Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program

National Institute of Standards and Technology
Communications Security Establishment

Initial Release: March 28, 2003

Last Update: August 03, 2010

Table of Contents

- 06/10/10: G.2 Completion of a test report: Information that must be provided to NIST and CSEC – Updated submission and billing information requirements.

- 06/10/10: G.13 Instructions for completing a FIPS 140-2 Validation Certificate – Additional caveat examples.

- 06/10/10: 1.3 Firmware Designation – Updated platform versioning requirements if physical security is Level 2, 3 or 4.

- 06/10/10: 5.4 Level 3: Hard Coating Test Methods – Modified temperature testing limits and removed testing methods using solvents.

- 06/10/10: 7.5 Strength of Key Establishment Methods – Added reference to draft NIST SP 800-131.

- 06/10/10: A.6 CAVP Requirements for Vendor Affirmation of FIPS 186-3 Digital Signature Standard – Updated with transition end date for ECDSA.

- 04/09/10: A.6 CAVP Requirements for Vendor Affirmation of FIPS 186-3 Digital Signature Standard – Updated with transition end date.

- 04/09/10 A.7 CAVP Requirements for Vendor Affirmation of NIST SP800-38E – Updated with transition end date.

- 03/19/10: 1.9 Definition and Requirements of a Hybrid Cryptographic Module - Updated the annotation for software-hybrid and, firmware-hybrid modules.

- 03/19/10: G.13 Instructions for completing a FIPS 140-2 Validation Certificate - Added examples for software-hybrid and firmware-hybrid modules.

- 01/27/10: 7.2 Use of IEEE 802.11i Key Derivation Protocols
  Guidance updated in regard to references to NIST SP 800-56A and IG 7.10.

- 01/27/10: G.13 Instructions for completing a FIPS 140-2 Validation Certificate
  Removed PIV Middleware reference. Added XTS-AES annotation reference.

- 10/21/09: To align Implementation Guidance that is associated with underlying algorithmic standards referenced in FIPS 140-2 Annexes A, C and D, the following algorithm specific IGs have been moved to new IG Annex sections:

  Moved IG 1.5 to IG A.1, IG 1.6 to IG A.2, IG 1.10 to A.3, IG 1.11 to IG D.1, IG 1.12 t IG C.1, IG 1.13-15 to IG A..4-6, IG 7.1 to IG D.2 and IG 7.3 to IG C.2

- 10/20/09: G.1 Request for Guidance from the CMVP and CAVP – Updated contact information.

- 10/20/09: G.2 Completion of a test report: Information that must be provided to NIST and CSEC – Minor editorial changes.

- 10/20/09: G.13 Instructions for completing a FIPS 140-2 Validation Certificate – Added FIPS 186-3 and SP 800-56A annotation examples.

- 10/20/09: 1.11 CAVP Requirements for Vendor Affirmation of NIST SP 800-56A – Added reference to the annotation requirements in IG G.13.

- 10/20/09: 1.15 CAVP Requirements for Vendor Affirmation of FIPS 186-3 Digital Signature Standard – Added transition information and reference to the annotation requirements in IG G.13.

- 10/20/09: 7.1 Acceptable Key Establishment Protocols – Added transition information.
- 08/31/09: 7.1 Acceptable Key Establishment Protocols – Added references to DTLS.
- 08/04/09: G.13 Instructions for completing a FIPS 140-2 Validation Certificate – Added additional certificate annotation examples.
- 08/04/09: 1.10 Vendor Affirmation of Cryptographic Security Methods – Additional certificate annotation examples.
- 08/04/09: 1.15 CAVP Requirements for Vendor Affirmation of FIPS 186-3 Digital Signature Standard – Certificate annotation examples.
- 08/04/09: 7.1 Acceptable Key Establishment Protocols – For Key Agreement; removed the KDF specified in the SRTP protocol (IETF RFC 3711). For Key Transport; added reference to EAP-FAST and PEAP-TLS.
- 03/10/09: G.1 Request for Guidance from the CMVP – Updated NIST POC.
- 03/10/09: G.5 Maintaining validation compliance of software or firmware cryptographic modules – Updated references to firmware and hybrid modules.
- 03/10/09: G.13 Instructions for completing a FIPS 140-2 Validation Certificate – Updated examples.
- 03/10/09: 1.9 Definition and Requirements of a Hybrid Cryptographic Module – Updated to include hybrid firmware modules.
- 03/10/09: 7.1 Acceptable Key Establishment Protocols – For Key Agreement; added the KDF specified in the SRTP protocol (IETF RFC 3711) is allowed only for use as part of the SRTP key derivation protocol. For Key Transport; wrapping a key using the GDOI Group Key Management Protocol described in the IETF RFC 3547.
- 07/09/08: 1.10 Vendor Affirmation of Cryptographic Security Methods – Updated examples of certificate algorithm notation.
- 06/25/08: G.13 Instructions for completing a FIPS 140-2 Validation Certificate – Updated file naming convention syntax
- 05/22/08: G.13 Instructions for completing a FIPS 140-2 Validation Certificate – Updated reference for symmetric key wrapping annotation
- 02/07/08: 7.1 Acceptable Key Establishment Protocols – Updated AES Key Wrap URL.
- 01/24/08: G.2 Completion of a test report: Information that must be provided to NIST and CSE – Added reference to CMVP comments document.
- 01/24/08 G.8 Revalidation Requirements – Added reference to the CMVP FAQ in change scenario 1.
- 01/16/08: G.13 Instructions for completing a FIPS 140-2 Validation Certificate – Added reference for listing multiple operating systems, and reference for symmetric key wrapping annotation.
- 01/16/08: 1.8 Listing of DES Implementations – Updated to reflect the ending of the DES transition period.

