Common Criteria

Security Target

(ST)

For

IAI/MLM Autonomous Air Combat Maneuvering Instrumentation

(AACMI)

Trusted Data Guard (TDG) v1.0

Issue: E

References

[CCp1]	Common Criteria for IT Security Evaluation, Part 1, v3.1r4, September 2012
[CCp2]	Common Criteria for IT Security Evaluation, Part 2, v3.1r4, September 2012
[CCp3]	Common Criteria for IT Security Evaluation, Part 3, v3.1r4, September 2012
[CEMe]	Common Methodology for IT Security Evaluation, v3.1r4, September 2012
[AACMISFR]	USAF policy letter “AACMI Security Requirements” dated 13 December 2000
[AACMIMEM]	Memorandum for AAC/WMRR: “AACMI Policy Clarifications”, date unknown
[NATO]	NATO Industrial Advisory Group, “AACMI Interoperability Study.” NIAG/SG-71

Abbreviations

AACMI	Autonomous Air Combat Maneuvering Instrumentation
BIT	Built-In Test
CAT	Certified Allocation table
FTG	Firmware Trusted Guard
HTG	Hardware Trusted Guard
IAI	Israel Aerospace Industries
IMM	Integrated Main Module
MLM	A subsidiary company of IAI
RWD	Residual (non volatile) memory Write Disable
RSM	Removable Storage Media
SDW	Security Double Wall
SiPP	Simulation Platform PCMCIA
SF	Security Function
SFV	security filtering and verification
SUC	Software Upgrade Connector
SUJ	Software Update Jumper
TDG	Trusted Data Guard
UDP	User Datagram Protocol

Content

1 ST Introduction

1.1 ST and TOE References

This is version E of the Security Target for the IAI/MLM Autonomous Air Combat Maneuvering Instrumentation (AACMI) Trusted Data Guard (TDG) v1.0.

The remainder of this ST will refer to the TOE as TDG.

1.2 TOE Overview

The IAI/MLM AACMI system (Autonomous Air Combat Maneuvering Instrumentation) is used for training fighter pilots. The IAI/MLM AACMI system consists of a pod (referred throughout this document as 'EHUD' (see Figure 1) that is attached to a fighter aircraft to continuously record and broadcast relevant training information such as position, speed, angle-of-attack, amount of weapons left, the firing of simulated weapons, etc. The combination of the training information is then used to build and maintain a coherent and realistic view of the current situation during the training flight.

Figure 1 The EHUD pod

The training information is recorded on a removable RSM or SIPP flash memory card and broadcast, via an unclassified AACMI RF Data Link network (referred to throughout this document as 'EHUD DL'), to other IAI/MLM AACMI-systems participants.

All the EHUD activities are managed and controlled by an on-board Data Processor (referred throughout this document as 'IMM MK5')

The information to be broadcasted via the AACMI RF Data Link can contain unauthorized information[1], which would allow attackers to derive detailed weapons performance information. Also, residual information left in the IMM MK5 memories after completing the exercise and shutdown pod power might facilitate attackers to get said restricted information as well.

The TDG consists of two components in EHUD pod: the SDW (Security Double Wall) and IMM MK5 (Integrated Main Module).

[1] Unauthorized information refers to sensitive information that is not allowed to be transmitted according to the SDW CAT table

The general Pod Architecture including the connection to a fighter aircraft is shown in Figure 2. It shows the components:

1. The *SDW* which receives 'data objects' from the IMM MK5 and 'data objects' from the Receiver. The *SDW* contains two filters.
2. The *F1 filter* (Firmware Trusted Guard) removes unauthorized information according to the rules defined in the CAT table (see below).
3. The *F2 filter* (Hardware Trusted Guard) in the SDW verifies whether the data received from F1 indeed does not contain unauthorized information (according to the rules defined in the CAT table, see below), and if so forwards it as authorized data to the transmitter.
4. The *IMM MK5*, the main Data Processor in EHUD pod: It contains Software and a RWD.
5. The *Software* on the IMM MK5 generates data objects, stores them on RSM or SIPP flash memory cards and sends them to the F1 Filter in the SDW.
6. The *RWD*, a Residual (non-volatile) memory Write Disable mechanism in IMM MK5 prevents write access (unless a SUC is inserted) to the non-volatile storage of the IMM MK5.
7. The *Transmitter*, for transmitting 'data objects' to other pods or ground stations.
8. The *Receiver*, for receiving 'data objects' from other Pods or ground stations.
9. The Protocol (not shown) used by the system is a so called UDP "Placement Oriented" one, where the location of the data has a central role in its interpretation. Moreover, the interpretation is tightly associated to the protocol type, so that any information not packed in the UDP format will be rejected by the system. Thus, the TDG addresses only data in the UDP protocol format and ignores any information not given in the UDP protocol.

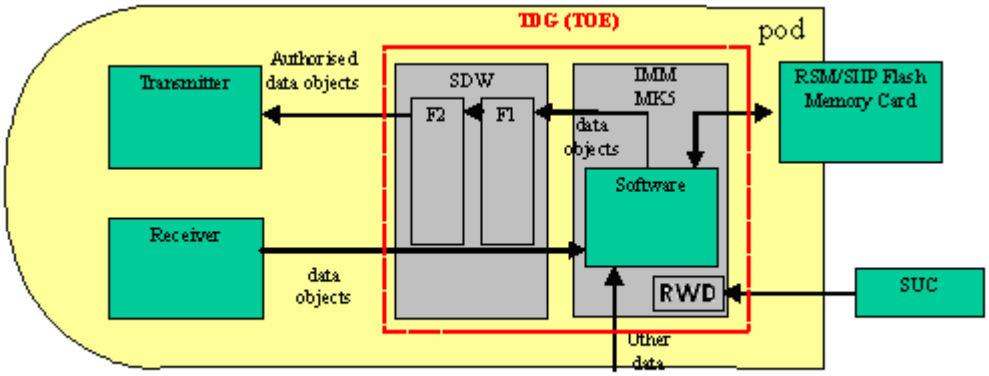

Figure 2 General Pod Architecture

The TOE consists of the SDW and the IMM MK5 but does not include Software in IMM MK5 and the CAT tables in the SDW.

