and their public key. CAs are responsible for validating this relationship before issuing the certificate. The trust that can be placed in the binding is a combination of the rigour of the checks (which is proportional to their intended use) and the trust that can be placed in the CA.

Armed with a public-private key pair, an associated certificate and relevant signing software it is possible to sign mobile code. This is usually based on a checksum of the data comprising the mobile code so that any alteration of the data after the signature has been created can be detected.

The TOE checks a digital signature by obtaining the signer's certificate and using the public key it contains to check that it matches the mobile code it accompanies. To ensure that the certificate is currently valid, the CA that issued the certificate is contacted to ensure that the certificate hasn't been revoked. This could happen, for example, if the signer's private key has been lost or stolen.

The trust that can be associated with signed code is only as strong as the trust that may be placed in the signatory and the CA, as:

- The meaning of a signature is specific to the context in which it is used, e.g. it may indicate the mobile code author, a person who is vouching for the code's non-malicious behaviour or something else entirely;
- Anybody can sign mobile code given a public-private key pair and the correct tools, perhaps removing an existing signature to do so. Signatures can only be used to associate signed mobile code with the identity of the signer, not the mobile code's author.

### 3 TOE Security Environment

The statement of TOE security environment describes the security aspects of the environment in which the TOE is intended to be used and the manner in which it is expected to be employed.

To this end, the statement of TOE security environment identifies and lists the assumptions made on the environment and the intended method of use of the TOE, defines the threats that the TOE and its environment is to counter, and the organisational security policies with which the TOE and its environment is to comply.

#### 3.1 Environmental and Method of Use Assumptions

##### 3.1.1 Method of use

**A.SOURCEPOL** It is assumed that the TOE is being used within an organisation that has a clearly defined policy for determining which other organisations and individuals can be considered to be trustworthy originators of mobile code.

This assumption is included to clearly define the scope of the ‘security problem’ to be addressed. Each of the threats defined in the following section is predicated on the assumption that, insofar as the organisation using the TOE is concerned, sources of mobile code can be categorised as trusted or untrusted. Clearly, if the organisation is unable to categorise sources of mobile code in this way, then there is little point in being able to determine where mobile code originated from.

##### 3.1.2 Connectivity

**A.CERTAUTH** There exists one or more Certification Authorities trusted by the organisation owning the trusted network which issue public key certificates, and with which the TOE can communicate.

The Certification Authorities with which the TOE can communicate are external to the TOE.

**A.CHOKE** If the TOE is a firewall or gateway proxy, it is assumed that it is deployed at the choke point between an untrusted external network and the trusted local network.

The intent of this assumption is to preclude the threat of bypass of firewall or gateway proxy TOEs. The assumption is clearly not relevant if the TOE is a desktop application.

## **3.2 Assumed Threats**

This section describes the threats to the assets that require protection.

### **3.2.1 Assets**

The primary assets of concern to this PP are the information and resources on the trusted network, the integrity and availability of which is important to the organisation which owns them. Specific data assets on the trusted network may also require protection of their confidentiality against more directed attacks, e.g. user passwords, secret or private cryptographic keys, personal information, and so on.

### **3.2.2 Threat agents**

The threat agents are attackers on the untrusted network who write mobile code. They are assumed to have various levels of expertise, motivation and resources. Their motivation may arise from a variety of reasons, including a desire to:

- impede or interfere with the operation of specific organisations;
- gain financially through attack directed against individuals rather than organisations, e.g. to obtain credit card details;
- seek publicity or gain notoriety.

### **3.2.3 Threat descriptions**

The general threat to be countered is that of mobile code imported from the untrusted network compromising integrity and availability of information and resources, as well as possibly the confidentiality of information (e.g. user passwords, cryptographic keys, or other personal information). For the purposes of the statement of threats in the PP, 'mobile code' excludes any code which requires explicit user intervention before it executes.

Note that the threat of mobile code which originates from a trustworthy source, but unintentionally behaves in such a way as to compromise the security of the assets, is not within the scope of the 'security problem' to be addressed.

Specific threats to be countered are as follows:

**T.UNTRUSTED** A compromise of assets occurs as a result of the execution of mobile code that originates from an untrusted source.

This threat concerns the case where a user on the trusted network executes mobile code that is not known to come from a source which the organisation considers to be trusted.

**T.ALTERED** A compromise of assets occurs as a result of the execution of mobile code that originates from an trustworthy source, but has subsequently been altered by an attacker to act in a malicious manner.

This threat concerns the case where a user on the trusted network executes mobile code that has in fact originated from a source that the organisation believes to be trusted. However this code has subsequently been modified by an attacker.

**T.SPOOF** A compromise of assets occurs as a result of the execution of mobile code that appears to originate from an trustworthy source, but which in fact has been written by an attacker to act in a malicious manner.

This threat concerns the case where the attacker tries to hoodwink a user into executing mobile code that apparently comes from a source that is regarded as trusted.

