MIS Field Descriptions – V2.8.3

This supplement to SP800-140B contains detailed descriptions of the fields of information required for the Module Information Structure (MIS). If the module doesn't have relevant information for a required field, enter "N/A". Optional fields (identified by [O]) may be left blank but the corresponding property should be included within the json.

If the value for a field is selected from a list of values, those values are listed in this document.

Laboratory Information (laboratory)

1. Lab Name
   - Lab names are verified against the list of accredited labs from NVLAP
   - Address Info is not needed as it is obtained from NVLAP

2. Lab Code
   - Two-digit number assigned to labs once they are accredited. This is set as a result of setting the Lab Name.

3. Lab internal ID [O]
   - An ID number designated by the lab. Not used by the CMVP in the module process.

4. Signature 1

5. Title 1

6. Signature 2 [O]

7. Title 2 [O]

8. Signature 3 [O]

9. Title 3 [O]

10. Tester 1

11. Tester 1 CVP Number

12. Tester 2 [O]

13. Tester 2 CVP Number [O]

14. Tech Reviewer 1

15. Tech Reviewer 1 CVP Number [O]
    - not collected yet in Web Cryptik

16. Tech Reviewer 2 [O]

17. Tech Reviewer 2 CVP Number [O]
    - not collected yet in Web Cryptik

Vendor Information (vendor)

1. Vendor Name
    - The name of the vendor (including Corp., Inc., Ltd., etc.) that developed the cryptographic module. Please include any registration marks or special characters.

2. Address 1

3. Address 2 [O]

4. Address 3 [O]

5. City

6. State/Provence

7. Postal Code

8. Country

9. Vendor Web site
    - Full URL

10. Product Link
    - A URL that may be specific to the module or products which utilize the module. Do not just duplicate the Vendor Web site URL.

11. Contact 1

12. Email 1

13. Phone 1

14. Fax 1 [O]

15. Contact 2 [O]

16. Email 2 [O]

17. Phone 2 [O]

18. Fax 2 [O]

Module Information (module)

1. Transmission Code
    - Defines the type of submission and controls the files generated during the Create Package process. Options are listed in the CMVP Management Manual. Please reference that document for descriptions on the types and uses.

2. Scenario
    - Used when the Transmission Code is sCMn (CMVP comments or returned CSTL addressed comments)
    - Description Needed

3. Explanation
    - Used when the Transmission Code is OTHR (Other)
    - Enter an explanation that describes the purpose of this submission

4. Comments Round
    - Used when the Transmission Code is sCMn (CMVP comments or returned CSTL addressed comments)
    - A number indicating which round of comments the submission is

5. CSTL TID
    - Unique four-digit number selected by the lab for each module.
    - This may include a combination of alphabetic and numeric characters.

6. CCCS TID
    - Included for backward compatibility. The value is “0000” for any new submissions.

7. Module Name(s)
    - The complete name of the cryptographic module. Do not include the version number with the name unless by vendor choice. The name of the cryptographic module must be consistent with ISO/IEC 24759:2017 AS02.11 and the name found in the Security Policy and test report. Please include any registration marks or special characters.

8. FIPS Version
    - Currently limited to and set to “FIPS 140-3”.

9. Module Count:

    The CMVP allows multiple modules to be validated on a single certificate. However, the identification of these modules in the report must be made clear throughout the report. the module count for a validation depends on the module type: Software, Hardware, Firmware, or a Hybrid as described below.

    Software

    For a software module, its binary package(s) compiled from its source code is the IUT. The same source code may result in different sets of binaries when it's compiled for the different target platforms. The module count shall be the number of distinct sets of binaries (may map to software version, but not necessarily).

Examples:

- If a software module was validated on software version 1.0, and this source code package was compiled on three operating environments of the same family (e.g., iOS 8.0 running on iPhone5, iOS 9.0 running on iPhone5, and iOS 9.1 running on iPhone5) resulting in a single binary set, the module count is “1”.
- If a software module was validated on software version 1.0, and this source code package was compiled on two operating environments (e.g., iOS 9.0 running on iPhone5 and Android 4.0 running on a Galaxy Nexus) resulting in two separate sets of binaries (each set forming the logical boundary of the module), the module count is “2”.
- If a software module was validated on software version 1.0 and software version 2.0, and these source code packages were compiled on four operating environments (e.g., iOS 9.0 running on iPhone5, iOS 9.1 running on iPhone5, Microsoft Windows Phone 8.1 running on Windows Phone 8.1, and Android 4.0 running on a Galaxy Nexus), where two of the environments are of the same family (iOS 9.0 and iOS 9.1) resulting in six separate sets of binaries (software versions 1.0 and 2.0 each map to three distinct sets of binaries), the module count is “6”. In this case, a single iOS binary maps to both iOS 9.0 and 9.1, a single Microsoft Windows Phone binary maps to Microsoft Windows Phone 8.1, and a single Android binary maps to the Android 4.0, resulting in three distinct binaries for each software version (1.0 and 2.0), for a total of 6.