- 01/16/08: 7.1 Acceptable Key Establishment Protocols
- 01/16/08: 9.4 Cryptographic Algorithm Tests for SHS Algorithms and Higher Cryptographic Algorithms Using SHS Algorithms – Added RSA KAT requirements regarding the relationship of the exponents.
- 11/08/07: G.2 Completion of a test report: Information that must be provided to NIST and CSE – Added clarification on output type of draft certificate.
- 10/18/07: Updated links
- 07/26/07: Minor editorial updates.
- 06/26/07: 7.1 Acceptable Key Establishment Protocols – Updated to reflect the publishing of NIST SP 800-56A.
- 06/26/07: G.8 Revalidation Requirements – Additional guidelines for determining <30% change for Scenario 3.
- 06/22/07: G.2 Completion of a test report: Information that must be provided to NIST and CSE - editorial changes for clarification.
- 06/22/07: G.8 Revalidation Requirements - editorial changes for clarification.
- 06/14/07: 3.1 Authorized Roles
- 03/19/07: Updated references to revision of NIST SP 800-57
- 02/26/07: 1.6 Use of Non-NIST-Recommended Asymmetric Key Sizes and Elliptic Curves
- 02/23/07: 7.4 Zeroization of Power-Up Test Keys
- 01/25/07: G.8 Revalidation Requirements
- 01/25/07: 7.5 Strength of Key Establishment Methods

Overview

This Implementation Guidance document is issued and maintained by the U.S. Government's National Institute of Standards and Technology (NIST) and the Communications Security Establishment Canada (CSEC), which serve as the validation authorities of the Cryptographic Module Validation Program (CMVP) for their respective governments. The CMVP is a program under which National Voluntary Laboratory Accreditation Program (NVLAP) accredited Cryptographic and Security Testing (CST) Laboratories test cryptographic modules for conformance to Federal Information Processing Standard Publication (FIPS) 140-2, *Security Requirements for Cryptographic Modules*. The Cryptographic Algorithm Validation Program (CAVP) addresses the testing of Approved security functions which are referenced in the annexes of FIPS 140-2..

This document is intended to provide clarifications of the CMVP, and in particular, clarifications and guidance pertaining to the *Derived Test Requirements for FIPS PUB 140-2* (DTR), which is used by CST Laboratories to test for a cryptographic module's conformance to FIPS 140-2. Guidance presented in this document is based on responses issued by NIST and CSEC to questions posed by the CST Labs, vendors, and other interested parties. *However, information in this document is subject to change by NIST and CSEC.*

Each section of this document corresponds with a requirements section of FIPS 140-2, with an additional first section containing general guidance that is not applicable to any particular requirements section. Within each section, the guidance is listed according to a subject phrase. For those subjects that may be applicable to multiple requirements areas, they are listed in the area that seems most appropriate. Under each subject there is a list, including the date of issue for that guidance, along relevant assertions, test requirements, and vendor requirements from the DTR. *(Note: For each subject, there may be additional test and vendor requirements which apply.)* Next, there is section containing a question or statement of a problem, along with a resolution and any additional comments with related information. This is the implementation guidance for the listed subject.

Below is a list of where the reader can find cryptographic modules validated to 140-1 and 140-2:

- Cryptographic Module Validation List

General Issues

G.1 Request for Guidance from the CMVP and CAVP

Applicable Levels:	*All*
Original Publishing Date:	*02/25/1997*
Effective Date:	*02/25/1997*
Last Modified Date:	*10/20/2009*
Relevant Assertions:	*General*
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Background

The Cryptographic Module Validation Program (CMVP) and the Cryptographic Algorithm Validation Program (CAVP) defines two types of questions: *Programmatic Questions* and *Test-specific Questions*. The CMVP and CAVP define two types of requests: *Informal Requests* and *Official Requests*.

