Guideline, FIPS 140-2 IG G.19 Operational Equivalency Testing for HW modules

Version: 0.13

August 22, 2019

Table of Contents

A. Revision History

Version	Date	Author/Editor	Notes
0.1	2018-06-18	Yi Mao	• Updated Equivalency Type Definitions. Removed 3BSUB and 3CSUB wordings and updated the table columns accordingly. Resized the width of columns. • Updated the Important Notes on the SP may need to include a statement that declares the tested models. The update was based on the CMVP's decision that the SP doesn't need to state exactly which models were tested. This decision was conveyed by Carolyn at the ICMC round table discussion on 2018-05-11. • Replace the "Equivalency Type" column with "IG requirements" column. • Removed the SKU as an equivalency category. Per the round table discussion at the ICMC on 2018-05-11, the equivalency on SKUs should be a derived result from the equivalency on other primitive equivalency categories.
0.2	2018-07-11	Michael Williamson	• Changed eMMC 5.1 vs. UFS 2.1 NAND testing requirement to CRT, in the Memory/Storage Device section, because of a concern about what type of microcontroller is within each device. • Added a BiCS3 vs. BiCS4 example to the Memory/Storage Device section. • Added ROM type examples to the ROM category within the Memory/Storage Device section • Moved the reasoning statements within the Equivalence Testing/Justification Effort column to the Security/Crypto Relevant column. • Added Table G.8.1 – Regression Test Suite as the testing effort requirement to all Memory/Storage devices that also show CRT as their IG Requirement. • Added examples for the Tape Drive category within the Memory/Storage Device section. • Split the USB drive listing into two rows. The first row covers capacity differences. The second row covers technology differences.
0.3	2018-07-12	Michael Williamson	• Added disk formatting as a Difference Type to the Memory/Storage Devices table • Updated voting selections in the Memory/Storage Devices section based on feedback collected during the 7/12/2018 meeting.
0.4	2018-11-07	Michael Williamson	• Consolidated redundant examples in the Field Replaceable and Stationary Accessories (FR & SA) section.
0.5	2018-11-08	Michael Williamson	• Edited the Port I/O section.
0.6	2019-04-10	Michael Williamson	• Edited the Port I/O section. Added EDSFF, M.2 and U.2 to the list. Added a difference type definitions for Ethernet, FCoE, Fibre Channel, FireWire, SATA, SCSI and USB ports. Added 'Port combination difference' to the Serial, RS-232, RS-422, RS-485 line item. Added two FireWire line items to define different difference types. Removed the reference to IEEE-488 and Lightning ports. Under FireWire, moved the DMA side channel attack concern from the Security/Crypto Relevant column to the Comments or Concerns column.
0.7	2019-05-02	Michael Williamson	• Consolidated the Required Testing (RT) definition into two bulleted items. Removed the Computational Devices table from section D Equivalence Categories. Removed the Computational Devices table from section E Table G.8.1 – Regression Test Suite Selections. The data captured in each table is archived in a separate file for later reference after the CPU Equivalency work group completes its analysis. With regard to the test requirements for computational devices, added a note expressing our deference to the CPU Equivalency Work Group.
0.8	2019-06-28	Michael Williamson	• Removed references to entropy in the Equivalence Category tables. Initial comments from the labs indicated that references to entropy in the table only serve to create confusion and clutter. The IG associated with this table clearly defines entropy as out of scope. • Removed Optical Tape drive from the Memory/Storage Device table because further research showed that while at least one working system and several prototypes were developed the technology failed to achieve even moderate adoption. • Consolidated components with the same difference type and IG requirement into a single row Memory/Storage Devices table. • Removed references to entropy and physical security concerns from the Field Replaceable & Stationary Accessories and Interfaces tables. The IG table clearly states that physical security concerns are out of scope. • Consolidated components with the same difference type and IG requirement into a single row of the Interface table. • Consolidated components with the same difference type and IG requirement into a single row of the Programmable Logic Device
0.9	2019-07-09	Michael Williamson	• Added notes in the Comments/Concerns column within the PLD table in an attempt to clarify why this is an FT review as opposed to an RT review. Also add CST requirements for FT and AO reviews. • Updated the definitions for Required Testing (RT) for Equivalency Category X and Complete Regression Testing (CRT) • Changed the Security/Crypto Relevant column heading to FIPS 140 Security Relevant to emphasis the need to consider all FIPS 140 security relevant requirements. • Changed the Component column heading to Component Examples. And merges cells with common component examples. • Added Focused Testing (FT) for Equivalency Category X to the Equivalency Review Categories section • Changed the testing classification of the security relevant PLD example from RT to FT
0.10	2019-07-18	Michael Williamson	• Removed the redundant category column from each table. Removed the superfluous "(Yes/No)" from the FIPS 140 Security Relevant column. Added a note to Section E under Regression Test Suite Selections stating that Section 5 Physical Security, Section 8 EMI/EMC and Section 11 Mitigation of Other Attacks are not applicable. • Added a title page and table of contents and removed the line numbers because MS Word treats tables and figures as one line.
0.11	2019-08-14	Carolyn French	• Clarified Table labels and how they are referred to in the IG and this document. • Clarified Focused Testing language.

Version	Date	Author/Editor	Notes
0.12	2019-08-21	Michael Williamson	• Changed "would be considered security relevant" to "are security relevant" within the definition for Focused Testing (FT) for Equivalency Category X
0.13	2019-08-22		• Changed the title to reference IG G.19 instead of IG G.8

B. Equivalency Review Categories

The types of the hardware module categories within the scope of this guidance are Memory/Storage Devices, Field Replaceable and Stationary Accessories, Interfaces (I/O ports), and Programmable Logic Devices. In this document and the accompanying IG, they are referred to as **Equivalency Category X**, where **X** can be **Memory/Storage Devices, Field Replaceable and Stationary Accessories, Interfaces (I/O ports), or Programmable Logic Devices**. Section D provides details and examples for each Equivalence Category.

- **Analysis Only (AO) for Equivalency Category X**
  - Once the equivalency evidence/argument is provided and validated for Equivalency Category X, no additional testing other than proof of its physical existence is required for a module with equivalent components in Category X to the module that has been fully tested under the same validation. For example, in the **Memory/Storage Devices** Equivalency category, capacity differences for the same technology would require Analysis Only.
- **Required Testing (RT) for Equivalency Category X**
  - If a module has security relevant differences within Equivalency Category X, in comparison to a fully tested module under the same validation, the module must be tested against the TEs designated for that equivalency category specified in E. Table G.19.1 – Regression Test Suite Selections.
  - If a module has security relevant differences in multiple equivalency categories in comparison to a fully tested module under the same validation, the module must be tested against the TEs designated for each claimed equivalency category specified in E. Table G.19.1 – Regression Test Suite Selections.
- **Focused Testing (FT) for Equivalency Category X**
  - The use of some technologies may introduce Security Relevant differences that cannot be predicted by this IG. For example, Programmable Logic Devices may be used to support the Cryptographic Module in a number of different ways that are security relevant (e.g. authentication). It is up to the lab to determine what section of the standard is affected by this security relevant difference, and apply the regression tests of the corresponding section of IG G.8 Table G.8.1 – Regression Test Suite. For other sections not affected by this difference, regression testing per Table G.19.2 shall be performed.
- **Complete Regression Testing (CRT)**
  - If an equivalency justification cannot be made using the guidance provide in section D. Equivalence Categories, all modules according to their security level, must satisfy each TE listed in IG G.8 Table G.8.1 – Regression Test Suite.