Hardware

For a hardware module report, the module count can be determined by the physical boundary of the module and understanding the components that are either tested individually and have their own boundary, or the boundary encompasses multiple components which are tested collectively.

If the boundary of the module consists of one hardware component with other hardware components within it, with each having its own hardware version number listed in the certificate (such as tamper seals, service processing cards, switch fabric, core switch blades, control processor blade, power supplies, fan kits, filler panels, management modules, network modules), then the module count shall be the number of ‘base’ modules which support the components within it.

Examples:

- If a hardware module report contains a switch (Series 1500, P/N 1010) which can optionally support four additional network modules for uplink ports without cryptographic capability (P/Ns 10, 20, 30, 40), then the module count is “1” (the switch being the ‘base’ component).
- If a hardware module report contains a router with three separately tested part numbers (Series 2000, P/Ns 10, 20, 30), and each router can be configured to use service processing card A (P/N 100) or service processing card B (P/N 101), along with tamper seal TAMP1 (P/N 500), then the module count is “3” (the routers, each part number – 10, 20 and 30 - being a ‘base’ component).
- If a hardware module report contains a series of four switches and two chassis-based switches (all running either the same firmware, or firmware with non-security relevant differences), and within the boundary of each of the chassis-based switches is a common control processor blade, four different core blades, fiber channel (FC) port blades, an optional extender blade, a power-supply and a tamper seal, then the module count is “6” (the switches being the ‘base’ component: four switches and two chassis-based switches).

If the report has several hardware modules that are individually tested and independent from one another, each having their own cryptographic boundary (flash drives, hard drives, single chips, multi-chips, etc.), but have

slight hardware differences (shape, capacity storage, number, or type of ports, etc.), then each of the independent hardware pieces shall contribute to the module count.
Examples:

- If a hardware module report contains two hard drive series with five separately tested configurations [Series SSD1 (P/Ns 128, 256, 500) and SSD2 (P/Ns 1000, 2000)], each with their own cryptographic boundary, the module count is “5”.
- If a hardware module report contains three switch series with eight separately tested configurations [Series 6000 (P/Ns 100, 101, 102), 7000 (P/Ns 200, 201) and 8000 (P/Ns 300, 301, 302)], each with their own cryptographic boundary, the module count is “8”.

If the hardware module report contains multiple firmware versions tested (with non-security relevant differences) on the same hardware platform, then the module count shall reflect the number of hardware modules only, not the number of firmware versions that are running on it.

Example

- If a hardware module includes two hard drives (one being a 250GB drive and the other being a 500GB drive), and each of these drives map to four firmware versions, the module count is “2” to reflect the hardware platforms.

Firmware

For a firmware module, its binary package(s) compiled from its source code imaged onto one or more hardware platforms is the IUT. The same source code may result in different sets of binaries when it's compiled for the different target platforms. The module count shall be the number of distinct sets of binaries (may map to firmware version, but not necessarily).

Examples

- If a firmware package was validated as firmware version 1.0 with only a single binary, and this package was tested on two hardware platforms (e.g., hardware X version 1.0 and hardware Y version 2.0), the module count is “1”.
- If a report includes firmware version 1.0 and firmware version 2.0 each with their own binary, then the module count is “2”, regardless of the number of hardware 2200 platforms these packages were tested on.
- If a firmware package was validated as firmware version 1.0, and this package results in two different sets of binaries that map to two tested hardware platforms (e.g., hardware X version 1.0 and hardware Y version 2.0), the module count is “2” based on distinct firmware binaries.

Hybrid

Since hybrid modules (hybrid firmware or hybrid software) are dependent on both the software/firmware and the hardware components, the module count shall be the total 2208 number of configurations that are possible that map to a single module boundary.

Examples:

- If a hybrid firmware includes hardware version 1.0 and firmware version 3.1, the module count is “1” since there is only a single combination of these two components.

- If a hybrid firmware includes hardware versions 1.0, 1.1, and 1.2, and firmware versions 1.1 and 1.2, and each of the hardware version can map to either of the firmware versions, then the total combination is equal to “6” (3 hardware versions times 2 firmware versions)

10. Module Description
    - Brief description of the module.

11. Module Embodiment
    - Single-chip
    - Multi-chip embedded
    - Multi-chip stand alone

    See ISO/IEC 19790:2012 Section 7.7.1 for a description of each.

12. Type
    - Software
    - Hardware
    - Firmware
    - Software-hybrid
    - Firmware-hybrid

    See ISO/IEC 19790:2012 Section 7.2.2 for a description of each.

13. Operational Environment Type
    - Modifiable
    - Limited
    - Non-modifiable

    See ISO/IEC 19790:2012 Section 7.6.1 for a description of each.