**T.NOREPUD** Mobile code originating from a source regarded as trustworthy is subsequently found to cause a compromise of assets, for which the originator denies any responsibility.

This threat concerns the case where damage to assets has occurred despite the code having originated from a source regarded as trustworthy. If the originator of the malicious code successfully denies responsibility then this may result in the originator continuing to be regarded as a trusted source; this in turn may place the assets of the organisation at risk of further compromise in the future.

This scenario might occur as a result of a trusted source being subverted to produce intentionally malicious code; alternatively, there may have been a serious failure of quality controls on the part of the originator, resulting in bugs in the code which damage the assets when it is executed. In either case the organisation will wish to consider whether the originator should continue to be regarded as a trusted source of mobile code.

### 3.3 Organisational security policies

The following organisational security policies (OSPs) apply to the TOE.

**P.DSV** Digital Signature Validation mechanisms shall be approved by the relevant cryptographic authority.

This OSP should be expanded in the ST for a conformant TOE by identifying the appropriate cryptographic authority and also any standards mandated by that authority with which the TOE must comply. For example, within HMG the relevant authority is CESG, and the relevant standard is [Memo21] (which identifies the algorithms that must be used and required characteristics of the associated keys).

**P.EAL** IT countermeasures shall be assured to a level equal to EAL3 as a minimum.

## 4 Security Objectives

### 4.1 TOE Security Objectives

**O.AUTHCHECK** The TOE shall employ an approved Digital Signature Validation mechanism to determine the authenticity and integrity of mobile code that is downloaded from the untrusted network.

The Digital Signature Validation mechanism shall be approved by the relevant cryptographic authority as stated in the P.DSV OSP.

**O.CERTCHECK** The TOE shall check the validity of public key certificates before confirming that the digital signature on downloaded mobile code is valid. This shall include a check for certificate revocation and against its validity period.

Validation of public key certificates is needed to establish trust in the binding between the claimed identity and the public key. This validity check is only required on entry to the trusted network: there is no requirement to check the certificate every time the mobile code is executed.

**O.ENFORCE** The TOE shall ensure that downloaded mobile code cannot be executed before a successful check on the validity of digital signatures and the associated public key certificates.

The intent of this objective is to ensure that the checks implemented by the TOE cannot be bypassed or otherwise circumvented.

**O.EAL** The TOE security functions shall be assured to EAL3 as a minimum.

This security objective is to be achieved by the TOE security assurance requirements, and is driven by the P.EAL OSP.

### 4.2 Security Objectives for the Environment

**O.USER\_ED** Those responsible for the TOE environment shall ensure that users are made aware of the dangers of executing mobile code from an untrusted source.

This security objective is relevant mainly where the TOE is a desktop application, and there is therefore a need to ensure users are made aware of the dangers posed by mobile code, and the importance of following the organisation's security operating procedures. This includes having an awareness of the organisation's policy with respect to trusting sources of mobile code on the untrusted network, and not disabling the checks made by the TOE.

**O.TRUSTSIGN** Those responsible for the TOE environment shall establish and implement procedures for determining the degree of trust that may be placed in individuals or organisations that sign mobile code.

The trust that may be placed in downloaded mobile code is limited by the extent to which the signatory of the mobile code can be trusted.

**O.CERTAUTH** The Certification Authorities shall exercise appropriate controls over the issuing and revocation of public key certificates to ensure that the owner of the public key is reliably identified. This shall include taking appropriate steps to protect the keys used by the CA to digitally sign such certificates, which shall be performed in accordance with P.DSV.

This security objective is necessary in order to be able to establish appropriate confidence in the binding between an individual and his or her public key. This includes establishing appropriate procedural measures to verify the claimed identity of the individual requesting a digital certificate. In the case where the individual claims to be a member of an organisation which is to be identified on the certificate, the CA should take appropriate steps to verify the claim.

In practice, this security objective means that those responsible for the TOE environment should determine the degree to which they trust the Certification Authority which issued the public key certificate for the signatory of the mobile code.

**O.SAFEKEY** Those responsible for the TOE environment shall establish and implement procedures to safeguard the integrity of the public keys of the Certification Authorities.

As a minimum the root key must be safeguarded, so that the binding between the CA and its public/private key pair is not undermined.

**O.KEYDIST** Those responsible for the TOE environment shall establish and implement procedures to ensure the authenticity of the public keys of the Certification Authorities.

Again, this objective is concerned with maintaining the binding between the CA and its public/private key pair.

**O.CONNECT** Those responsible for the TOE environment shall ensure that:

1. The TOE is able to communicate with the necessary Certification Authorities.
2. If the TOE is a firewall or gateway proxy, it is deployed at the choke point between an untrusted external network and the trusted local network.

## 5 IT Security Requirements

### 5.1 TOE Security Functional Requirements

This section identifies the security functional requirements (SFRs) required of the TOE to meet its security objectives.