C. Important Notes:

1. If different hardware configurations require object code that is derived from different source code (drivers), subset/equivalence testing is required for all different hardware configurations.
2. Vendors cannot claim physical security equivalence for modules with different cryptographic boundaries.
3. The Security Policy does not need to differentiate exactly which models were fully tested versus which were only partially tested per Equivalency IG requirements.
4. Entropy can cut across all components (e.g., the Linux kernel's built-in timer events from storage I/O is affected by hard disk vs. SSD, entropy might be harvested from cold memory harvesting, thus memory size can affect entropy). The above equivalence does not hold for Entropy, and the vendor and laboratory are responsible for appropriately analyzing entropy across all "different" devices. For example, if all devices use the same chip as a noise source, then the entropy analysis may focus only on that chip, and the dispersal of that noise/entropy throughout the product. In addition, platter count, a filtered air atmospheres vs a sealed helium atmosphere affects turbulence within an HDD. If the drive uses head tracking data as a noise source changes in turbulence affect the distribution of entropy data.
5. Computational Devices are outside the scope of this equivalency analysis. The CPU Equivalency Work Group is responsible to defining the equivalency criteria and testing requirements. Until such time that the CPU Equivalency Work Group completes its analysis the CAVS equivalency criteria defines the operational testing requirements for modules within a validation that include computational device differences.
6. The examples provided within section D. Equivalence Categories illustrate equivalency conditions that trigger an AO, RT, FT or CRT equivalency review. The examples do not serve as a definitive list for all past, present and future technology types.

D. Table G.19.2 - Equivalence Categories

	Memory/Storage Devices						
#	**Component Examples**	**Difference Type**	**Example**	**FIPS 140 Security Relevant? Justification**	**IG Requirements**	**Equivalence Testing/ Effort**	**Comments/Concerns**
1	Hard Disk Drive (HDD[i])	Capacity differences	500GB SATA hard drive vs. 1TB SATA hard drive	No Platter count, which only affects capacity, is not security relevant.	AO	Bill of Materials information is sufficient to document the difference.	
2		Technology differences	• Heat-assisted magnetic recording (HAMR) • Shingled magnetic recording (SMR) • Two dimensional magnetic recording (TDMR) • Microwave-assisted magnetic recording (MAMR) • Perpendicular magnetic recording (PMR)	No Advancements in magnetic recording technology is not security relevant.	AO	Bill of Materials information is sufficient to document the difference.	
3		Format differences	• 4K native (4Kn) • 512 native (512n) • 512 emulation (512e)	No HDD formatting is not security relevant.	AO	Bill of Materials information is sufficient to document the difference.	
4	Hard Disk Drive (HDD) or Solid State Drive (SSD)	Technology differences	256GB HDD vs. 256GB SSD	Yes HDDs spread firmware and CSP data across reserved areas in NOR and NAND flash as well as magnetic media. SSDs utilize NOR and NAND flash devices.	CRT	Test for all assurances listed within Table G.19.1 – Regression Test Suite Selections	
5		Security Architecture	TCG Enterprise, TCG Opal, TCG Ruby, ATA Security Feature Set, etc.	Yes TCG Enterprise, Opal and Ruby have different security architectures.	CRT	Test for all assurances listed within Table G.19.1 – Regression Test Suite Selections	
6	Solid State Memory Device	Technology differences	NAND vs. NOR Flash.	Yes Read and write, implementations differ across technology types.	RT	See Table G.19.1 – Regression Test Suite Selections	Zeroization at the very least must be tested for each technology type.
7	Solid State Drive (SSD)	Technology differences	• eMMC 5.1 vs. UFS 2.1 NAND flash • eMMC: parallel bus and half-duplex communication channel • UFS: serial bus and full-duplex communication channel	Yes Embedded controller and bus structure are different. Software drivers are different.	CRT	Test for all assurances listed within Table G.19.1 – Regression Test Suite Selections	Zeroization definitely must be tested for each technology type.
8		Capacity difference	4TB device vs. 12TB device	No The quantity of NAND flash within the device to store user data is not security relevant	AO	Bill of Materials information is sufficient to document the difference.	
9	Solid State Memory Device	Technology difference & size difference	BiCS3 NAND vs. BiCS4 NAND	No BiCS3 devices contain 64 layers while BiCS4 devices contain 96 layers. The increased layer count only adds capacity and therefore is not security relevant.	AO	Bill of Materials information is sufficient to document the difference.	

	Memory/Storage Devices						
#	**Component Examples**	**Difference Type**	**Example**	**FIPS 140 Security Relevant? Justification**	**IG Requirements**	**Equivalence Testing/ Effort**	**Comments/Concerns**
10	DRAM[ii]	Technology & Size differences	• DRAM vs SDRAM • Single data rate (SDR), double data rate (DDR), DDR3 or DD4 • 64GB vs 128GB	No. • Synchronous vs asynchronous operation does not affect cryptographic calculations. • Clock rate does not affect cryptographic calculations. • Capacity does not affect cryptographic calculations.	AO	Bill of Materials information is sufficient to document the difference (no need for physical access to device)	
11	MRAM[iii]	Technology & Size differences	Conventional vs Spin-transfer Torque (STT)	No. Memory cell technology difference does not affect cryptographic calculations.	AO	Bill of Materials (no need for physical access to device)	
12	NAND[iv] Flash	Technology & Size differences	• SLC v. MLC vs TLC NAND • BiCS3 vs. BiCS4 NAND	Need to assure that zeroization or other security services complete.	RT	See Table G.19.1 – Regression Test Suite Selections	If a justification is found to support an assertion that NAND memory cell technological differences affect cryptographic calculations the testing requirements should be upgraded from RT to CRT.
13		Capacity differences	8GB vs 64GB	No. Capacity does not affect cryptographic calculations.	AO	Bill of Materials (no need for physical access to device)	
14	NOR[v] Flash	Technology & Size differences	Serial vs. Parallel Interface 256Mb vs 1GB	No. • Interface type does not affect cryptographic calculations. • Capacity does not affect cryptographic calculations.	AO	Bill of Materials (no need for physical access to device)	
15	Optical Disk Drive[vi]	Technology & Size differences	CD, DVD, Blu-ray, etc.	No. Technology and capacity does not affect cryptographic calculations.	AO	Bill of Materials (no need for physical access to device)	
16	ROM[vii]	Technology differences	Mask ROM vs. EPROM vs. PROM vs. EEPROM, etc.	Yes if, • any security function accesses the ROM • any function executed from ROM memory affects a security function	CRT	Test for all assurances listed within Table G.19.1 – Regression Test Suite Selections	Need to assure that the contents of the Masked ROM and any EPROM type are identical
17		Image difference	Non-identical bit maps	Yes if, • any security function accesses the ROM • any function executed from ROM memory affects a security function	CRT	Test for all assurances listed within Table G.19.1 – Regression Test Suite Selections	
18		Size difference or bus width	• 4Mb vs 2Mb • 8-bit bus vs 16-bit bus	Yes if, • any security function accesses the ROM • any function executed from ROM memory affects a security function	CRT	Test for all assurances listed within Table G.19.1 – Regression Test Suite Selections	