The exact rules to determine which information is unauthorized are defined in a classified customized CAT table. The CAT is a Certified Allocation Table embedded in the firmware of the SDW, and holds patterns which identify certified data objects allowed to pass through to the transmitter. The CAT is prepared together with the final user and provides a fast and modular way to configure the SDW security filter and match it to the specific security requirements of the end user. Non-volatile memory components in the IMM MK5 are protected against accidental writing by the Software. RWD is an electrical circuit that is intended to enable writing to some of the non-volatile memory of the IMM MK5 with the intention of software updates.

1.3 TOE Description

1.3.1 Physical scope

The TOE consists of the following:

Category	Description	Part number	Version	Media
Hardware	Security Double Wall (SDW)	B-18600-***-300000	N/A	Physical item
Hardware	IMM MK5	B-18500-+++-300000	N/A	Physical item
Software	FTG software	B-00707-500000	A	Software code
Firmware	FTG firmware	B-610-00197	#	Firmware code
Firmware	HTG firmware	B-610-00198	#	Firmware code
Documentation	User Guide - UG	B-00119-100000	Issue B	Paper document
Documentation	Tamper Label Track Log	B-720-00125	Issue E	Paper document
Documentation	HW Design document For SDW FPGAs	B-18617-300000	Issue C	Paper document

Notes:

- "***" means any 3 digits. This value reflects the embedded CAT table and non-security-relevant hardware changes.
- "+++" means any 3 digits. This value reflects non-security-relevant hardware changes.
- “#” means any digit. This value reflects the embedded CAT table.
- The policy for the use of wildcards for the embedded CAT table and non-security-relevant hardware changes, and the process to implement this wildcard policy during development, is included within the scope of the evaluation.

The following are not included within the boundaries of the TOE:

- Any software in the IMM MK5.
- The CAT table embedded in the SDW (FTG and HTG firmware).
- RSM or SIPP data storage card that is inserted into a slot in the IMM MK5.
    - The card contains classified information and must be removed from the EHUD pod upon leaving the Flight Environment;
- The cable harness used to connect the various components in the EHUD pod.
    - The TOE contains physically separated parts which communicate over these cables.

1.3.2 Logical scope

The major security features provided by the TOE are:

- The TOE filters: the TOE ensures that the end user filtering policy, as depicted in the CAT table, is followed: no unauthorized information is passed to the Transmitter;
- The TOE filters under failure: the TOE tests itself upon start-up and, when a failure is detected, ensures that no unauthorized information is passed to the Transceiver. In addition, when the TOE undergoes a single failure during operation, it ensures that still no unauthorized information is passed to the Transmitter;
- The TOE is tamper-evident: if it has been physically tampered with, this will be detectable;
- The TOE clears residual information in its non-volatile memory upon power down.
- The SDW protects against writing to its non-volatile memory.
- The IMM MK5 protects against **accidental** writing to its volatile memory.

1.3.3 Organisational scope

The IMM MK5 protects against **accidental** writing to non-volatile memory only. Therefore it is vital to ensure that the organisations that handle any returned units are trusted to handle the confidential information contained in the IMM MK5.

2 Conformance Claims

This ST conforms to:

- Common Criteria for Information Technology Security Evaluation
  - Part 1: Introduction and general model, September 2012, Version 3.1 Revision 4 (CCMB-2012-09-001)
  - Part 2: Security functional components, September 2012, Version 3.1 Revision 4 (CCMB-2012-09-002)
  - Part 3: Security assurance components, September 2012, Version 3.1 Revision 4 (CCMB-2012-09-003)
- CC Part 2 as CC Part 2 conformant
- CC Part 3 as CC Part 3 conformant

This ST conforms to no Protection Profile.

This ST conforms to EAL 4+ with ASE_TSS.2, and to no other packages.

3 Security Problem Definition

This section starts out with a definition of all subjects, objects and operations relevant to the TOE. It then describes the different environments in which the TOE is intended to be used, and the relevant security aspects of each environment.

- Subjects

Regular subjects

Pod	A metal container containing (among others) the Receiver, the Transmitter and the TOE.
IMM Mk5 (SW)	An IT entity capable of generating[2] Data Objects and sending these to the Transmitter.
Transmitter[3]	An IT entity capable of receiving Data Objects from the TOE and transmitting them on a RF datalink
O-Level Staff	Administrative personnel in the O-Level Environment (see section 3.3)
I-Level Staff	Administrative personnel in the I-Level Environment (see section 3.4)
D-Level Staff	Administrative personnel in the D-Level Environment (see section 3.5)

Threat agents

TOE Failure	A failure in the TOE. TOE Failures may be spontaneous or caused by other Threat Agents.
Pod Failure	A failure in the Pod. Pod Failures may be spontaneous or caused by other Threat Agents.
Physical Attacker	A person, who, given physical access to the TOE is able to logically or physically alter it, and has the tools and skills to do so effectively.
Remote Attacker	A person, with a transmitter and receiver[4] capable of interacting with the pod, and the skills to use these.

Notes:

- *The motivation of Physical and Remote Attacker are to break the confidentiality of the Unauthorized Information (which is the only asset). Pod Failure and TOE Failure have no motivation, but threaten the same asset.*
- *The out-of-scope software is not considered a threat agent.*

- Objects

Data Object	A block of data suitable to be used on the AACMI RF data link network generated by the Software in IMM MK5 and sent to the Transmitter. Data Objects consist of Messages, which themselves contain Parameters. Data Objects are UDP packets only.

[2] Software in IMM MK5 may either generate Data Objects by itself, or pass them through from other parts in the Pod or in the aircraft that the Pod is attached to. As far as the TOE is concerned there is no difference.
[3] Formally the Transmitter is not a subject: it passively receives Data Objects and transmits these. It is listed here for completeness and clarity.
[4] These may be the same devices as a Receiver (capital R) and Transmitter (capital T) but need not be.

The information in a Data Object has a security attribute called "Unauthorized/Authorized" which can take two values: Authorized or Unauthorized. These values are assigned to the information in a Data Object as follows:

Unauthorized Information	*1) All Messages with UDP protocol not listed in the CAT table;* *2) All Messages with UDP protocol listed in the CAT table for which it is stated that "This message shall not be transmitted"* *3) Any Parameter of a UDP Message listed in the CAT table for which it is stated that "The parameter shall not be transmitted"*
Authorized Information	*All other information*

- Operations

Pass Through	An operation where the TOE receives a Data Object from the Software in IMM MK5 and sends this Data Object to the Transmitter.