The components taken from [CC2] to specify the SFRs are listed in the table below together with an indication of whether the components are *iterated* (indicated by “(\*N)” where N identifies the number of iterations) or *refined*. (See [CC2, 2.1.4] for an explanation of permitted operations on functional components.)

Assignment and selection operations to be completed by the ST author are indicated using the same notation as used in [CC2]. Partially completed operations are denoted by the use of *italicised text* for the key word *assignment* or *selection* (as appropriate). Completed assignment and selection operations are indicated by *italicised text*. Refinements of components are indicated by **emboldened text**.

CLASS	FAMILY	COMPONENT	REFINED?
FCO	FCO_NRO	FCO_NRO.1	Y
FCS	FCS_COP	FCS_COP.1 (*2)	
FDP	FDP_DAU	FDP_DAU.2	Y
	FDP_IFC	FDP_IFC.1	
	FDP_IFF	FDP_IFF.1	Y
	FDP_ITC	FDP_ITC.1	Y
FPT	FPT_RVM	FPT_RVM.1	Y
	FPT_SEP	FPT_SEP.1	

**Table 1 – TOE Security Functional Requirements**

### 5.1.1 Communication (FCS)

#### 5.1.1.1 Proof of origin of mobile code (FCO\_NRO.1)

FCO\_NRO.1.1 The TSF shall be able to **extract digital signatures** for **downloaded mobile code** at the request of the *recipient*.

*Application note:* This has been refined for clarity, replacing 'generate evidence of origin' with 'extract digital signatures' and 'transmitted' with 'downloaded'.

FCO\_NRO.1.2 The TSF shall be able to relate *the identity* of the originator of the **mobile code** and the code to which the **digital signature** applies.

*Application note:* This has been refined for clarity, replacing 'information' with 'mobile code' and 'evidence' with 'digital signature', and deleting the superfluous words 'of the information'.

FCO\_NRO.1.3 The TSF shall provide a capability to verify **digital signatures** to *the recipient* given [assignment: *limitations on the evidence of origin*].

### 5.1.2 Cryptographic Services (FCS)

#### 5.1.2.1 Cryptographic operations: digital signature verification (FCS\_COP.1)

FCS\_COP.1.1(1) The TSF shall perform *digital signature verification* in accordance with a specified cryptographic algorithm [assignment: *approved digital signature verification algorithm*] and cryptographic key sizes [assignment: *approved cryptographic key sizes*] that meet the following: [assignment: *list of standards defined by the relevant cryptographic authority*].

*Application note:* The assignments are to be completed in accordance with the requirements derived from the P.DSV OSP.

#### 5.1.2.2 Cryptographic operations: hash generation (FCS\_COP.1)

FCS\_COP.1.1(2) The TSF shall perform *hash generation* in accordance with a specified cryptographic algorithm [assignment: *approved hash generation algorithm*] and cryptographic key sizes [assignment: *approved cryptographic key sizes*] that meet the following: [assignment: *list of standards defined by the relevant cryptographic authority*].

*Application note:* The assignments are to be completed in accordance with the requirements derived from the P.DSV OSP.

### 5.1.3 User Data Protection (FDP)

#### 5.1.3.1 Mobile code authentication with identity of guarantor (FDP\_DAU.2)

FDP\_DAU.2.1 The TSF shall provide a capability to **obtain, from a Certification Authority, public key certificates** that can be used as a guarantee of the validity of *the*

*binding between the identified name and the identified public key, and possession of the associated private key.*

*Application note: This element has been refined by replacing 'generate evidence' with 'obtain, from a Certification Authority, public key certificates'. This SFR means that the TOE must have the capability to communicate with the CA to obtain the correct public key certificate.*

FDP\_DAU.2.2 The TSF shall provide [assignment: *list of subjects*] with the ability to verify **public key certificates** and the identity of the **Certification Authority** that generated the **certificate**.

**Refinement: Certificate verification shall involve, as a minimum:**

- a) signature verification;**
- b) checking the validity period;**
- c) checking for revocation.**

*Application note: The assignment should be completed according to the type of TOE, e.g. if a desktop application the assignment should be users (or similar).*

5.1.3.2 Mobile code information flow control policy (FDP\_IFC.1)

FDP\_IFC.1.1 The TSF shall enforce the *mobile code information flow control policy* on:

- a) *subjects on the trusted network*
- b) *mobile code on the untrusted network*
- c) *operations that cause mobile code to be downloaded from the untrusted network.*

5.1.3.3 Mobile code information flow control rules (FDP\_IFF.1)

FDP\_IFF.1.1 The TSF shall enforce the *mobile code information flow control policy* based on the following types of subject and information security attributes:

- a) *digital signatures on mobile code and associated public key certificates*
- b) *[assignment: other security attributes used as the basis for information flow control decisions].*

FDP\_IFF.1.2 The TSF shall permit **mobile code to be downloaded from the untrusted network to the trusted network** if the following rules hold:

- a) *download is permitted only if the mobile code authentication check succeeds, as follows:*
  - *the mobile code is signed with a valid digital signature*
  - *verification of the public key certificate succeeds.*

- b) [assignment: other rules implemented by the TOE that govern whether or not mobile code may be downloaded].