	Memory/Storage Devices						
#	**Component Examples**	**Difference Type**	**Example**	**FIPS 140 Security Relevant? Justification**	**IG Requirements**	**Equivalence Testing/ Effort**	**Comments/Concerns**
19		Technology difference, image difference, capacity difference or bus width difference	• Mask ROM vs. EPROM or PROM vs. EEPROM • Non-identical bit maps • 4Mb vs 2 Mb8-bit bus vs 16-bit bus	No if, • no security functions are directly or indirectly affected by the ROM code.	AO	Bill of Materials (no need for physical access to device)	The vendor must provide evidence that proves the lack of linkage between the ROM device and FIPS 140-2 security functions.
20	Magnetic Tape[viii] Drive	Format, Technology & Size differences	• Linear, linear serpentine and helical recording methods • 100GB vs 6TB	No. Technology and capacity does not affect cryptographic calculations.	AO	Bill of Materials (no need for physical access to device)	
21	USB Flash Drive	Size differences	4TB device vs. 12TB device	No. Capacity does not affect cryptographic calculations.	AO	Bill of Materials (no need for physical access to device)	
22		Technology difference	Internal microcontroller based on a different CPU core.	Yes Different CPU cores affect cryptographic calculations.	CRT	Test for all assurances listed within Table G.19.1 – Regression Test Suite Selections.	

	Field Replaceable and Stationary Accessories (FR & SA).						
#	**Component Examples**	**Difference Type**	**Example**	**FIPS 140 Security Relevant? Justification**	**IG Requirements**	**Equivalence Testing/Justification Effort**	**Comments/Concerns**
1	Fans[ix]	Fans (size/number/positioning)	1U vs. 2U sized fans. One vs multiple fans	No	AO	Bill of Materials (no need for physical access to device)	
2	Power Supply[x]	• AC vs. DC power supply • External adapter vs internal adapter.	Power and power supplies are easily identifiable	Yes We need to assure that the module powers up.	RT	Bill of Materials and demonstrate that the module powers up and completes the power-up self-test	
3		Different number of power supplies	Single vs. multiple power supplies	Yes We need to assure that the module powers up.	RT	Bill of Materials and demonstrate that the module powers up and completes the power-up self-test	

	Interface (I/O Ports)						
#	**Component Examples**	**Difference Type**	**Example**	**FIPS 140 Security Relevant? Justification**	**IG Requirements**	**Equivalence Testing/Justification Effort**	**Comments/Concerns**
1	Port Card	Fewer or more ports of the same type	The 8-port variant of a 16-port assembly that uses the same PCB layout and surface mount devices, but without the extra riser card or with some depopulated circuits	No	AO	Bill of Materials (no need for physical access to device)	
2		Similar interface technology but the same firmware drivers	A 10/100 Ethernet port card versus a 1GbE port card that utilize the same firmware drivers.	No if source code analysis assures that, the drivers do not differ with port type.	AO	Bill of Materials (no need for physical access to device)	

	Interface (I/O Ports)						
#	**Component Examples**	**Difference Type**	**Example**	**FIPS 140 Security Relevant? Justification**	**IG Requirements**	**Equivalence Testing/Justification Effort**	**Comments/Concerns**
3		Similar interface technology but different firmware drivers	A 10/100 Ethernet port card versus a 1GbE port card that utilize different firmware drivers.	Yes if the source code analysis shows that, the firmware driver differences introduce vulnerabilities.	RT	See Table G.19.1 – Regression Test Suite Selections	
4		Different interface technology	Fiber channel vs. Ethernet	Yes	RT	See Table G.19.1 – Regression Test Suite Selections	Subset testing on equivalent products to assure the exercising of all driver code.
5	Line Card	Different number of line-card slots that support the same non-crypto/non-security relevant technology.	• Chassis or pizza box type product family that include various line card/blade slots. • For example, the Brocade 6510 (48-port) and 6520 (96-port) fit this situation.	No	AO	Bill of Materials (no need for physical access to device)	Test on one variant of the multi-slot device and apply equivalency on other variants of chassis with different number of slots.
6		Different combination of line-cards that include different security relevant technology.	• Combination of line-cards or blades that incorporate cryptography or other security relevant technology. • For example, key managers, encryption line cards, HSMs etc.	Yes	RT	See Table G.19.1 – Regression Test Suite Selections	Test on one combination of all possible line-cards/blades that incorporate crypto/security relevant technology and apply equivalency on any combination of the tested line-cards/blades.
7	DVI[xi]	Different port count	Single port vs dual port computers.	No Physical interface/layer has no security relevance	AO	Bill of Materials and/or schematics. No need for physical access to all devices. Test on one variant and apply equivalency to other variants.	High-bandwidth Digital Content Protection (HDCP) is a form of digital copy protection that could introduce encryption
8	Port Types • EDSFF[xii] • eSATA[xiii] • Ethernet[xiv] • FCoE[xv] • Fibre Channel[xvi] • FireWire (IEEE 1394) • Gigabit Ethernet[xvii] • InfiniBand[xviii] • M.2[xix] • U.2[xx] • SATA[xxi] • SCSI[xxii] • Serial, RS-232, RS-422, RS-485[xxiii]	Data rate difference Port count difference	Multi-port vs single port modules 10 GB vs 100 GB port blades	No Physical interface/layer has no security relevance	AO	Bill of Materials and/or schematics. No need for physical access to all devices	Test on one variant and apply equivalency to other variants.
9	Fiber optic[xxiv]	Single mode vs multi-mode and data rate or port count differences		No Physical interface/layer has no security relevance	AO	Bill of Materials and/or schematics. No need for physical access to all devices. Test on one variant and apply equivalency to other variants.	Test on one variant and apply equivalency to other variants.

	Interface (I/O Ports)						
#	**Component Examples**	**Difference Type**	**Example**	**FIPS 140 Security Relevant? Justification**	**IG Requirements**	**Equivalence Testing/Justification Effort**	**Comments/Concerns**
10	FireWire[xxv] (IEEE 1394)	Port absence or inclusion		Yes if some variants include and FireWire port and some do not.	RT	See Table G.19.1 – Regression Test Suite Selections	Susceptible to DMA side channel attack. Could lead to malicious external components dumping the module's memory to find CSPs.
11	FireWire (IEEE 1394)	Connector configuration differences. For example, 4-pin/6-pin FireWire 400, 9-pin FireWire 800 and Ethernet type 1394a connectors		No Link/physical layer differences are not security relevant	AO	Bill of Materials and/or schematics. No need for physical access to all devices.	Test on one variant and apply equivalency to other variants
12	USB[xxvi]	Data rate differences Connector construction differences. For example, standard type A, B or C, mini type A or B, and micro types A, B and AB.		Physical interface/layer - no security	AO	Bill of Materials and/or schematics. No need for physical access to all devices.	Test on one variant and apply equivalency to other variants