14. Section Levels
    - The total level is computed as the floor of all the levels. Levels for A and B are also set as the floor of the levels 1-12.
    - If Module Type is “Software”: Section 7 is N/A otherwise Section 7 cannot be N/A
    - If Module Type is “Software”: Section 6 cannot be Level 3 or Level 4
    - See 140-3 MM Section 7.5 *Partial validations and non-applicable areas*

15. Administrative Flags
    - ITAR
    - Add Module to MIP list

16. Cert Caveat

    This entry identifies the specific stipulations that make this certificate valid. The examples below list the potential caveats. These caveat may be modified or expanded by the CMVP during the validation process.

    Mode of Operation Caveats
    - <no caveat>

The module can only be installed and operated in an approved mode of operation.

- **When operated in approved mode**
  The module can be installed or operated in either an approved or non-approved mode of operation.
- **When installed, initialized and configured as specified in Section [section number] of the Security Policy**
  The module can be installed, initialized and/or configured in order to be considered a FIPS 140-3 recognized module. Without this configuration, the module is not considered a FIPS-validated module. After this configuration, a module may run in approved mode or non-approved mode (if supported) which may require additional configuration and/or procedural guidance to invoke.
- **The <tamper evident seals> and <security devices> installed as indicated in the Security Policy**
  Installation of the referenced components required for the module to operate in an approved mode of operation.
- **When operated in approved mode and initialized to overall level 2 per Security Policy**
  The module can be initialized to operate at different overall levels. E.g., A module can be initialized to either support level 2 role-based authentication or initialized to support only level 3 identity based authentication.

Bound/Embedded Module Caveats

- **When operated in approved mode with module [module name] validated to FIPS 140-3 under Cert. #xxxx operating in approved mode**
  The module's validation is bound to another validated cryptographic module. E.g., A software cryptographic module which requires services from another validated software cryptographic module operating in the same operational environment. Application services are available from either module.
- **This module contains the embedded module [module name] validated to FIPS 140-3 under Cert. #xxxx operating in approved mode**
  If the module incorporates an embedded validated cryptographic module.
  Example 1: a software cryptographic module which is compiled with a privately linked validated software cryptographic module operating in the same operational environment. Application services are only available from the module indicated on the certificate.
  Example 2: A hardware cryptographic module which has embedded within its physical boundary a validated cryptographic module.

Key/Entropy Caveats

- **The module generates SSPs (e.g., keys) whose strengths are modified by available entropy**
  Entropy used to generate the module's SSPs is at least 112 bits while the module generates SSPs with a comparable cryptographic strength greater than the amount of available entropy. Please refer to IG 9.3.A.
- **The module generates random strings whose strengths are modified by available entropy**
  The module generates random strings that are not SSPs, and the security strength of a generated string is less than the bit length of the string due to limited entropy. Please refer to IG 9.3.A.
- **The module generates SSPs (e.g., keys) and random strings whose strengths are modified by available entropy**
  The module generates both SSPs and random strings that are not SSPs that have security strengths less than the presumed strengths of the SSPs and strings. Please refer to IG 9.3.A.
- **No assurance of the minimum strength of generated SSPs (e.g., keys)**
  The entropy seed is obtained from outside of the module's boundary with no CMVP evaluation on the entropy. Please refer to IG 9.3.A.
- **When entropy is externally loaded, no assurance of the minimum strength of generated SSPs (e.g., keys)**
  Entropy source falls under a particular condition specified in IG 9.3.A scenario 3.

- **No assurance of minimum security of SSPs (e.g., keys, bit strings) that are externally loaded, or of SSPs established with externally loaded SSPs**
  The module receives SSPs (e.g., keys or bit strings) from outside of its boundary for use within an approved algorithm (e.g., a key used for AES encryption; or a bit string used for generating the k values of DSA and ECDSA sigGen algorithms). This caveat does not apply to a seed used for an internal DRBG, or a seed used in an asymmetric key generation algorithm since that must be obtained from an approved DRBG in compliance with SP 800-133r2.
- **The output of the DRBG may not be used to generate SSPs (e.g., keys)**
  The module implements a DRBG and the underlying entropy source does not meet the requirements of IGs 9.3.A, D.J and D.K.

Other Caveats

- **When utilizing a Trusted Channel as specified in the Security Policy**
  If the use of the Trusted Channel is claimed to meet the FIPS 140-3 compliance requirements of ISO/IEC 19790:2012 Section 7.3.4. Please refer to IGs 3.4.A and 9.5.A*.*
- **The protocol(s) <TLS, SSH, …> shall not be used when operated in approved mode**
  If the module implements a KDF from NIST SP 800-135rev1 and this KDF has not been validated by the CAVP. Please refer to IG D.C.

17. PIV Cert #
    - When a module implements a validated PIV application, the application validation certificate type and number shall be included. Additional information relating to PIV versioning can be found in the MM Section 7.6, *CMVP requirements for PIV validations References*.

18. Special Instructions
    - E.g., which module submissions should be grouped, what are the dependencies, etc.