3.1 Environments

As far as security is concerned, the TOE exists in four distinct security environments:

- **The Flight Environment**: the TOE is connected to an Aircraft;
- **The O-Level Environment**: the TOE is stored at an Airbase;
- **The I-Level Environment**: the TOE is at an Intermediate Repair Facility;
- **The D-Level Environment**: the TOE is at the IAI/MLM.

The normal[5] transitions between these environments are provided in Figure 3. The four environments are described in more detail in the following sections. Each environment contains its own threats, assumptions and organizational security policies (OSP).

[5] There also exist other transitions such as: crashing an operational aircraft, stealing an operational aircraft (including the pod), but these are considered to be exceedingly unlikely.

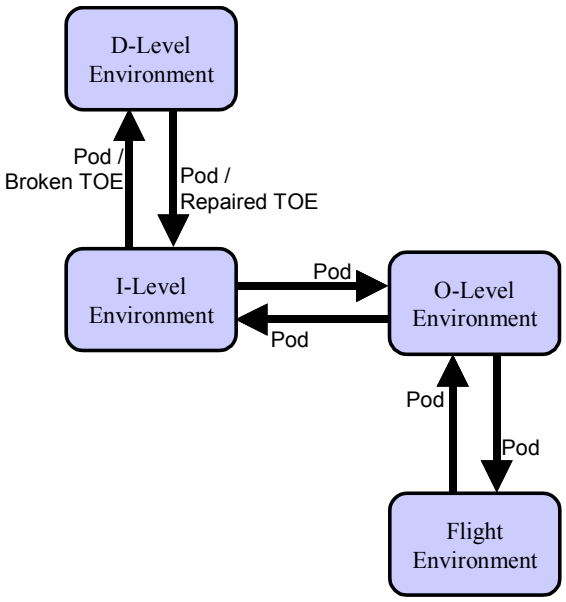

Figure 3 Possible transitions between environments

3.2 The Flight Environment

The Flight Environment is the environment in which the TOE is operational. In this environment the TOE is contained in a Pod, and the Pod is attached to the wing of an operational[6] aircraft. The aircraft sends information to the IMM MK5, and based on this, and other data, the Software of IMM MK5 generates Data Objects and sends these to the TOE (SDW part). The TOE (SDW) will not Pass Through Data Objects containing Unauthorized Information to the Transmitter.

3.2.1 Assumptions on the Flight Environment

A.F.LOCATION The TOE (SDW) assumes that it is contained in a Pod, and is connected between TOE (IMM MK5) and the Transmitter. The TOE assumes that the Pod is connected to an operational aircraft.

A.F.ONLY_WAY The TOE (SDW) assumes that it is the only communication path between the TOE (IMM MK5) and the Transmitter, and that no other transmitters in the Pod transmit Data Objects.

A.F.TRANSPORT_TO_O The TOE assumes that adequate procedures exist to securely transfer the Pod to the O-Level Environment and that the TOE is transported to no other environment than the O-Level Environment.

A.F.TRUSTED_SOFTWARE The author of the software running on the IMM MK5 is assumed to be non-malicious, the software does not **intentionally** write data objects to the non-volatile memory of the IMM MK5.

3.2.2 OSPs in the Flight Environment

P.F.CORRECT_OUT The TOE must not Pass Through Data Objects containing Unauthorized Information to the Transmitter.

[6] An operational aircraft is an aircraft being prepared for take-off, during flight and landing.

3.2.3 Threats in the Flight Environment

T.F.PHYSICAL	Physical Attacker may physically or logically modify the TOE, causing it to Pass Through Data Objects containing Unauthorized Information, either immediately, or at some point in the future.
T.F.REMOTE	Remote Attacker may logically modify the TOE, causing it to Pass Through Data Objects containing Unauthorized Information, either immediately, or at some point in the future.
T.F.TOE_FAIL	TOE Failure may cause the TOE to Pass Through Data Objects containing Unauthorized Information, either immediately, or at some point in the future.
T.F.POD_FAIL	Pod Failure may cause the TOE to Pass Through Data Objects containing Unauthorized Information, either immediately, or at some point in the future.

3.3 The O-Level Environment

The O-Level Environment is typically an Airbase. In this environment the TOE is contained in a Pod, and this Pod is stored on the Airbase.

3.3.1 Assumptions on the O-Level Environment

A.O.LOCATION	The TOE assumes that it is contained in a Pod.
A.O.CLOSED_POD	The TOE assumes that in the O-Level environment, the Pod is never opened.
A.O.NO_OFFERING	The TOE assumes that no Data Objects are offered to the TOE in the O-Level Environment.
A.O.PERSONNEL	The TOE assumes that the O-Level Staff has adequate clearance, is adequately trained to perform its duties, follows all supplied guidance and is reasonably trustworthy.
A.O.TRANSPORT	The TOE assumes that adequate procedures exist to securely transfer the Pod to the Flight Environment and the I-Level Environment and that the TOE is transported to no other environment than the Flight Environment or the I-Level Environment.
A.O.TRUSTED_ENV	The TOE assumes that the Pod will be handled and protected consistent with other assets in the O-Level environment of similar value. This means that the TOE assumes that adequate measures exist to protect against unauthorized physical access to the TOE.

3.3.2 OSPs in the O-Level Environment

P.O.INSPECT	The O-Level Staff will inspect (monitor) the Pod for failures, and if a failure is found the Pod is transferred to the I-Level Environment.

3.3.3 Threats in the O-Level Environment

T.O.PHYSICAL	Physical Attacker may physically or logically modify the TOE, causing it to pass through Data Objects containing Unauthorized Information, either immediately, or at some point in the future.
T.O.RESIDUAL	Physical Attacker may read Unauthorized Information that has been retained by the TOE when it was in the Flight Environment.

3.4 The I-Level Environment

The I-Level Environment is an Intermediate Repair Facility, which is may be on the same location as the O-Level environment (i.e. an Airbase).
When the TOE is broken, the I-level Environment can only replace the TOE in the Pod with a new identical TOE: the TOE is never repaired in this environment. The I-Level Environment receives broken-down Pods (containing the TOE) from the O-Level Environment. It then disassembles the Pod, diagnoses the Pod, repairs/replaces faulty non-TOE parts or replaces the TOE and sends the Pod back to the O-Level Environment. Any parts that were replaced are sent to the D-level Environment for further repair or disposal.