	Programmable Logic Device						
#	**Component Examples**	**Difference Type**	**Example**	**FIPS 140 Security Relevant? Justification**	**IG Requirements**	**Equivalence Testing/Justification Effort**	**Comments/Concerns**
1	CPLD[xxvii] FPGA[xxviii] PAL[xxix] GAL[xxx]	Soft IP core[xxxi] or Hard IP core[xxxii] differences	Programming code modification For example, Verilog or VHDL.	Yes, if the code differences affect one or more FIPS 140 security sections.	RT & FT	Subsection of IG G.8 Table G.8.1 – Regression Test Suite for affected FIPS 140-2 section (e.g. Section 3), plus Table G.19.1 – Regression Test Suite Selections for the remainder of the sections	FPGAs that incorporate a CPU, PLDs that mediate interface access and enforce logical disconnection requirements, PLDs that govern the module's FSM or initiate the tamper responses are examples of PLDs, which implement FIPS 140-2 security relevant items. The CST laboratory must provide a summary of the changes and rationale for mapping the code changes to FIPS 140 security sections 1 to 9.
2		Soft IP core or Hard IP core differences	Programming code modification For example, Verilog or VHDL.	No, if the code differences do not affect FIPS 140 security relevant items.	AO	Bill of Materials. Consider IP core code diff review. No need for physical access to device	The BOM should list CPLD version data. The CST laboratory must provide a summary of the changes and rationale of why the differences do not affect FIPS 140 security relevant items. Reviewing externally developed IP core is impractical.
3		Gate and Macrocell count difference	Xilinx XC2C32A: 750 Gates & 32 macrocells Xilinx XC2C256: 6000 Gates & 256 macrocells	No. Like memory devices, capacity is not security relevant.	AO	Bill of Materials. Product Datasheet	The BOM should list the manufacture's part number

E. Table G.19.1 – Regression Test Suite Selections

(NOTE: This table is for background information purposes only during review. The Normative table (Table G.19.1) is in IG G.19, and will be the only one published, i.e. the one below will be removed)

Section 5 Physical Security, Section 8 EMI/EMC and Section 11 Mitigation of Other Attacks are not applicable.

1. Memory/Storage Devices

	Memory/Storage Devices	
TEs	**TE Summary**	**TE Required**
	Section 1 Cryptographic Module Specification	
TE.01.03.02	Invoke approved mode of operation. Obtain the Approved mode of operation indicator.	Y
	Section 2 Cryptographic Module Ports and Interfaces	
TE.02.06.02	Enter error state to observe that data output is inhibited	Y
TE.02.06.04	Perform self-tests and observe that data output is inhibited	Y
TE.02.13.03	Verify the output data path is logically or physically disconnected from key generation/entry/zeroization.	N
TE.02.14.02	Verify two independent internal actions needed to output keys/CSPs in plaintext.	N
TE.02.16.02 (Levels 3 and 4)	Verify physical port(s) used for the input/output of plaintext keys/CSPs are physically separated from other ports.	N
TE.02.17.02 (Levels 3 and 4)	Verify logical interfaces used for the input/output of plaintext keys/CSPs are logically separated from other interfaces using a trusted path.	N
	Section 3 Role, Services, and Authentication	
TE.03.02.02	For modules supporting concurrent operators, verify that the module maintains the separation of the separation of the roles assumed by each operator and the corresponding services.	N
TE.03.02.03	Verify restrictions on concurrent operators, if the module support any.	N
TE.03.12.03	Verify two independent internal actions needed to invoke a bypass capability, if the module supports it.	N
TE.03.13.02	Verify the Show Status indicator of the bypass state.	N
TE.03.14.02	Verify services assigned to each role.	N
TE.03.15.02	Verify services that do not assume an authorized role.	N
TE.03.17.02 (Level 2)	Observe the denial of access to each role upon the failure of authentication.	N
TE.03.18.02 (Level 2)	Observe the change to an authorized role allows access to services, but not the change to an unauthorized role.	N
TE.03.19.02 (Levels 3 and 4)	Observe the denial of module access upon the failure of identity-based authentication.	N
TE.03.19.03 (Levels 3 and 4)	Observe the denial of services assigned to the roles that the authenticated individual is not authorized to assume.	N
TE.03.21.02	Observe the need of re-authentication after power-recycling.	N
TE.03.22.02 (Levels, 2, 3 and 4)	Observe the authentication data is protected against unauthorized disclosure, modification, and substitution.	Y
TE.03.23.02	Observe the failure to access the module before initialization, if the access is controlled.	N
	Section 4 Finite State Model	
TE.04.03.01	Observe the recovery from error states.	N
TE.04.05.08	Exercise the module to enter each of its major states.	Y
	Section 6 Operational Environment	
TE.06.05.01 (Level 1 only)	Attempt to access keys/CSPs while the crypto functions are executing.	N
TE.06.06.01 (Level 1 only)	Attempt to execute another process and observer no interruption to the module execution.	N
TE.06.07.01	Try to perform unauthorized accesses/modifications to software and firmware source and executable code.	N
TE.06.08.02	Observe the failure of the integrity check upon the corruption of the crypto software and firmware components.	Y
TE.06.11.02 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role with privileges to execute the stored crypto software and firmware components.	N
TE.06.11.03 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role without privileges to execute the stored crypto software and firmware components.	N
TE.06.12.02 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role with privileges to modify the stored crypto software and firmware components and their keys/CSPs.	N

	Memory/Storage Devices	
TEs	**TE Summary**	**TE Required**
TE.06.12.03 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role without privileges to modify the stored crypto software and firmware components and their keys/CSPs.	N
TE.06.13.02 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role with privileges to read keys/CSPs stored within crypto boundary of stored crypto software and firmware components.	N
TE.06.13.03 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role without privileges to read keys/CSPs stored within crypto boundary of stored crypto software and firmware components.	N
TE.06.14.02 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role with privileges to enter keys/CSPs.	N
TE.06.14.03 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role without privileges to enter keys/CSPs.	N
TE.06.15.02 (Levels 2, 3 and 4)	Attempt to modify executing crypto processes.	N
TE.06.16.02 (Levels 2, 3 and 4)	Attempt to read crypto software stored within the crypto boundary.	N
TE.06.17.02 (Levels 2, 3 and 4)	Verify audit records for modifications, accesses, deletions, and additions of keys/CSPs.	N
TE.06.22.02 (Levels 3 and 4)	Perform the use of the trusted mechanism to communicate all keys/CSPs.	N
TE.06.22.03 (Levels 3 and 4)	Attempt to enter or output the information via an untrusted mechanism.	N
TE.06.24.02 (Levels 3 and 4)	Invoke the trusted path via TSF.	N
TE.06.24.03 (Levels 3 and 4)	Attempt to invoke the trusted path via non-TSF.	N
TE.06.25.02 (Levels 3 and 4)	Observe the audit records for trusted path.	N
	Section 7 Cryptographic Key Management	
TE.07.01.02	Attempt to access/modify keys/CSPs by circumventing the documented protection mechanisms.	Y
TE.07.02.02	Attempt to modify/substitute public keys by circumventing the documented protection mechanisms.	Y
TE.07.15.02	Verify that no intermediate key generation values are output from the module during the key generation process.	N
TE.07.15.03	Observe the output interface and verify no plaintext intermediate key generation values.	N
TE.07.15.04	Verify that upon completion, the output of intermediate key generation values is output either 1) in encrypted form or 2) under split knowledge procedures.	N
TE.07.25.02	Verify that each key is associated with the correct entity by failing key entry using a different entity from the one under which the key was output.	N
TE.07.27.02	Verify no plaintext key display upon manual key entry.	N
TE.07.28.02	Verify manual key entry and output methods are documented correctly.	N
TE.07.29.02	Verify automated key entry/output are in encrypted form.	N
TE.07.31.04 (Levels 3 and 4)	Verify that key output is either (1) in encrypted form or (2) using split knowledge procedures, if manual methods are used to establish keys.	N
TE.07.39.02	Attempt to perform crypto functions and verify the failure if the association of key and entity is altered.	N
TE.07.41.02	Verify the non-existence of keys/CSPs after key zeroization.	Y
	Section 9 Self Tests	
TE.09.04.03	Run self-tests and cause the module to enter every error state.	N
TE.09.05.03	Verify that the crypto functions are inhibited while the module is in an error state.	N
TE.09.09.02	Verify that the module performs the POST without requiring any operator intervention.	Y
TE.09.10.02	Verify that the results of POST is output via the “status output” interface.	N
TE.09.12.02	Verify the on-demand POST is as documented.	N
TE.09.22.07	Observe the failure of POST by modifying the software/firmware components to fail integrity check.	Y
TE.09.35.05	Observe the failure of SW/FW load test by modifying the SW/FW components to fail authentication check.	N
TE.09.40.03	Verify that the manual key entry tests using EDCs is as documented.	N
TE.09.40.04	Verify that the manual key entry tests using duplicate key entries is as documented.	N
TE.09.45.03	Switch the module from the exclusive bypass service to the exclusive crypto service and verify that plaintext info is not output.	N
TE.09.46.03	Verify the correct operation of the bypass test.	N
	Section 10 Design Assurance	
TE.10.03.02	Perform the procedures for the secure installation, initialization, and startup of the crypto module and verify their correctness.	Y