*Application note:* The code authentication checks in paragraph a) refer, respectively, to the digital signature verification checks specified in FCO\_NRO.1, and the public key certificate verification specified in FDP\_DAU.2.2 and FDP\_ITC.1.3.

*Paragraph b) should be completed with any other checks that are enforced by the TOE (this will be appropriate if the ST claims conformance with the Mobile Code Quarantine PP). An assignment of 'none' is permitted; in this event, the paragraph should be omitted to improve readability.*

- FDP\_IFF.1.3 The TSF shall enforce the [assignment: additional information flow control SFP rules].

*Application note:* This SFR should be completed to describe any rules enforced by the TOE that are additional to the rules stated in FDP\_IFF.1.2, i.e. other determinations that may be made to decide whether or not to download the mobile code. Note that any 'override' of the FDP\_IFF.1.2 rules should be specified using FDP\_IFF.1.5 or FDP\_IFF.1.6 as appropriate. An assignment of 'no additional rules' is acceptable.

- FDP\_IFF.1.4 The TSF shall provide the following [assignment: list of additional SFP capabilities].

*Application note:* This SFR should be completed to describe any additional capabilities provided by the TOE in respect of making mobile code download decisions. An assignment of 'no additional rules' is acceptable.

- FDP\_IFF.1.5 The TSF shall explicitly authorise an information flow based on the following rules: [assignment: rules, based on security attributes, that explicitly authorise information flows].

*Application note:* This SFR should be completed to describe any exceptions to the rules stated in FDP\_IFF.1.2. For example, a browser may permit a user to explicitly authorise the download of unsigned mobile code (this might be permitted where the code is known to originate from a web-site on the untrusted network that belongs to a trusted organisation).

- FDP\_IFF.1.6 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:* This SFR should be completed to describe any exceptions to the rules stated in FDP\_IFF.1.2.

#### 5.1.3.4 Import of public key certificates (FDP\_ITC.1)

- FDP\_ITC.1.1 The TSF shall enforce the mobile code information flow control policy when importing **public key certificates** from outside of the TSC.

- FDP\_ITC.1.2 The TSF shall ignore any security attributes associated with the **public key certificates** when imported from outside the TSC.

*Application note:* FDP\_ITC.1.2 is simply included for conformance with [CC2]. It has no meaning within the context of import of public key certificates.

FDP\_ITC.1.3 The TSF shall enforce the following rules when importing **public key certificates** from outside the TSC: *the public key certificate shall be discarded if the check on the validity of the digital signature of the CA fails.*

## 5.1.4 Protection of TOE security Functions (FPT)

5.1.4.1 Non-bypassability of the TOE Security Policy (FPT\_RVM.1)

FPT\_RVM.1.1 The TSF shall ensure that **the mobile code authentication** functions are invoked and succeed before **import of mobile code** within the TSC is allowed to proceed.

*Application note:* The component has been refined to emphasise that the SFR is concerned with prevention of bypass of the mobile code authentication function.

5.1.4.2 TSF domain separation (FPT\_SEP.1)

FPT\_SEP.1.1 The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT\_SEP.1.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

## 5.2 TOE Security Assurance Requirements

As stated in Section 1.2, the security assurance requirements are specified as either EAL3 or EAL4, as defined in [CC3].

The effect of this approach is that this (physical) PP document contains two (logical) PPs that differ only in respect of the EAL mandated. As stated in Section 1.4, an ST claiming conformance to the Mobile Code Authentication Protection Profile shall clearly state the level at which conformance is claimed.

## 5.3 SOF claim

The minimum SOF rating required by this PP is **SOF-medium**.

The strength of cryptographic algorithms is outside the scope of the CC, and hence the assessment of algorithmic strength will not form part of the TOE evaluation.

## 5.4 Security requirements for the IT environment

The effectiveness of the TOE depends on the controls implemented by the Certification Authorities to maintain the authenticity and integrity of the public key certificates it signs. Note that the following requirements represent the *minimum* requirements necessary; no attempt is made to resolve dependencies between the requirements listed below, as that is an issue for a separate PP/ST dealing with the underlying PKI requirements.

CLASS	FAMILY	COMPONENT	REFINED?
FCS	FCS_CKM	FCS_CKM.1	Y
		FCS_CKM.4	Y
	FCS_COP	FCS_COP.1 (*2)	
FMT	FMT_REV	FMT_REV.1	Y

**Table 2 – IT Environment Security Functional Requirements**

The following requirements relate to the generation and destruction of the public/private key pair used by the CA to sign public key certificates.

FCS\_CKM.1.1 The **CA** shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [*assignment: approved cryptographic key generation algorithm*] and specified cryptographic key sizes [*assignment: approved cryptographic key sizes*] that meet the following: [*assignment: list of standards defined by the relevant cryptographic authority*].