Question/Problem

What is the difference between *Informal Requests* verses *Official Requests*? To whom should these questions be directed? If an official reply is requested for a question, is there a defined format for these types of requests?

Resolution

Programmatic Questions*:* These are questions pertaining to the general operation of the Cryptographic Module Validation Program or the Cryptographic Algorithm Validation Program. The CMVP and CAVP suggest reviewing the CMVP Management Manual, CMVP Frequently Asked Questions (FAQ), the CAVP Frequently Asked Questions (FAQ), CMVP Announcements and CMVP Notices posted on the CMVP and CAVP web sites first as the answer may be readily available. The information found on the CMVP web site provides the official position of the CMVP and CAVP.

Test-specific Questions*:* These are questions concerning specific test issues of the Cryptographic Module Validation Program or the Cryptographic Algorithm Validation Program. These issues may be technology related or related to areas of the standard that may appear to be open to interpretation.

General Guidance: Programmatic questions regarding the CMVP or the CAVP can be directed to either NIST or CSEC by contacting the appropriate points of contact listed below. The complete list of NIST and CSEC points of contacts **shall** be included on copy for all questions.

Vendors who are under contract with a CST laboratory for FIPS 140-2 or algorithm testing of a particular implementation(s) must contact the contracted CST laboratory for any questions concerning the test requirements and how they affect the testing of the implementation(s).

CST Laboratories must submit all *test-specific questions* in the RFG format described below. These questions must be submitted to all points of contact.

Federal agencies and departments, and vendors not under contract with a CST laboratory who have specific questions about a FIPS 140-2 test requirements or any aspect of the CMVP or CAVP should contact the appropriate NIST and CSEC points of contact listed below.

Questions can either be submitted by e-mail, telephone, and facsimile or written (if electronic document, Microsoft Word document format is preferred).

Informal Request: Informal requests are considered as *ad hoc* questions aimed at clarifying issues about the FIPS 140-2 and other aspects of the CMVP and CAVP. Replies to informal requests by the CMVP are non-binding and subject to change. It is recommended that informal requests be submitted to all points of contact. Every attempt is made to reply to informal request with accurate, consistent, clear replies on a very timely basis.

Official Request: If an official response is requested, then an official request must be submitted to the CMVP and/or CAVP written in the Request for Guidance (RFG) format described below. An official response requires internal review by both NIST and CSEC, as well as with others as necessary, and may require follow-up questions from the CMVP and/or CAVP. Therefore such requests, while time sensitive, may not be immediate.

Request for Guidance Format: Questions submitted in this format will result in an official response from the CMVP and CAVP that will state current policy or interpretations. This format provides the CMVP and CAVP a clear understanding of the question. An RFG **shall** have the following items:

1. Clear indication of whether the RFG is **PROPRIETARY** or **NON-PROPRIETARY**,
2. A descriptive title,
3. Applicable statement(s) from FIPS 140-2,
4. Applicable assertion(s) from the FIPS 140-2 DTR,
5. Applicable required test procedure(s) from the FIPS 140-2 DTR,
6. Applicable statements from FIPS 140-2 Implementation Guidance,
7. Applicable statements from algorithmic standards,
8. Background information if applicable, including any previous CMVP or CAVP official rulings or guidance,
9. A concise statement of the problem, followed by a clear and unambiguous question regarding the problem, and
10. A suggested statement of the resolution that is being sought.

All questions should be presented in a detailed and implementation-specific format, rather than an academic or hypothetical format. This information should also include a brief non-proprietary description of the implementation and the FIPS 140-2 target security level. All of this will enable a more efficient and timely resolution of FIPS 140-2 related questions by the CMVP and CAVP. The statement of resolution **shall** be stated in a manner which the CMVP and CAVP can either answer "YES" or "NO". The CMVP may optionally provide rationale if the answer is not in line with the suggested statement of resolution.

When appropriate, the CMVP and CAVP will derive general guidance from the problem and response, and add that guidance to this document. Note that general questions may still be submitted, but these questions should be identified as not being associated with a particular validation effort.

Preferably, questions should be non-proprietary, as their response will be distributed to ALL CST laboratories. Distribution may be restricted on a case-by-case basis.