2. Field Replaceable and Stationary Accessories

TEs	Field Replaceable and Stationary Accessories (FR & SA) TE Summary	TE Required
	Section 1 Cryptographic Module Specification	
TE.01.03.02	Invoke approved mode of operation. Obtain the Approved mode of operation indicator	Y
	Section 2 Cryptographic Module Ports and Interfaces	
TE.02.06.02	Enter error state to observe output.	N
TE.02.06.04	Perform self-tests and observe output.	Y
TE.02.13.03	Verify the output data path is logically or physically disconnected from key generation/entry/zeroization.	N
TE.02.14.02	Verify two independent internal actions needed to output keys/CSPs in plaintext.	N
TE.02.16.02 (Levels 3 and 4)	Verify physical port(s) used for the input/output of plaintext keys/CSPs are physically separated from other ports.	N
TE.02.17.02 (Levels 3 and 4)	Verify logical interfaces used for the input/output of plaintext keys/CSPs are logically separated from other interfaces using a trusted path.	N
	Section 3 Role, Services, and Authentication	
TE.03.02.02	For modules supporting concurrent operators, verify that the module maintains the separation of the separation of the roles assumed by each operator and the corresponding services.	N
TE.03.02.03	Verify restrictions on concurrent operators, if the module support any.	N
TE.03.12.03	Verify two independent internal actions needed to invoke a bypass capability, if the module supports it.	N
TE.03.13.02	Verify the Show Status indicator of the bypass state.	N
TE.03.14.02	Verify services assigned to each role.	N
TE.03.15.02	Verify services that do not assume an authorized role.	N
TE.03.17.02 (Level 2)	Observe the denial of access to each role upon the failure of authentication.	N
TE.03.18.02 (Level 2)	Observe the change to an authorized role allows access to services, but not the change to an unauthorized role.	N
TE.03.19.02 (Levels 3 and 4)	Observe the denial of module access upon the failure of identity-based authentication.	N
TE.03.19.03	Observe the denial of services assigned to the roles that the authenticated individual is not authorized to.	N
TE.03.21.02	Observe the need of re-authentication after power-recycling.	N
TE.03.22.02 (Levels, 2,3 and 4)	Observe the authentication data is protected against unauthorized disclosure, modification, and substitution.	N
TE.03.23.02	Observe the failure to access the module before initialization, if the access is controlled.	N
	Section 4 Finite State Model	
TE.04.03.01	Observe the recovery from error states.	N
TE.04.05.08	Exercise the module to enter each of its major states.	N
	Section 6 Operational Environment	
TE.06.05.01 (Level 1 only)	Attempt to access keys/CSPs while the crypto functions are executing.	N
TE.06.06.01 (Level 1 only)	Attempt to execute another process and observer no interruption to the module execution.	N
TE.06.07.01	Try to perform unauthorized accesses/modifications to software and firmware source and executable code.	N
TE.06.08.02	Observe the failure of the integrity check upon the corruption of the crypto software and firmware components.	N
TE.06.11.02 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role with privileges to execute the stored crypto software and firmware components.	N
TE.06.11.03 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role without privileges to execute the stored crypto software and firmware components.	N
TE.06.12.02 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role with privileges to modify the stored crypto software and firmware components and their keys/CSPs.	N
TE.06.12.03 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role without privileges to modify the stored crypto software and firmware components and their keys/CSPs.	N
TE.06.13.02 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role with privileges to read keys/CSPs stored within crypto boundary of stored crypto software and firmware components.	N
TE.06.13.03 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role without privileges to read keys/CSPs stored within crypto boundary of stored crypto software and firmware components.	N
TE.06.14.02 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role with privileges to enter keys/CSPs.	N

	Field Replaceable and Stationary Accessories (FR & SA)	
TEs	**TE Summary**	**TE Required**
TE.06.14.03 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role without privileges to enter keys/CSPs.	N
TE.06.15.02 (Levels 2, 3 and 4)	Attempt to modify executing crypto processes.	N
TE.06.16.02 (Levels 2, 3 and 4)	Attempt to read crypto software stored within the crypto boundary.	N
TE.06.17.02 (Levels 2, 3 and 4)	Verify audit records for modifications, accesses, deletions, and additions of keys/CSPs.	N
TE.06.22.02 (Levels 3 and 4)	Perform the use of the trusted mechanism to communicate all keys/CSPs.	N
TE.06.22.03 (Levels 3 and 4)	Attempt to enter or output the information via an untrusted mechanism.	N
TE.06.24.02 (Levels 3 and 4)	Invoke the trusted path via TSF.	N
TE.06.24.03 (Levels 3 and 4)	Attempt to invoke the trusted path via non-TSF.	N
TE.06.25.02 (Levels 3 and 4)	Observe the audit records for trusted path.	N
	Section 7 Cryptographic Key Management	
TE.07.01.02	Attempt to access/modify keys/CSPs by circumventing the documented protection mechanisms.	N
TE.07.02.02	Attempt to modify/substitute public keys by circumventing the documented protection mechanisms.	N
TE.07.15.02	Verify that no intermediate key generation values are output from the module during the key generation process.	N
TE.07.15.03	Observe the output interface and verify no plaintext intermediate key generation values.	N
TE.07.15.04	Verify that upon completion, the output of intermediate key generation values is output either 1) in encrypted form or 2) under split knowledge procedures.	N
TE.07.25.02	Verify that each key is associated with the correct entity by failing key entry using a different entity from the one under which the key was output.	N
TE.07.27.02	Verify no plaintext key display upon manual key entry.	N
TE.07.28.02	Verify manual key entry and output methods are documented correctly.	N
TE.07.29.02	Verify automated key entry/output are in encrypted form.	N
TE.07.31.04 (Levels 3 and 4)	Verify that key output is either (1) in encrypted form or (2) using split knowledge procedures, if manual methods are used to establish keys.	N
TE.07.39.02	Attempt to perform crypto functions and verify the failure if the association of key and entity is altered.	N
TE.07.41.02	Verify the non-existence of keys/CSPs after key zeroization.	N
	Section 9 Self Tests	
TE.09.04.03	Run self-tests and cause the module to enter every error state.	N
TE.09.05.03	Verify that the crypto functions are inhibited while the module is in an error state.	N
TE.09.09.02	Verify that the module performs the POST without requiring any operator intervention.	Y
TE.09.10.02	Verify that the results of POST is output via the “status output” interface.	N
TE.09.12.02	Verify the on-demand POST is as documented.	N
TE.09.22.07	Observe the failure of POST by modifying the software/firmware components to fail integrity check.	N
TE.09.35.05	Observe the failure of SW/FW load test by modifying the SW/FW components to fail authentication check.	N
TE.09.40.03	Verify that the manual key entry tests using EDCs is as documented.	N
TE.09.40.04	Verify that the manual key entry tests using duplicate key entries is as documented.	N
TE.09.45.03	Switch the module from the exclusive bypass service to the exclusive crypto service and verify that plaintext info is not output.	N
TE.09.46.03	Verify the correct operation of the bypass test.	N
	Section 9 Self Tests	
TE.10.03.02	Perform the procedures for the secure installation, initialization, and startup of the crypto module and verify their correctness.	N