FCS\_CKM.4.1 The **CA** shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [*assignment: approved cryptographic key destruction method*] that meets the following: [*assignment: list of standards defined by the relevant cryptographic authority*].

The following security requirements relate to the generation of public key certificates by the CA (the specific requirements deriving from O.CERTAUTH).

FCS\_COP.1.1(1) The **CA** shall perform *digital signature generation for public key certificates* in accordance with a specified cryptographic algorithm [*assignment: approved digital signature verification algorithm*] and cryptographic key sizes [*assignment: approved cryptographic key sizes*] that meet the following: [*assignment: list of standards defined by the relevant cryptographic authority*].

FCS\_COP.1.1(2) The **CA** shall perform *hash generation for public key certificates* in accordance with a specified cryptographic algorithm [*assignment: approved hash generation*]

*algorithm*] and cryptographic key sizes [*assignment: approved cryptographic key sizes*] that meet the following: [*assignment: list of standards defined by the relevant cryptographic authority*].

The following requirement relates to the revocation of public key certificates.

FMT\_REV.1.1 The **CA** shall restrict the ability to revoke **public key certificates** associated with the **signatories of mobile code** to [*assignment: the authorised identified roles*].

FMT\_REV.1.2 The **CA** shall enforce the rules:

- a) *once revoked the public key certificate is no longer valid;*
- b) *the CA shall inform the TOE on request if a specified public key certificate has been revoked.*

## 6 Protection Profile Rationale

This section demonstrates the suitability of the choice of security objectives and security requirements to address the security needs posed by the statement of TOE security environment.

### 6.1 Security Objectives Rationale

This section demonstrates how the threats are countered, and the organisational security policies and assumptions are met, by the security objectives.

#### 6.1.1 Security objectives are suitable to counter the threats

The table below shows that each threat is countered by at least one security objective, and that each security objective contributes to countering at least one threat.

Threats	Objectives									
	O.AUTHCHECK	O.CERTCHECK	O.ENFORCE	O.EAL	O.USER_ED	O.TRUSTSIGN	O.CERTAUTH	O.SAFEKEY	O.KEYDIST	O.CONNECT
T.UNTRUSTED	x	x	x	x	x	x	x	x	x	x
T.ALTERED	x	x	x	x			x	x	x	x
T.SPOOF	x	x	x	x	x	x	x	x	x	x
T.NOREPUD	x	x		x			x	x	x	x

**Table 3 - Correlation between Threats and Objectives**

The following rationale demonstrates how the objectives counter the threats:

**T.UNTRUSTED** *A compromise of assets occurs as a result of the execution of mobile code that originates from an untrusted source.*

O.AUTHCHECK counters this threat by providing the mechanism by which the validity of the digital signature on the mobile code may be checked. This is supported by O.ENFORCE which ensures that the mobile code cannot be executed before these checks are made. Unsigned mobile code will therefore not be executed.

O.USER\_ED ensures that where the TOE is a desktop application that users are aware of the dangers of executing mobile code from an untrusted source.

O.TRUSTSIGN ensures that the trust that may be placed in the signatories of mobile code has been determined, and thus, in particular, whether this signatory can be trusted, countering the threat that untrusted code might be executed which may lead to compromise of the assets.

O.CERTCHECK supports O.AUTHCHECK by providing confidence in the validity of the public key certificate for the identified signatory of the mobile code. This is supported by the following security objectives:

- O.CONNECT ensures that the TOE is able to communicate with the necessary Certification Authorities, and that if it is a firewall or gateway proxy it cannot be bypassed.
- O.CERTAUTH ensures that there is adequate control of the issuance and revocation of public key certificates.
- O.SAFEKEY and O.KEYDIST ensure the initial and continued authenticity of public keys of the Certification Authorities used to check the validity of the public key certificates.

O.EAL supports O.AUTHCHECK, O.CERTCHECK and O.ENFORCE by providing appropriate confidence that these security objectives are upheld by the TOE.

#### **T.ALTERED**

*A compromise of assets occurs as a result of the execution of mobile code that originates from a trustworthy source, but has subsequently been altered by an attacker to act in a malicious manner.*

O.AUTHCHECK counters this threat by providing the means to detect whether signed mobile code has been altered since it was digitally signed. Altering the code will produce a different checksum from original code, causing the authentication check to fail. O.ENFORCE provides necessary support by ensuring that this check cannot be bypassed, and that the mobile code will not be executed if this check fails.

O.CONNECT ensures that the TOE is able to communicate with the necessary Certification Authorities, and that if it is a firewall or gateway proxy it cannot be bypassed.

O.CERTCHECK ensures that the digital certificate identifying the signatory of the mobile code is valid and has not been revoked. This is supported by the following objectives:

- O.CERTAUTH ensures that there is adequate control of the issuance and revocation of public key certificates.
- O.SAFEKEY and O.KEYDIST ensure the initial and continued authenticity of public keys of the Certification Authorities used to check the validity of the public key certificates.