3.4.1 Assumptions on the I-Level Environment

A.I.NO_OFFERING	The TOE assumes that no Data Objects are offered to the TOE in the I-Level Environment.
A.I.PERSONNEL	The TOE assumes that the I-Level Staff has adequate clearance, is adequately trained to perform its duties, follows all supplied guidance and is reasonably trustworthy and competent.
A.I.TRANSPORT	The TOE assumes that adequate procedures exist to securely transfer the Pod to the D-Level Environment and the O-Level Environment and that the TOE is transported to no other environment than the D-Level Environment or the O-Level Environment.
A.I.TRUSTED_ENV	The TOE assumes that the TOE and the Pod will be handled and protected consistent with other assets in the I-Level environment of similar value. This means the TOE assumes that adequate measures exist to protect against unauthorized physical access to the TOE.

3.4.2 OSPs in the I-Level Environment

P.I.INSPECT	All TOEs brought into the I-Level Environment from the D-Level Environment will be inspected by the I-Level Staff to determine whether physical tampering has occurred. If this has occurred, the TOE will be handled according to the appropriate procedures.
P.I.NO_OPEN	The TOE is normally never physically opened in the I-Level Environment. If TOEs are found to be malfunctioning, the I-Level Staff will send the TOE to the D-Level Environment.
P.I.SEALED	The TOE is received sealed from the D-Level environment.

3.4.3 Threats in the I-Level Environment

T.I.PHYSICAL	Physical Attacker may physically or logically modify the TOE, causing it to Pass Through Data Objects containing Unauthorized Information, either immediately, or at some point in the future.
T.I.RESIDUAL	Physical Attacker may read Unauthorized Information that has been retained by the TOE when it was in the Flight Environment.

3.5 The D-Level Environment

The D-Level Environment is the environment of the IAI/MLM. It produces Pods (containing the TOE) and sends these to the I-Level Environment. It also receives faulty parts (including TOEs) from the I-Level Environment. It sends back repaired or new parts (including TOEs) to the I-Level Environment.

3.5.1 Assumptions on the D-Level Environment

None.

3.5.2 OSPs in the D-Level Environment

None.

3.5.3 Threats in the D-Level Environment

T.D.RESIDUAL Physical Attacker may read Unauthorized Information that has been retained by the TOE when it was in the Flight Environment.

4 Security Objectives

These security objectives describe how the threats described in the previous section will be addressed. It is divided into:

- The Security Objectives for the TOE, describing what the TOE will do to address the threats
- The Security Objectives for the Operational Environment, describing what other entities must do to address the threats

A rationale that the combination of all of these security objectives indeed addresses the threats may be found in section 7.1 of this Security Target.

4.1 Security objectives for the TOE

4.1.1 Functional Security Objectives

O.SF.CLEAR_WHEN_DONE
The TOE shall delete all Unauthorized Information from itself before leaving the Flight Environment.

O.SF.DATA_FILTER
The TOE shall not Pass Through Data Objects containing Unauthorized Information to the Transmitter.

O.SF.FAILSAFE
The TOE shall test itself at start-up. If it fails this test, the TOE shall not Pass Through any Data Objects to the Transmitter. In addition to this, a single failure during operation of the TOE shall not cause the TOE to Pass Through Data Objects containing Unauthorized Information to the Transmitter

O.SF.NO_REPROGRAM
The TOE cannot be altered through any interaction through its interface with the Transmitter.

O.SF.TAMPER_EVIDENT
The TOE shall be encased in tamper-evident covers.

4.2 Security objectives for the Operational Environment

To clarify this section, we have divided these objectives into three groups[7], to distinguish between the different environments:

1. Security Objectives for the Flight Environment;
2. Security Objectives for the O-Level Environment;
3. Security Objectives for the I-Level Environment.

4.2.1 Security Objectives for the Flight Environment

OE.F.ONLY_WAY
The TOE is the only communication path between the TOE (IMM MK5) and the Transmitter. No other transmitters in the Pod transmit Data Objects.

OE.F.TOE_LOCATION

7 There are no security objectives for the Operational Environment for the D-Level Environment.

The TOE is contained in a Pod. In this Pod the TOE (SDW) is connected between TOE (IMM MK5) and the Transmitter. The Pod is connected to an operational aircraft.

OE.F.TRANSPORT_TO_O
The O-Level Staff is responsible for the transport of the Pod (containing the TOE) to the O-Level environment. The O-Level Staff shall have procedures that maintain the security when the TOE is transported from the Flight Environment to the O-Level Environment. These procedures shall include the removal of any components from the Pod that may include Unauthorized Information.

OE.F.TRUSTED_SOFTWARE
The developer of the software running on the IMM MK5 is non-malicious. The developer (through his software) does not **intentionally** write to the non-volatile memory of the IMM MK5.

OE.F.LOGICAL_PROTECTION
The POD provides logical protection against logical modification of the software running on the TOE.

4.2.2 Security Objectives for the O-Level Environment

OE.O.INSPECT
The O-Level Staff shall have procedures to inspect (monitor) the Pod for failures, and if a failure is found to transfer the Pod to the I-Level Environment.

OE.O.NO_OFFERING
The O-Level Staff shall have procedures that provide assurance that no Data Objects are offered to the TOE in the O-Level Environment.

OE.O.PERSONNEL
The O-Level Staff has adequate clearance, is adequately trained to perform its duties, follows all supplied guidance and is reasonably trustworthy and competent.

OE.O.TOE_LOCATION
The TOE is contained in a Pod. In this Pod the TOE (SDW) is connected between TOE (IMM MK5) and the Transmitter.

OE.O.TOE_TRANSPORT
The O-Level Staff shall have procedures that provide assurance that the security of the Pod (containing the TOE) is maintained when the TOE is being distributed to the Flight Environment and the I-Level Environment.

OE.O.TRUSTED_ENV
The O-Level Staff shall have procedures that provide assurance that the Pod is handled and protected consistent with other assets in the O-Level environment of similar value, including restriction of unauthorized physical access to the TOE.