3. Interfaces (I/O Ports)

	Required TEs for Interfaces (I/O Ports)	
TEs	**TE Summary**	**TE Required**
	Section 1 Cryptographic Module Specification	
TE.01.03.02	Invoke approved mode of operation. Obtain the Approved mode of operation indicator.	Y
	Section 2 Cryptographic Module Ports and Interfaces	
TE.02.06.02	Enter error state to observe output.	Y
TE.02.06.04	Perform self-tests and observe output.	Y
TE.02.13.03	Verify the output data path is logically or physically disconnected from key generation/entry/zeroization.	Y
TE.02.14.02	Verify two independent internal actions needed to output keys/CSPs in plaintext.	N
TE.02.16.02 (Levels 3 and 4)	Verify physical port(s) used for the input/output of plaintext keys/CSPs are physically separated from other ports.	Y
TE.02.17.02 (Levels 3 and 4)	Verify logical interfaces used for the input/output of plaintext keys/CSPs are logically separated from other interfaces using a trusted path.	Y
	Section 3 Role, Services, and Authentication	
TE.03.02.02	For modules supporting concurrent operators, verify that the module maintains the separation of the separation of the roles assumed by each operator and the corresponding services.	N
TE.03.02.03	Verify restrictions on concurrent operators, if the module support any.	N
TE.03.12.03	Verify two independent internal actions needed to invoke a bypass capability, if the module supports it.	N
TE.03.13.02	Verify the Show Status indicator of the bypass state.	N
TE.03.14.02	Verify services assigned to each role.	N
TE.03.15.02	Verify services that do not assume an authorized role.	N
TE.03.17.02 (Level 2)	Observe the denial of access to each role upon the failure of authentication.	N
TE.03.18.02 (Level 2)	Observe the change to an authorized role allows access to services, but not the change to an unauthorized role.	N
TE.03.19.02 (Levels 3 and 4)	Observe the denial of module access upon the failure of identity-based authentication.	N
TE.03.19.03	Observe the denial of services assigned to the roles that the authenticated individual is not authorized to.	N
TE.03.21.02	Observe the need of re-authentication after power-recycling.	N
TE.03.22.02 (Levels, 2,3 and 4)	Observe the authentication data is protected against unauthorized disclosure, modification, and substitution.	N
TE.03.23.02	Observe the failure to access the module before initialization, if the access is controlled.	N
	Section 4 Finite State Model	
TE.04.03.01	Observe the recovery from error states.	N
TE.04.05.08	Exercise the module to enter each of its major states.	Y
	Section 6 Operational Environment	
TE.06.05.01 (Level 1 only)	Attempt to access keys/CSPs while the crypto functions are executing.	N
TE.06.06.01 (Level 1 only)	Attempt to execute another process and observer no interruption to the module execution.	N
TE.06.07.01	Try to perform unauthorized accesses/modifications to software and firmware source and executable code.	N
TE.06.08.02	Observe the failure of the integrity check upon the corruption of the crypto software and firmware components.	N
TE.06.11.02 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role with privileges to execute the stored crypto software and firmware components.	N
TE.06.11.03 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role without privileges to execute the stored crypto software and firmware components.	N
TE.06.12.02 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role with privileges to modify the stored crypto software and firmware components and their keys/CSPs.	N
TE.06.12.03 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role without privileges to modify the stored crypto software and firmware components and their keys/CSPs.	N
TE.06.13.02 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role with privileges to read keys/CSPs stored within crypto boundary of stored crypto software and firmware components.	N
TE.06.13.03 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role without privileges to read keys/CSPs stored within crypto boundary of stored crypto software and firmware components.	N
TE.06.14.02 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role with privileges to enter keys/CSPs.	N

	Required TEs for Interfaces (I/O Ports)	
TEs	**TE Summary**	**TE Required**
TE.06.14.03 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role without privileges to enter keys/CSPs.	N
TE.06.15.02 (Levels 2, 3 and 4)	Attempt to modify executing crypto processes.	N
TE.06.16.02 (Levels 2, 3 and 4)	Attempt to read crypto software stored within the crypto boundary.	N
TE.06.17.02 (Levels 2, 3 and 4)	Verify audit records for modifications, accesses, deletions, and additions of keys/CSPs.	N
TE.06.22.02 (Levels 3 and 4)	Perform the use of the trusted mechanism to communicate all keys/CSPs.	N
TE.06.22.03 (Levels 3 and 4)	Attempt to enter or output the information via an untrusted mechanism.	N
TE.06.24.02 (Levels 3 and 4)	Invoke the trusted path via TSF.	N
TE.06.24.03 (Levels 3 and 4)	Attempt to invoke the trusted path via non-TSF.	N
TE.06.25.02 (Levels 3 and 4)	Observe the audit records for trusted path.	N
	Section 7 Cryptographic Key Management	
TE.07.01.02	Attempt to access/modify keys/CSPs by circumventing the documented protection mechanisms.	N
TE.07.02.02	Attempt to modify/substitute public keys by circumventing the documented protection mechanisms.	N
TE.07.15.02	Verify that no intermediate key generation values are output from the module during the key generation process.	N
TE.07.15.03	Observe the output interface and verify no plaintext intermediate key generation values.	N
TE.07.15.04	Verify that upon completion, the output of intermediate key generation values is output either 1) in encrypted form or 2) under split knowledge procedures.	N
TE.07.25.02	Verify that each key is associated with the correct entity by failing key entry using a different entity from the one under which the key was output.	N
TE.07.27.02	Verify no plaintext key display upon manual key entry.	N
TE.07.28.02	Verify manual key entry and output methods are documented correctly.	N
TE.07.29.02	Verify automated key entry/output are in encrypted form.	N
TE.07.31.04 (Levels 3 and 4)	Verify that key output is either (1) in encrypted form or (2) using split knowledge procedures, if manual methods are used to establish keys.	N
TE.07.39.02	Attempt to perform crypto functions and verify the failure if the association of key and entity is altered.	N
TE.07.41.02	Verify the non-existence of keys/CSPs after key zeroization.	Y
	Section 9 Self Tests	
TE.09.04.03	Run self-tests and cause the module to enter every error state.	N
TE.09.05.03	Verify that the crypto functions are inhibited while the module is in an error state.	N
TE.09.09.02	Verify that the module performs the POST without requiring any operator intervention.	N
TE.09.10.02	Verify that the results of POST is output via the “status output” interface.	N
TE.09.12.02	Verify the on-demand POST is as documented.	N
TE.09.22.07	Observe the failure of POST by modifying the software/firmware components to fail integrity check.	N
TE.09.35.05	Observe the failure of SW/FW load test by modifying the SW/FW components to fail authentication check.	N
TE.09.40.03	Verify that the manual key entry tests using EDCs is as documented.	N
TE.09.40.04	Verify that the manual key entry tests using duplicate key entries is as documented.	N
TE.09.45.03	Switch the module from the exclusive bypass service to the exclusive crypto service and verify that plaintext info is not output.	N
TE.09.46.03	Verify the correct operation of the bypass test.	N
	Section 10 Design Assurance	
TE.10.03.02	Perform the procedures for the secure installation, initialization, and startup of the crypto module and verify their correctness.	N