O.EAL supports O.AUTHCHECK, O.CERTCHECK and O.ENFORCE ensuring that these security objectives are upheld.

## **T.SPOOF**

*A compromise of assets occurs as a result of the execution of mobile code that appears to originate from an trustworthy source, but which in fact has been written by an attacker to act in a malicious manner.*

O.CERTCHECK counters this threat by checking the validity of the public key certificate for the identified signatory of the mobile code. This will detect any attempt by an attacker to masquerade as a trustworthy individual since the check on the certificate will fail (the key used to sign the code is not associated with the individual). A check for revocation is enforced to ensure that a public key certificate will not be identified as valid in the event that an individual's private key is lost or stolen. O.ENFORCE provides essential support by ensuring that these checks cannot be bypassed and must succeed before the mobile code is executed.

The following security objectives also provide support to O.CERTCHECK:

- O.AUTHCHECK provides the mechanism by which the validity of the digital signatures of the signatory of the mobile code may be checked.
- O.CONNECT ensures that the TOE is able to communicate with the appropriate Certification Authorities, and that if it is a firewall or gateway proxy it cannot be bypassed.
- O.CERTAUTH ensures that there is adequate control of the issuance and revocation of public key certificates.
- O.SAFEKEY and O.KEYDIST ensure the initial and continued authenticity of public keys of the Certification Authorities used to check the validity of the public key certificates.

O.USER\_ED ensures that where the TOE is a desktop application that users are aware of the dangers of executing mobile code from an untrusted source, and that mobile code must therefore be suitably signed.

O.TRUSTSIGN ensures that the trust that may be placed in the signatories of mobile code has been determined, and thus, in particular, that the purported signatory can be trusted.

O.EAL supports O.AUTHCHECK, O.CERTCHECK and O.ENFORCE ensuring that these security objectives are upheld.

#### **T.NOREPUD**

*Mobile code originating from a source regarded as trustworthy is subsequently found to cause a compromise of assets, for which the originator denies any responsibility.*

O.AUTHCHECK counters this threat by providing the means to establish proof of the origin of mobile code, i.e. that the mobile code could only have been signed by the owner of the private key indicated by the digital signature.

O.CERTCHECK ensures that the digital certificate identifying the signatory of the mobile code is valid and has not been revoked. Hence the owner of the private key cannot claim with any credibility that his or her key has been lost or stolen and thereby deny responsibility for the mobile code. This is supported by the following objectives:

- O.CERTAUTH ensures that there is adequate control of the issuance and revocation of public key certificates.
- O.SAFEKEY and O.KEYDIST ensure the initial and continued authenticity of public keys of the Certification Authorities used to check the validity of the public key certificates.

O.EAL supports O.AUTHCHECK and O.ENFORCE ensuring that these security objectives are upheld.

#### **6.1.2**

#### **Security objectives are suitable to meet the OSPs**

The table below shows that each organisational security policy is met by at least one security objective.

OSP	Objectives									
	O.AUTHCHECK	O.CERTCHECK	O.ENFORCE	O.EAL	O.USER_ED	O.TRUSTSIGN	O.CERTAUTH	O.SAFEKEY	O.KEYDIST	O.CONNECT
P.DSV	x						x			
P.EAL				x						

**Table 4 - Correlation between OSPs and Objectives**

The following rationale demonstrates how the Organisational Security Policies are upheld by the security objectives:

**P.DSV** *Digital Signature Validation mechanisms shall be approved by the relevant cryptographic authority.*

O.AUTHCHECK (for the TOE) and O.CERTAUTH (for the CA) uphold this policy directly.

**P.EAL** *IT countermeasures shall be assured to a level equal to EAL3 as a minimum.*

O.EAL upholds this policy directly.

**6.1.3 Security objectives are suitable to uphold the assumptions**

The table below shows that each assumption is supported by at least one security objective.

Assumptions	Objectives									
	O.AUTHCHECK	O.CERTCHECK	O.ENFORCE	O.EAL	O.USER_ED	O.TRUSTSIGN	O.CERTAUTH	O.SAFEEKEY	O.KEYDIST	O.CONNECT
A.SOURCEPOL					x	x				
A.CHOKE										x
A.CERTAUTH										x

**Table 5 - Correlation between Assumptions and Objectives**

The following rationale demonstrates how the assumptions are supported by the objectives:

**A.SOURCEPOL** *It is assumed that the TOE is being used within an organisation that has a clearly defined policy for determining which other organisations and individuals can be considered to be trustworthy originators of mobile code.*

O.TRUSTSIGN upholds this assumption directly. O.USER\_ED is also relevant as this policy has to be communicated to the end-users.

**A.CHOKE** *If the TOE is a firewall or gateway proxy, it is assumed that it is deployed at the choke point between an untrusted external network and the trusted local network.*

O.CONNECT (point 2) upholds A.CHOKE directly. (O.ENFORCE is consistent with A.CHOKE as otherwise it might be possible for mobile code to be executed without checking the validity or even existence of its digital signatures, that is bypass of O.ENFORCE.)