4.2.3 Security Objectives for the I-Level Environment

OE.I.INSPECT_FROM_D
All TOEs brought into the I-Level Environment shall be inspected when received from the D-Level Environment. The inspection shall include a functional check and a check to determine whether physical

tampering has occurred. If tampering is determined the TOE shall be handled according to the appropriate procedures.

OE.I.NO_OFFERING
The I-Level Staff shall have procedures that provide assurance that no Data Objects are offered to the TOE in the I-Level Environment.

OE.I.NO_OPEN
The I-Level Staff shall have procedures to prevent the TOE be opened physically. If a TOE is found to be malfunctioning, the I-Level Staff sends the TOE to the D-Level Environment.

OE.I.PERSONNEL
The I-Level Staff has adequate clearance, is adequately trained to perform its duties, follows all supplied guidance and is reasonably trustworthy.

OE.I.TOE_TRANSPORT
The I-Level Staff shall have procedures that provide assurance that the security of the TOE is maintained when the TOE is being distributed to the O-Level Environment or the D-Level Environment.

OE.I.TRUSTED_ENV
The I-Level Staff shall have procedures that provide assurance that the Pod is handled and protected consistent with other assets in the I-Level environment of similar value, including restriction of unauthorized physical access to the TOE.

5 Security Requirements

5.1 Extended components definition

There are no extended components defined.

5.2 Definitions

The following notational conventions are used in the requirements. Operations are indicated in **bold**, except refinements, which are indicated in **bold *italic***. In general refinements were applied to clarify requirements and/or make them more readable. Iterations were indicated by adding three letters to the component name.

5.3 Security Functional Requirements

For clarity, the SFRs are divided into three sections:

1. *Security Filter*: Describing the main functionality of the TOE: the filtering of Unauthorized Information from Data Objects;
2. *Residual information*: How the TOE deletes Unauthorized Information from itself after operation;
3. *TOE integrity*: How the TOE protects itself against failures and attempts to change the TOE.

5.3.1 Security Filter

FDP_IFC.1 Subset information flow control

FDP_IFC.1.1 The TSF shall enforce the **Filter Policy** on [**Software (IMM MK5) (subject)], [information in Data Objects], [Pass Through (operation)].**

FDP_IFF.1 Simple security attributes

FDP_IFF.1.1 The TSF shall enforce the **Filter Policy** based on the following types of subject and information security attributes: **Unauthorized/Authorized**.

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

- **Pass Through is allowed if a Data Object contains no Unauthorized Information flow**
- **Pass Through is allowed if a Data Object contains Unauthorized Information but this Unauthorized Information is removed.**

FDP_IFF.1.3 -[8]

FDP_IFF.1.4 -

FDP_IFF.1.5 -

[8] FDP_IFF.1.3 to FDP_IFF.1.5 were refined away as they were all completed with empty lists.

5.3.2 Residual information

FDP_RIP.1 Subset residual information protection

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource ***in the TOE*** is made unavailable upon the **deallocation of the resource *when leaving the Flight Environment*[9] from Data Objects and data derived from Data Objects**.

5.3.3 TOE integrity

FPT_FLS.1 Failure with preservation of secure state

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: **single failures**.

FPT_PHP.1 Passive detection of physical attack

FPT_PHP.1.1 The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.

FPT_PHP.1.2 The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.

FMT_MSA.1 Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the **Filter Policy** to restrict the ability to **perform any operation on** the security attributes **Authorized/Unauthorized** to **nobody**.

FMT_MSA.3 Static attribute initialization

FMT_MSA.3.1 The TSF shall enforce the **Filter Policy** to provide **restrictive** default values for security attributes that are used to enforce the ***Filter Policy*[10]**.

FMT_MSA.3.2 The TSF shall allow **nobody** to specify alternative initial values to override the default values when an object or information is created.

FPT_TST.1 TSF testing

FPT_TST.1.1 The TSF shall run a suite of self-tests **during each start-up of the TSF** to demonstrate the correct operation of **the** SDW.

FPT_TST.1.2 The TSF shall provide authorized users with the capability to verify the integrity of **SDW data**.

FPT_TST.1.3 The TSF shall provide authorized users with the capability to verify the integrity of **stored SDW executable code**.

5.4 Security Assurance Requirements

The assurance requirements are EAL4+ with ASE_TSS.2 and have been summarized in Table 1:

[9] Added a refinement to show exactly when the deallocation **must** happen (it may happen more often), and rewritten to make it more readable.

[10] Requirement was reworded to make it easier to read.

Table 1 Assurance requirements overview

Assurance Class	Assurance Components	
	Identifier	**Name**
ADV: Development	ADV_ARC.1	Security architecture description
	ADV_FSP.4	Complete functional specification
	ADV_TDS.3	Basic modular design
	ADV_IMP.1	Implementation representation of the TSF
AGD: Guidance documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
ALC: Life-cycle support	ALC_CMC.4	Production support, acceptance procedures and automation
	ALC_CMS.4	Problem tracking CM coverage
	ALC_DEL.1	Delivery procedures
	ALC_DVS.1	Identification of security measures
	ALC_TAT.1	Well-defined development tools
	ALC_LCD.1	Developer defined life-cycle model
ASE: Security Target evaluation	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.2	Security objectives
	ASE_REQ.2	Derived security requirements
	ASE_SPD.1	Security problem definition
	ASE_TSS.2	TOE summary specification with architectural design summary
ATE: Tests	ATE_COV.2	Analysis of coverage
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing - sample
	ATE_DPT.1	Testing: basic design
AVA: Vulnerability assessment	AVA_VAN.3	Focused vulnerability analysis

5.5 Security Assurance Requirements Rationale

The Security Assurance Requirements for this Security Target are EAL4+ with ASE_TSS.2. The TOE is part of a mission support system for air combat training that will not have an immediate impact on operational readiness or mission effectiveness as a system. The protected information is important to deployed and contingent forces and can affect mission effectiveness if it were compromised. This means that there must be a moderate to high level of assurance in the correct operation of the TOE. Paragraph 112 of CC Part 3 translates this to EAL 4.

6 TOE Summary Specification

This section describes the security functions of the TOE and the assurance measures taken to ensure correct implementation, and maps them to the security requirements.