4. Programmable Logic Devices

TEs	Programmable Logic Devices TE Summary	TE Required
	Section 1 Cryptographic Module Specification	
TE.01.03.02	Invoke approved mode of operation. Obtain the Approved mode of operation indicator.	Y
	Section 2 Cryptographic Module Ports and Interfaces	
TE.02.06.02	Enter error state to observe output.	Y
TE.02.06.04	Perform self-tests and observe output.	Y
TE.02.13.03	Verify the output data path is logically or physically disconnected from key generation/entry/zeroization.	Y
TE.02.14.02	Verify two independent internal actions needed to output keys/CSPs in plaintext.	Y
TE.02.16.02 (Levels 3 and 4)	Verify physical port(s) used for the input/output of plaintext keys/CSPs are physically separated from other ports.	N
TE.02.17.02 (Levels 3 and 4)	Verify logical interfaces used for the input/output of plaintext keys/CSPs are logically separated from other interfaces using a trusted path.	N
	Section 3 Role, Services, and Authentication	
TE.03.02.02	For modules supporting concurrent operators, verify that the module maintains the separation of the separation of the roles assumed by each operator and the corresponding services.	N
TE.03.02.03	Verify restrictions on concurrent operators, if the module support any.	N
TE.03.12.03	Verify two independent internal actions needed to invoke a bypass capability, if the module supports it.	Y
TE.03.13.02	Verify the Show Status indicator of the bypass state.	N
TE.03.14.02	Verify services assigned to each role.	Y
TE.03.15.02	Verify services that do not assume an authorized role.	Y
TE.03.17.02 (Level 2)	Observe the denial of access to each role upon the failure of authentication.	N
TE.03.18.02 (Level 2)	Observe the change to an authorized role allows access to services, but not the change to an unauthorized role.	N
TE.03.19.02 (Levels 3 and 4)	Observe the denial of module access upon the failure of identity-based authentication.	N
TE.03.19.03	Observe the denial of services assigned to the roles that the authenticated individual is not authorized to.	N
TE.03.21.02	Observe the need of re-authentication after power-recycling.	N
TE.03.22.02 (Levels, 2,3 and 4)	Observe the authentication data is protected against unauthorized disclosure, modification, and substitution.	N
TE.03.23.02	Observe the failure to access the module before initialization, if the access is controlled.	N
	Section 4 Finite State Model	
TE.04.03.01	Observe the recovery from error states.	N
TE.04.05.08	Exercise the module to enter each of its major states.	Y
	Section 6 Operational Environment	
TE.06.05.01 (Level 1 only)	Attempt to access keys/CSPs while the crypto functions are executing.	N
TE.06.06.01 (Level 1 only)	Attempt to execute another process and observer no interruption to the module execution.	N
TE.06.07.01	Try to perform unauthorized accesses/modifications to software and firmware source and executable code.	N
TE.06.08.02	Observe the failure of the integrity check upon the corruption of the crypto software and firmware components.	N
TE.06.11.02 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role with privileges to execute the stored crypto software and firmware components.	N
TE.06.11.03 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role without privileges to execute the stored crypto software and firmware components.	N
TE.06.12.02 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role with privileges to modify the stored crypto software and firmware components and their keys/CSPs.	N
TE.06.12.03 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role without privileges to modify the stored crypto software and firmware components and their keys/CSPs.	N
TE.06.13.02 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role with privileges to read keys/CSPs stored within crypto boundary of stored crypto software and firmware components.	N
TE.06.13.03 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role without privileges to read keys/CSPs stored within crypto boundary of stored crypto software and firmware components.	N

	Programmable Logic Devices	
TEs	**TE Summary**	**TE Required**
TE.06.14.02 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role with privileges to enter keys/CSPs.	N
TE.06.14.03 (Levels 2, 3 and 4)	Verify the correct configuration of OS access control mechanisms by assuming a role without privileges to enter keys/CSPs.	N
TE.06.15.02 (Levels 2, 3 and 4)	Attempt to modify executing crypto processes.	N
TE.06.16.02 (Levels 2, 3 and 4)	Attempt to read crypto software stored within the crypto boundary.	N
TE.06.17.02 (Levels 2, 3 and 4)	Verify audit records for modifications, accesses, deletions, and additions of keys/CSPs.	N
TE.06.22.02 (Levels 3 and 4)	Perform the use of the trusted mechanism to communicate all keys/CSPs.	N
TE.06.22.03 (Levels 3 and 4)	Attempt to enter or output the information via an untrusted mechanism.	N
TE.06.24.02 (Levels 3 and 4)	Invoke the trusted path via TSF.	N
TE.06.24.03 (Levels 3 and 4)	Attempt to invoke the trusted path via non-TSF.	N
TE.06.25.02 (Levels 3 and 4)	Observe the audit records for trusted path.	N
	Section 7 Cryptographic Key Management	
TE.07.01.02	Attempt to access/modify keys/CSPs by circumventing the documented protection mechanisms.	N
TE.07.02.02	Attempt to modify/substitute public keys by circumventing the documented protection mechanisms.	N
TE.07.15.02	Verify that no intermediate key generation values are output from the module during the key generation process.	N
TE.07.15.03	Observe the output interface and verify no plaintext intermediate key generation values.	N
TE.07.15.04	Verify that upon completion, the output of intermediate key generation values is output either 1) in encrypted form or 2) under split knowledge procedures.	N
TE.07.25.02	Verify that each key is associated with the correct entity by failing key entry using a different entity from the one under which the key was output.	N
TE.07.27.02	Verify no plaintext key display upon manual key entry.	N
TE.07.28.02	Verify manual key entry and output methods are documented correctly.	N
TE.07.29.02	Verify automated key entry/output are in encrypted form.	N
TE.07.31.04 (Levels 3 and 4)	Verify that key output is either (1) in encrypted form or (2) using split knowledge procedures, if manual methods are used to establish keys.	N
TE.07.39.02	Attempt to perform crypto functions and verify the failure if the association of key and entity is altered.	N
TE.07.41.02	Verify the non-existence of keys/CSPs after key zeroization.	Y
	Section 9 Self Tests	
TE.09.04.03	Run self-tests and cause the module to enter every error state.	N
TE.09.05.03	Verify that the crypto functions are inhibited while the module is in an error state.	N
TE.09.09.02	Verify that the module performs the POST without requiring any operator intervention.	Y
TE.09.10.02	Verify that the results of POST is output via the “status output” interface.	N
TE.09.12.02	Verify the on-demand POST is as documented.	N
TE.09.22.07	Observe the failure of POST by modifying the software/firmware components to fail integrity check.	N
TE.09.35.05	Observe the failure of SW/FW load test by modifying the SW/FW components to fail authentication check.	N
TE.09.40.03	Verify that the manual key entry tests using EDCs is as documented.	N
TE.09.40.04	Verify that the manual key entry tests using duplicate key entries is as documented.	N
TE.09.45.03	Switch the module from the exclusive bypass service to the exclusive crypto service and verify that plaintext info is not output.	N
TE.09.46.03	Verify the correct operation of the bypass test.	N
	Section 10 Design Assurance	
TE.10.03.02	Perform the procedures for the secure installation, initialization, and startup of the crypto module and verify their correctness.	N

[i] A data storage device that uses magnetic storage to store and retrieve digital information using one or more rigid rapidly rotating disks (platters) coated with magnetic material.