**A.CERTAUTH** *There exists one or more Certification Authorities, with which the TOE can communicate, which issue public key certificates.*

O.CONNECT (point 1) upholds this assumption directly.

## 6.2 Security Requirements Rationale

This section demonstrates that the security requirements are appropriate, both in terms of their suitability to achieve the security objectives, and in collectively providing a mutually supportive and internally consistent whole.

### 6.2.1 Security requirements are suitable to achieve the security objectives

This section provides the correlation and justification of suitability between the objectives and the Security Functional Requirements. Iteration numbers of components are given where appropriate - if no iteration number is given then **all** iterations of that component help to achieve the security objective.

TOE Security Objectives	Security Requirement
O.AUTHCHECK	Proof of origin of mobile code FCO_NRO.1 Cryptographic operations: digital signature verification FCS_COP.1(1) Cryptographic operations: hash generation FCS_COP.1(2) Mobile code information flow control policy FDP_IFC.1 Mobile code information flow control rules FDP_IFF.1
O.CERTCHECK	Mobile code authentication with identity of guarantor FDP_DAU.2 Import of public key certificates FDP_ITC.1 Cryptographic operations: digital signature verification FCS_COP.1(1) Cryptographic operations: hash generation FCS_COP.1(2) Mobile code information flow control policy FDP_IFC.1 Mobile code information flow control rules FDP_IFF.1
O.ENFORCE	Non-bypassability of the TOE Security Policy FPT_RVM.1 TSF domain separation FPT_SEP.1
O.EAL	EAL3 or EAL4, as appropriate.

**Table 6 - Correlation between Objectives for the TOE and SFRs**

O.AUTHCHECK *The TOE shall employ an approved Digital Signature Validation mechanism to determine the authenticity and integrity of mobile code that is downloaded from the untrusted network.*

FCO\_NRO.1 ensures that the TOE is able to verify digital signatures for downloaded mobile code, thereby determining the authenticity and integrity of the code.

FCS\_COP.1(1) and FCS\_COP.1(2) ensure that the signature verification is performed with an approved digital signature validation mechanism consisting of: an approved digital signature algorithm with appropriate key sizes, and an approved hash generation algorithm.

FDP\_IFC.1 and FDP\_IFF.1 ensure that checks on the validity of the digital signatures are enforced as part of the checks on downloaded mobile code.

O.CERTCHECK *The TOE shall check the validity of public key certificates before confirming that the digital signature on downloaded mobile code is valid. This shall include a check for certificate revocation and against its validity period.*

FDP\_ITC.1 ensures that only valid public key certificates may be imported into the TSC.

FDP\_DAU.2 ensures that the correct public key certificates may be obtained from the CAs, and that they can be appropriately validated for signature verification, validity period and revocation.

FCS\_COP.1(1) and FCS\_COP.1(2) provide the means of validating the digital signatures of the CA that acts as proof of the authenticity and integrity of the public key certificates.

FDP\_IFC.1 and FDP\_IFF.1 ensure that checks on the validity of the appropriate public key certificates are enforced as part of the checks on downloaded mobile code.

O.ENFORCE *The TOE shall ensure that downloaded mobile code cannot be executed before a successful check on the validity of digital signatures and the associated public key certificates.*

FPT\_RVM.1 ensures that the mobile code authentication (validation of the appropriate public key certificate and validation of the digital signature of the code) by the TSF is always invoked and succeeds before the import of mobile code into the TSC is allowed to proceed, i.e. the checks cannot be bypassed.

FPT\_SEP.1 ensures that the TSF maintains a separate security domain for its execution that protects it from interference and tampering that might otherwise permit circumvention of the mobile code authentication checks.

IT Environment Security Objectives	Security Requirement
O.CERTAUTH	Cryptographic key generation FCS_CKM.1 Cryptographic key destruction FCS_CKM.4 Cryptographic operation FCS_COP.1(1) FCS_COP.1(2) Revocation FMT_REV.1

**Table 7 - Correlation between Objectives for the TOE and SFRs**

O.CERTAUTH *The Certification Authorities shall exercise appropriate controls over the issuing and revocation of public key certificates to ensure that the owner of the public key is reliably identified. This shall include taking appropriate steps to protect the keys used by the CA to digitally sign such certificates, which shall be performed in accordance with P.DSV.*

FCS\_CKM.1 ensures that the CA has the capability of generating its own public/private key pair used for signing the public key certificates it issued. FCS\_CKM.4 ensures that the CA has a means of securely destroying such keys when they are no longer needed.

FCS\_COP.1(1) and FCS\_COP.1(2) ensure that the digital signature and hash generation are performed by the CA in accordance with the requirements of the relevant cryptographic authorities.

FMT\_REV.1 ensures that the CA performs revocation of public key certificates in a secure manner.

Note that there are also procedural aspects to this security objective, governing the operation of the CA. Such aspects are outside the scope of this PP.