6.1 TOE Security Functions

This section presents the IT security functions (SFs) and a mapping of security functions to security functional requirements. The TOE performs the following security functions:

- Security Filtering and Verification
- Built-In Test (BIT)
- Fail-Safe
- Residual Information Protection
- Tamper-evidence

6.1.1 Security Filtering and Verification

6.1.1.1 Message Filtering

The SDW filters all UDP messages received from the IMM MK5 for transmission to the RF DL Transmitter. The Firmware Trusted Guard (FTG) of the SDW performs the filtering function on the messages received from the IMM MK5, prior to passing the message to the Hardware Trusted Guard (HTG) of the SDW for verification.

The filtering function uses certain message attributes to determine whether the message should be passed through, masked, or discarded.

The TOE performs the following filtering logic:

- Authorized messages are passed through unmodified;
- Unauthorized messages are overwritten;
- Partially authorized messages have unauthorized parameters masked out.

There are two types of messages generated by the IMM MK5: Inoffensive Messages and Offensive Messages. Only Offensive messages may be classified. All others are explicitly authorized. With Offensive messages, certain message attributes determine the message classification according to a customized embedded security list, the CAT table, adapted to the security requirements of the Customer.

6.1.1.2 Filter Verification

The SDW HTG includes its own CAT table in parallel to the one used by the firmware. The hardware inspects all messages to be transmitted, and determines that there are no classified segments or unmasked classified parameters.

Detection of classified segments or unmasked classified parameters causes the hardware to enter fail-safe mode and to abort all further Pass-Through operations.

6.1.1.3 SFR Mapping

The following SFRs are satisfied by the Security Filtering and Verification SFs:

- **FDP_IFC.1 and FDP_IFF.1** – the Filter Policy is enforced on all objects that are generated by the Software on the IMM MK5 for the Transmitter or received by the Receiver for the IMM MK5 (the latter information flow is explicitly authorized).

  The SFV SF enforces the removal of all data that is not authorized data by performing message filtering in the SDW FTG and filter verification in the SDW HTG.

6.1.2 Built-In Test (BIT)

6.1.2.1 Self-Test

Both FTG and HTG of the SDW (TOE) perform self-tests during system start-up to verify code integrity.

The firmware and hardware filters do not allow any Data Objects through before the BIT was successfully completed (output buffer of each filter is physically blocked until end of start-up BIT).

6.1.2.2 SFR Mapping

The following SFRs are satisfied by the BIT SFs:

- **FPT_TST.1 –** the TOE tests itself on startup to verify the integrity of executable code and TSF data by activating security-critical functionality and processing paths and verifying expected results. Critical functionality, including memory integrity and internal security environment communications channels are tested before the SDW will initiate normal operations.

6.1.3 Fail-Safe

6.1.3.1 Fail-Safe

If a self-test error condition is identified, the TOE reverts to a blocked mode (fail safe), in which it will not forward any messages for transmission. In this mode, messages passed to the SDW from the Software in IMM MK5 are discarded with no feedback to the sender.

Security filtering is performed by the SDW firmware before a message is passed on to the SDW hardware for transmission. If a classified message or a non-masked classified byte is detected by the SDW hardware, Pass-Through will be disabled and Security Error flag will be set in a BIT register. The SDW hardware will also enter fail-safe mode if it receives a message with communication error from the IMM MK5.

6.1.3.2 Recovery to a Secure State

After a fail-safe event, data transmission is re-enabled only in the following cases:

- SDW power-off/power-on sequence.
- The software on IMM MK5 resets the SDW.

In both cases, buffers are flushed and the unauthorized Data Object that triggered fail-safe mode is not transmitted.

6.1.3.3 SFR Mapping

The following SFRs are satisfied by the fail-safe SFs:

- **FPT_FLS.1** – for any single SDW internal failure, the TOE preserves a secure state by entering the S.FAIL state in which all data transmission from the IMM MK5 to the RF DL Transmitter is blocked by the still-operational internal subsystem. The implementation of the SDW functionality using redundant independent mechanisms ensures that a single failure will not impact SDW functionality. Power-down invokes the FDP_RIP.1 residual information protection, ensuring that no classified data remains in the TOE.

6.1.3.4 State Model

The SDW contains six (6) possible security-relevant states, identified in Table 2. This Table also depicts the association and applicability of each SFR to the different states of the TOE. Figure 4 and Table 3 below describe the possible state transitions.

Table 2 Security States of the TOE

#	Designation	Title: Description	Ref to SFR	Comments
1	S.NO_PWR	TOE Unpowered: No electrical power is applied to the TOE; TOE is non-operational. TOE installed in the AACMI Pod.	FDP_RIP.1 FMT_MSA.3	No residual classified Data Objects are retained in the TOE when unpowered
2	S.INITIAL	Initialization: Power-up BIT The TOE does not allow any RF network data link objects to be transmitted until the TOE has acknowledged full functionality. TOE installed in the AACMI Pod.	FMT_MSA.3 FPT_FLS.1 FPT_TST.1	During power-up, both at initial start-up or recovery from an interruption in TOE service due to power failure, the TOE performs an initialization process that includes self-test and data integrity tests. Initial values for security attributes will be loaded as part of the initialization process of the TOE
3	S.OPERATE	Normal Operation of TSF: Continuous Security Filtering/Verification (SFV) of the Data Objects intended for transmission on the RF Data Link Network. TOE installed in the AACMI Pod.	FDP_IFC.1 FDP_IFF.1 FDP_RIP.1 FMT_MSA.1 FPT_FLS.1	The TOE performs security filtering and verification (SFV) on the Data Objects intended for transmission on the RF network data link.
4	S.FAIL	TOE Failure: The TSF preserves a secure state upon TOE malfunction. TOE installed in the AACMI Pod.	FMT_MSA.1F PT_FLS.1	Transmission of Data Objects on the RF network data link is terminated in the event of SFV or other failure.
5	S.NON-MISSION	Non-Mission State: Data storage cartridge is removed from the TOE Environment. TOE installed in the AACMI Pod.	FDP_RIP.1 FMT_MSA.3 FPT_PHP.1	The TOE and environment do not contain any residual Data Objects after data storage cartridge is removed.
6	S.NON_INST	Non-Installed State: TOE is external to the TOE's operational environment due to	FDP_PHP.1	