[ii] Dynamic random-access memory (DRAM) is a type of random access semiconductor memory that stores each bit of data in a separate tiny capacitor within an integrated circuit. The capacitor can either be charged or discharged; these two states are taken to represent the two values of a bit, conventionally called 0 and 1.

[iii] Magnetoresistive random-access memory (MRAM) is a non-volatile random-access memory technology. Unlike conventional RAM chip technologies, data in MRAM is not stored as electric charge or current flows, but by magnetic storage elements.

[iv] In flash memory, each memory cell resembles a standard MOSFET, except that the transistor has two gates instead of one. On top is the control gate, as in other MOS transistors, but below this there is a floating gate, which is insulated all around by an oxide layer. The floating-gate transistors in NAND flash are connected in a way that resembles a NAND gate. Several transistors are connected in series, and the bit line is pulled low only if all the word lines are pulled high.

[v] In NOR flash, each cell has one end connected directly to ground, and the other end connected directly to a bit line. This arrangement is called "NOR flash" because it acts like a NOR gate. When one of the word lines, connected to the cell's control gate is pulled high, the corresponding storage transistor acts to pull the output bit line low.

[vi] Optical storage is the storage of data on an optically readable medium. Data is recorded by making marks in a pattern that can be read back with the aid of light, usually a beam of laser light precisely focused on a spinning optical disc. Common examples include Blu-ray, DVD and CD.

[vii] Strictly, read-only memory refers to memory that is hard-wired, such as diode matrix and the later mask ROM (MROM), which cannot be changed after manufacture.

[viii] A tape drive is a data storage device that reads and writes data on a magnetic tape. Magnetic tape data storage is typically used for offline, archival data storage.

[ix] A mechanical fan is an electrically powered machine used to create a flow within a fluid, such as air. Fans consist of a rotating arrangement of vanes or blades which act on the air.

[x] A power supply is an electrical device that supplies electric power to an electrical load. The primary function of a power supply is to convert electric current from a source to the correct voltage, current, and frequency to power the load.

[xi] Digital Visual Interface (DVI) is a video display interface used to connect a video source, such as a video display controller, to a display device, such as a computer monitor. DVI's digital video transmission format is based on panelLink, a serial format developed by Silicon Image that utilizes a high-speed serial link called transition minimized differential signaling (TMDS).

[xii] The Enterprise & Data Center SSD Form Factor (EDSFF) is a storage form factor for use in the data center that is being developed by the EDSFF Working Group.

[xiii] The e in eSATA standing for external. eSATA is a variant of SATA designed for external connectivity. It uses a more robust connector, longer shielded cables, and stricter, but backward-compatible, electrical standards. The protocol and logical signaling in the link layer, transport layer and above are identical to internal SATA.

[xiv] Ethernet is a family of computer networking technologies commonly used in local area networks (LAN), metropolitan area networks (MAN) and wide area networks (WAN).

[xv] Fibre Channel over Ethernet (FCoE) is a computer network technology that encapsulates Fibre Channel frames over Ethernet networks. This allows Fibre Channel to use 10 Gigabit or higher Ethernet networks while preserving the Fibre Channel protocol.

[xvi] Fibre Channel is a high-speed optical network interface primarily used to connect computer data storage to servers.

[xvii] Gigabit Ethernet (GbE or 1 GigE) is a term describing various technologies for transmitting Ethernet frames at a rate of a gigabit per second (1,000,000,000 bits per second), as defined by the IEEE 802.3-2008.

[xviii] InfiniBand (abbreviated IB) is a computer-networking communications standard used in high-performance computing that features very high throughput and very low latency. It is used for data interconnect both among and within computers. InfiniBand is also used as either a direct or switched interconnect between servers and storage systems, as well as an interconnect between storage systems.

[xix] M.2 (aka Next Generation Form Factor (NGFF)) is a specification for internally mounted computer expansion cards and associated connectors. It replaces the mSATA standard, which uses the PCI Express Mini Card physical card layout and connectors.

[xx] **U.2** is a computer interface for connecting SSDs to a computer. It uses up to four PCI Express lanes.

[xxi] A computer bus interface that connects host bus adapters to mass storage devices such as hard disk drives, optical drives, and solid-state drives.

[xxii] Small Computer System Interface is a set of standards for physically connecting and transferring data between computers and peripheral devices. The SCSI standards define commands, protocols, electrical and optical interfaces. SCSI is most commonly used for hard disk drives and tape drives.

[xxiii] Common digital signal communication interface. For example, RS-422 provides for data transmission, using balanced, or differential, signaling, with unidirectional/non-reversible, terminated or non-terminated transmission lines, point to point, or multi-drop. In contrast to RS-485, RS-422 does not allow multiple drivers but only multiple receivers.

[xxiv] Fiber-optic communication is a method of transmitting information from one place to another by sending pulses of light through an optical fiber.

[xxv] IEEE 1394 is an interface standard for a serial bus for high-speed communications and isochronous real-time data transfer. It was developed in the late 1980s and early 1990s by Apple, which called it FireWire. The 1394 interface is also known by the brands i.LINK (Sony), and Lynx (Texas Instruments).

[xxvi] Universal Serial Bus (USB), is an industry standard that was developed to define cables, connectors and protocols for connection, communication, and power supply between personal computers and their peripheral devices. USB was designed to standardize the connection of computer peripherals, such as keyboards, pointing devices, digital cameras, printers, portable media players, disk drives and network adapters, to personal computers. It provides a communication channel and means to supply power to peripheral devices.

[xxvii] A programmable logic device with complexity between that of PALs and FPGAs, and architectural features of both.

[xxviii] A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing – hence the term "field-programmable"

[xxix] PAL devices have arrays of transistor cells arranged in a "fixed-OR, programmable-AND" plane used to implement "sum-of-products" binary logic equations for each of the outputs in terms of the inputs and either synchronous or asynchronous feedback from the outputs.

[xxx] The generic array logic device, or GAL, has the same logical properties as the PAL but can be erased and reprogrammed

[xxxi] Soft IP cores are typically offered as synthesizable RTL. Synthesizable cores are delivered in a hardware description language such as Verilog or VHSIC hardware description language (VHDL)

[xxxii] Hard cores are defined as IP cores that cannot be modified and are thus "**hard**", analogous to the etymology of hardware and software