## 6.2.2 Security requirements are mutually supportive

### 6.2.2.1 Dependency analysis

The following table shows the dependencies between the SFRs. The rows list all the components included in this PP to specify the SFRs. The columns identify the dependencies of those components as specified in [CC2].

	FIA_UID.1	FDP_ACC.1	FDP_IFC.1	FDP_IFF.1	FDP_ITC.1	FMT_MSA.1	FMT_MSA.2	FMT_MSA.3	FMT_SMR.1	FCS_CKM.1	FCS_CKM.4
FCO_NRO.1	x										
FCS_COP.1(1)		i	i		o	i	x		i	o	x
FCS_COP.1(2)		i	i		o	i	x		i	o	x
FDP_DAU.2	x										
FDP_IFC.1	i			x		i		i	i		
FDP_IFF.1	i		x			i		i	i		
FDP_ITC.1	i		x	i		i		x	i		
FPT_RVM.1											
FPT_SEP.1											

**Table 8 - Correlation between Assumptions and Objectives**

Key x – direct dependencies  
i – indirect dependencies  
o – optional dependencies

All dependencies highlighted in the above table are satisfied by the TOE SFRs, with the following exceptions:

- The dependencies of FCO\_NRO.1 and FDP\_DAU.2 on FIA\_UID.1 do not need to be satisfied because the individual signing the mobile code is not a user of the TOE.

- The dependencies of FCS\_COP.1(1) and FCS\_COP.1(2) on FMT\_MSA.2 do not need to be satisfied since the keys involved in the digital signature verification are not generated or stored by the TOE. This also accounts for the dependency on FCS\_CKM.4 (there are no keys stored by the TOE that require secure destruction) as well as the indirect dependencies on FMT\_MSA.1, FMT\_SMR.1, FIA\_UID.1, FDP\_ACC.1 and FDP\_IFC.1. Whilst FCS\_CKM.4 is a requirement on the CA, this relates only to the secure handling of the CA's private key.
- The dependency of FCS\_COP.1(1) on FDP\_ITC.1 is satisfied through the import controls on public key certificates. No keys need to be imported for the hash generation covered by FCS\_COP.1(2), and hence there is no need for this dependency to be satisfied. FCS\_CKM.1 is a requirement on the CA, but this relates only to the generation of the CA's public/private key pair.
- The dependency of FDP\_IFF.1 on FMT\_MSA.3 does not need to be satisfied since there is no notion of assignment of default values of digital signatures and public key certificates, which are the only security attributes used in the information flow control decisions. This also accounts for all of the indirect dependencies of FDP\_IFF.1 and FDP\_IFC.1.

Dependencies between security assurance requirements are satisfied because they are defined in terms of a self-contained assurance package (EAL3 or EAL4 as appropriate), with no augmentations.

#### 6.2.2.2 Security requirements form a mutually supportive and consistent whole

The SFRs divide into three groups, which together provide the capability for mobile code authentication to be enforced before mobile code may be downloaded. These are all in support of the mobile code information flow control SFRs FDP\_IFC.1 and FDP\_IFF.1:

1. The first group of security functional requirements, namely FCO\_NRO.1, FCS\_COP.1(1) and FCS\_COP.1(2) provide the capability and policy for extracting and verifying the digital signature of the mobile code using approved cryptographic algorithms.
2. The second group of security functional requirements, namely FDP\_DAU.2 and FDP\_ITC.1 provide the capability and policy for extracting and verifying the public certificate from the CA for the supposed signatory of the code. FCS\_COP.1(1) and FCS\_COP.1(2) provide support by ensuring the means of verifying the digital signature of the CA on the public key certificates.

3. The third group provide support by ensuring that the mobile code authentication checks cannot be bypassed or tampered with. FPT\_RVM.1 enforces the requirement that the mobile code be authenticated before the code may be imported, whilst FDP\_SEP.1 provides separation of the trusted and untrusted security domains.

These different functional areas support each other forming a cohesive and consistent whole.

The Security Assurance Requirements comprise a set of mutually supportive requirements as they correspond to a self-contained assurance package (EAL3 or EAL4 as appropriate). All such requirements by definition support the SFRs by providing confidence that the TOE satisfies those requirements. Therefore the security requirements form a cohesive and consistent whole.

### 6.2.3 **Security assurance requirements are appropriate**

The minimum evaluation assurance level for this PP, namely EAL3, is selected because it trivially satisfies the security objective O.EAL (which is, itself, mandated by the OSP P.EAL).

An assurance level of EAL4 clearly satisfies this minimum requirement. It is identified as an option in this PP document since it may be more appropriate for particular types of environment. EAL4 represents the highest EAL that can be practically achieved without requiring substantial specialist knowledge, skills and other resources.

### 6.2.4 **Strength of Function claim is appropriate**

The SOF claim of **SOF-medium** is consistent with the security objectives since it is commensurate with an assurance level of EAL3 or EAL4. The appropriateness of the assurance level has already been justified.