#	Designation	Title: Description	Ref to SFR	Comments
		maintenance and/or transportation.		

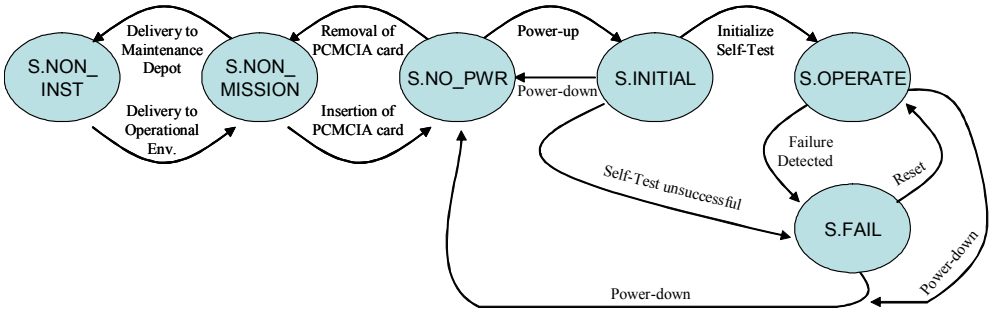

Figure 4 TOE State Transition Diagram

Table 3 TOE State Transition Table

Pre-Transition	Post-Transition	Condition	Preservation of secure state
S.NON_INST	S.NON_MISSION	Delivery from depot	No classified data
S.NON_MISSION	S.NON_INST	Delivery to depot	No classified data
S.NON_MISSION	S.NO_PWR	RSM/SIPP insertion	No classified data
S.NO_PWR	S.NON_MISSION	RSM/SIPP removal	No classified data
S.NO_PWR	S.INITIAL	Power-up	Transmission blocked
S.INITIAL	S.NO_PWR	Power-down	No classified data
S.INITIAL	S.FAIL	Self-Test failure	Transmission blocked
S.INITIAL	S.OPERATE	Self-Test successful	SDW Operational
S.OPERATE	S.FAIL	SFV failure	Transmission blocked
S.OPERATE	S.NO_PWR	Power-down	No residual information
S.FAIL	S.OPERATE	SDW reset	Unauthorized data discarded
S.FAIL	S.NO_PWR	Power-down	No residual information

6.1.4 Residual Information Protection

6.1.4.1 Non-volatile Memory Protection

Flash memory is protected from modification by untrusted subjects (specifically, software running on the IMM MK5).

The TOE is engineered such that no residual information is retained after power-off. This is ensured by protecting all non-volatile memory that can be accessed by the TOE from modification by the TOE while it is in its operational environment.

Non-volatile memory in the IMM MK5 consists of flash memories.

The IMM MK5 flash memories are inaccessible for write by external subjects, and are maintained by modules that do not receive any classified information.

The RWD component in the IMM MK5 assures that IMM MK5 non-volatile memory can be write-enabled only through a dedicated Software Upgrade Connector (SUC) that is used for uploading software application upgrades. The SUC is a master-disable input.

The SDW non-volatile memory can be write-enabled only through a dedicated Software Update Jumper (SUJ) that is used for the first upload of security software and CATS for maintenance purpose only. The SUJ is a master-disable input, which disables writing to the non-volatile memories in the SDW.

6.1.4.2 SFR Mapping

The following SFRs are satisfied by the residual information protection SFs:

- **FDP_RIP.1** – the TOE does not contain any non-volatile memory that is writable by untrusted subjects. Classified data is exclusively written into the RSM or SIPP data storage card, which is removed from EHUD immediately after the flight. TOE non-volatile type memory is write disable during the mission, therefore they obviously cannot retain any mission relevant information when the TOE is powered down.
- **FMT_MSA.1 & FMT_MSA.3** – the CAT tables are implemented in hardware and firmware, and the capability to change the CAT table outside development environment is disabled by hardware. This ensures that the requirement that nobody be able to modify the initial value of the attributes that are used for enforcement of the Filter Policy, or to perform any management operation on these attributes, is met.

6.1.5 Tamper-Evidence

6.1.5.1 Tamper-Evident Label

Tampering with the TOE is indicated by means of “Tamper-Evident Labels”, which are installed on the TOE.

6.1.5.2 SFR Mapping

The following SFRs are satisfied by the tamper-evidence SFs:

- **FPT_PHP.1** – The Tamper-Evident Labels provide detection of physical tampering with the TOE, its devices and elements.
- **FMT_MSA.3** – the management of TSF data requires access to on-board jumpers, which can only be achieved in the maintenance depot after removal of the Tamper-Evident Labels.

6.2 TOE protects itself against interference and logical tampering

The Residual Information Protection SF, which disables writing to the non-volatile memories and FPGAs, is implemented in hardware. This SF ensures the protection of the TOE against tampering and interference by any untrusted subject.

6.3 TOE protects itself against bypass

The TOE SDW location between the IMM MK5 and the Transmitter ensures that Security Filtering and Verification is always activated on all Data Objects that are delivered by the IMM MK5, before they are allowed to be sent to the Transmitter, to be transmitted on the EHUD DL.

7 Rationales

7.1 Security Objectives Rationale

Assumption/OSP/Threat	Objective
A.F.ONLY_WAY (Flight Env)	This assumption is being met by OE.F.ONLY_WAY, which is a direct translation of this assumption.
A.F.TRANSPORT_TO_O (Flight Env)	This assumption is being met by OE.F.TRANSPORT_TO_O, which includes a direct translation of this assumption.
A.F.LOCATION (Flight Env)	This assumption is being met by OE.F.TOE_LOCATION, which is a direct translation.
A.F.TRUSTED_SOFTWARE	This assumption is being met by OE.F.TRUSTED_SOFTWARE, which is a direct translation.
P.F.CORRECT_OUT (Flight Env)	This OSP is met by O.SF.DATA_FILTER which is a direct translation.
T.F.REMOTE (Flight Env)	This threat is countered by O.SF.NO_REPROGRAM showing that the TOE cannot be logically altered through the Transmitter. Furthermore, this threat is countered by OE.F.LOGICAL_PROTECTION showing that the TOE cannot be logically altered through other interfaces.
T.F.PHYSICAL (Flight Env)	This threat is countered by OE.F.TOE_LOCATION. In the Flight environment the TOE is inside a Pod attached to an operational aircraft. It is not possible to physically attack the TOE in this configuration. In addition (mentioned here only for completeness), O.SF.TAMPER_EVIDENT provides extra protection through a tamper-evident seal, making later detection of physical tampering attempts certain.
T.F.TOE_FAIL (Flight Env)	This threat is countered by O.SF.FAILSAFE, which provides protection against failure at start-up or during operation. Multiple failures may cause Data Objects containing Unauthorized Information to be transmitted.
T.F.POD_FAIL (Flight Env)	This threat is countered by OE.F.ONLY_WAY, which says that no matter what, the TOE will be between the IMM MK5 and the Transmitter. It will therefore always ensure that no Data Objects containing Unauthorized Information will reach the Transmitter. It may be the case that the Transmitter, through failure, will mangle Data Objects reaching it, thus making it appear that they contain Unauthorized Information, but this will only appear to be the case. Additionally, O.SF.NO_REPROGRAM ensures that TOE failures cannot reprogram the TOE, as the TOE cannot be logically altered.
P.O.INSPECT (O Env)	This policy is being met by OE.O.INSPECT, which is a direct translation.
A.O.CLOSED_POD and A.O.TRUSTED_ENV (O Env)	These assumptions are being met by OE.O.TRUSTED_ENV, defining the procedures how to handle equipment such as a

