

---

# I-0339: Assurance Of RVM Is By Testing And Design Analysis

---

TYPE: NIAP Interpretation  
NUMBER: I-0339  
STATUS: Approved but Pending Rescission  
REASON: Paragraphs 1570 and 1571 of the CEM note that there are some cases where testing through the interface is not completely possible, and that other means may be necessary. These paragraphs were not noted when I-0338 was originally approved. In light of them, however, I-0338 is an unnecessary interpretation, and should be rescinded.

TITLE: Assurance Of RVM Is By Testing And Design Analysis

EFFECTIVE DATE: 2000-03-27

SOURCE REFERENCE: CC v2.1 Part 2 Subclause 10.10 FPT\_RVM  
CC v2.1 Part 2 Subclause J.10 FPT\_RVM

RELATED TO:  
[I-0382](#) TSF Architectural Protections Are Really Assurances  
CCIMB ENTRY: CCIMB-INTERP-0100

## STATEMENT

The following provides technical guidance regarding the element FPT\_RVM.1.1: "The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed."

Assurance that FPT\_RVM.1.1 is satisfied is achieved through a combination of testing and design analysis.

## SPECIFIC INTERPRETATION:

To address this interpretation, the following should be added in the Part 2 Annex for FPT\_RVM, at the end of the introductory paragraphs of Annex J.10:

In order to provide assurance that this element is satisfied, the developer must provide a convincing argument that the design provides the enforcement, with this argument verified through testing. Foundations of such argument generally revolve around the construction of the interface to the TSF (e.g., call gates, network cards) and the limitations placed on those interfaces.

As this interpretation moves some of the verification burden from developer interface testing to the evaluator, as well as imposing additional requirements for developer arguments, changes to the CEM will be required.

## PROJECTED IMPACT:

Negligible impact anticipated.

## SUPPORT:

Most of the CC functional requirements are completely testable through the TSF interface. However, that is not true for this element. Determining the internal invocation sequence underlying a call is difficult to assure solely through testing. In such a case, examination of the design must come into play.

In order to provide assurance that this element is satisfied, the developer must provide a convincing argument that the design provides the enforcement, with this argument verified through testing. Foundations of such argument generally revolve around the construction of the interface to the TSF (e.g., call gates, network cards) and the limitations placed on those interfaces. In addition, the nature of the convincing argument should be

based on the assurance package which has been chosen to be associated with the functional requirements. The depth (i.e., how much detail is involved) is dependent on the nature of assurance being pursued (i.e., the lower the level of assurance the less detail required).

The elements from Part 2 of the CC are usually testable through the TSF interface. This is not so for this element.