Assumption/OSP/Threat	Objective
	pod.
A.O.LOCATION (O Env)	This assumption is being met by OE.O.TOE_LOCATION, which is a direct translation.
A.O.NO_OFFERING (O Env)	This assumption is being met by OE.O.NO_OFFERING, which is a direct translation.
A.O.PERSONNEL (O Env)	This assumption is being met by OE.O.PERSONNEL, which is a direct translation.
A.O.TRANSPORT	This assumption is being met by OE.O.TOE_TRANSPORT, which is a direct translation.
T.O.PHYSICAL (O Env)	This threat is countered by O.SF.TAMPER_EVIDENT showing that the SDW is encased in a tamper-evident cover. In addition, the threat is countered by OE.O.TRUSTED_ENV and OE.O.TOE_TRANSPORT showing that the O-Level Staff have procedures that provide assurance that the Pod is handled and protected consistent with equipment of similar value and that unauthorized physical access to the TOE is restricted and that provide assurance that the security of the TOE is maintained when the TOE is being distributed to the Flight Environment and the I-Level Environment.
T.O.RESIDUAL (O Env)	This threat is countered by O.SF.CLEAR_WHEN_DONE by deleting all Data Objects and data derived from Data Objects upon leaving the Flight Environment.
P.I.INSPECT (I Level)	This policy is being met by OE.I.INSPECT_FROM_D, which is a direct translation.
P.I.NO_OPEN (I Level)	This policy is being met by OE.I.NO_OPEN, which is a direct translation.
P.I.SEALED (I Level)	This policy is countered by O.SF.TAMPER_EVIDENT showing that the TOE shall be encased in a tamper-evident cover.
A.I.PERSONNEL (I Level)	This assumption is being met by OE.I.PERSONNEL, which is a direct translation.
A.I.NO_OFFERING (I Level)	This assumption is being met by OE.I.NO_OFFERING, which is a direct translation.
A.I.TRANSPORT	This assumption is being met by OE.I.TOE_TRANSPORT, which is a direct translation.
A.I.TRUSTED_ENV (I Level)	This assumption is being met by OE.I.TRUSTED_ENV, which is a direct translation.
T.I.PHYSICAL (I Level)	This threat is countered by O.SF.TAMPER_EVIDENT showing that the TOE shall be encased in a tamper-evident cover.
T.I.RESIDUAL (I Level)	This threat is countered by O.SF.CLEAR_WHEN_DONE by deleting all Data Objects and data derived from Data Objects upon leaving the Flight Environment.
T.D.RESIDUAL (D Level)	This threat is countered by O.SF.CLEAR_WHEN_DONE, by deleting all Data Objects and data derived from Data Objects upon leaving the Flight Environment.

Assumption/OSP/Threat	Objective
A.F.TRUSTED_SOFTWARE	This assumption is being met by OE.F.TRUSTED_SOFTWARE, which is a direct translation.

7.2 Security Functional Requirements Rationale

Security objectives	SFRs addressing the security objectives
O.SF.DATA_FILTER	This Security Objective is implemented by FDP_IFF.1 (showing which operations are allowed), FDP_IFC.1 (showing that the policy is enforced)
O.SF.CLEAR_WHEN_DONE	This Security Objective is directly implemented by FDP_RIP.1.
O.SF.FAILSAFE	This Security Objective is directly implemented by FPT_FLS.1 for failure during operation. Furthermore FPT_TST.1 performs additional checks during each start-up of the TSF to determine that the TSF operates correctly
O.SF.TAMPER_EVIDENT	This Security Objective is directly implemented by FPT_PHP.1.
O.SF.NO_REPROGRAM	This Security Objective is realized by FMT_MSA.1 and FMT_MSA.3 that nobody can change the values of the security attributes, either during creation of Data Objects or during operation of the TOE.

7.3 Dependencies

SFR	Dependencies
FDP_IFC.1	FDP_IFF.1 Simple security attributes (included)
FDP_IFF.1	FDP_IFC.1 Subset information flow control (included)
	FMT_MSA.3 Static attribute initialization (included)
FDP_RIP.1	No dependencies
FPT_FLS.1	No dependencies
FPT_PHP.1	No dependencies
FMT_MSA.1	[FDP_ACC.1 Subset access control or FDP_IFC.1 Subset information flow control] (FDP_IFC.1 included)
	FMT_SMF.1 Specification of management functions (not satisfied[11])
	FMT_SMR.1 Security roles (not satisfied[12])
FMT_MSA.3	FMT_MSA.1 Management of security attributes
	FMT_SMR.1 Security roles (not satisfied[13])
FPT_TST.1	No dependencies
SAR	**Dependencies**
EAL 4	All dependencies within an EAL are satisfied

[11] This dependency is not satisfied, because as specified in FMT_MSA.1, the security attributes cannot be managed by anybody at all, so no management functions need to be specified.

[12] [12] This dependency is not satisfied, because as specified in FMT_MSA.1, the security attributes cannot be managed by anybody at all, so no security role needs to be specified.

[13] This dependency is not satisfied, because as specified in FMT_MSA.3, the default security attributes cannot be managed by anybody at all, so no security role needs to be specified.