NIST and CSEC Points of Contact:

- **National Institute of Standards and Technology – CMVP**

  Randall J. Easter   CMVP@nist.gov
  (301) 975-4641

- **National Institute of Standards and Technology (NIST) – CAVP**

  Sharon Keller   skeller@nist.gov
  (301) 975-2910

- **Communications Security Establishment Canada (CSEC) – CMVP**

  Ken Lu   CMVP@cse-cst.gc.ca
  (613) 991-8122

Additional Comments

G.2 Completion of a test report: Information that must be provided to NIST and CSEC

Applicable Levels:	*All*
Original Publishing Date:	*02/25/1997*
Effective Date:	*02/25/1997*
Last Modified Date:	*06/10/2010*
Relevant Assertions:	*General*
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Question/Problem

What information should be submitted to NIST and CSEC upon completion of the CST laboratory conformance testing in order for NIST and CSEC to perform a validation review? Are there any other additional requirements during report COORDINATION?

Resolution

The following test report information **shall** be provided to both NIST and CSEC by the CST laboratory upon report submission. The ZIP file and files within the ZIP file **shall** follow all programmatic naming conventions[1] and submitted to the CMVP using the specified encryption methods.

1. **Non-proprietary Security Policy** <PDF>

   a. Reference FIPS 140-2 Appendix C, FIPS 140-2 DTR Appendix C and IG Section 14 for requirements.

[1] *CMVP Convention for E-mail Submittal*

b. The non-proprietary security policy **shall** not be marked as proprietary or copyright without a statement allowing copying or distribution.

2. **CRYPTIK v7.0 (or higher) Reports**

The validation report submission must be output from the NIST provided CRYPTIK tool.

a. **Signature page / Cover Sheet** <PDF>

1. Scanned image of the CRYPTIK Signature Page / Cover Sheet report with the appropriate signatures.

b. **General Vendor / Module Information** <PDF>

1. CRYPTIK General report.
2. Reference IG G.13 for requirements.
3. If a CRYPTIK field is not large enough to contain information that will be reflected on the draft certificate, please state in the field "Provided in draft certificate". Then the complete information **shall** be manually entered on the draft certificate.

c. **Report Overview with Assessments** (5SUB submission) <PDF>

d. **Re-Validation Report with Assessments** (2SUB and 3SUB submissions) <PDF>

e. **Full Report with Assessments** <PDF>

f. **Draft Certificate** <DOC>

1. CRYPTIK Certificate report.
2. The .rtf Certificate output from CRYPTIK **shall** be renamed to a .doc file.
3. The only modifications made **shall** be those identified in IG G.13 or above in Section 2.b.2.

g. **Vendor Text File** <TXT>

1. Export the General Vendor file from the CRYPTIK FILE I/O menu.
2. Rename the _vendor.txt file per the programmatic naming convention.

h. **Definitions** **or** **References** <PDF - *optional*>

3. **Physical Test Report** <PDF – *mandatory* at FIPS 140-2 Section 4.5 Physical Security Levels 2, 3 and 4>

The laboratory's physical testing report with photos, drawing, etc. as applicable.

4. **Executive Overview with Section Summaries** < PDF – *mandatory* for 2SUB and 3SUB submissions; optional for 5SUB submission>

Provide an executive overview of the module. If a 2SUB or 3SUB annotate the validation certificate of the module the report is based on. Briefly describe how the requirements in each section are met.

Note: Billing information is no longer required as part of the CRYPTIK PDF output as the information is included within the _vendor.txt file.

The PDF files **shall** not be locked. All CRYPTIK PDF submission output **shall** be *merged* into a single PDF document in the following order: Signature Page / Cover Sheet, General Vendor / Module Information, Executive Overview with Section Summaries, Report Overview or Re-Validation Report, Full report, and

Physical Test Report.

The submission documents (Security Policy, CRYPTIK Report, _vendor.txt and Draft Certificate) **shall** be ZIP'ed into a single file, encrypted and sent to the following NIST and CSEC points of contact:

- **NIST:** CMVP@nist.gov
- **CSEC:** CMVP@cse-cst.gc.ca

Once the electronic report submission document is received by the CMVP it will be placed in the report queue in order received. Those reports marked to be listed, will appear in the weekly published Modules-In-Process listing posted on the CMVP web site. The listing and the definition of the five stages of the Modules-In-Process listing is found at: http://csrc.nist.gov/groups/STM/cmvp/inprocess.html

During the COORDINATION phase the CST laboratory will address each CMVP comment and update any applicable files as necessary in addition to providing a response and additional clarification as necessary in the CMVP comments document. The laboratory will re-submit the report in its entirety as above (i.e. full report submission) including the updated CMVP comments file.

5. **CMVP Comments** <DOC>

Additional Comments

The naming convention for the submitted ZIP file, e-mail subject line, and files within the ZIP file is provided to the CST Labs in a separate document *CMVP Convention for E-mail Submittal.* Contact Beverly Trapnell for the latest version of this document. The CRYPTIK ***File I/O and EMAIL*** function will generate the proper e-mail subject line name depending on the transaction.

An initial or preliminary review will not be performed on the submission documents to determine their completeness. The report information in the _vendor.txt file will be imported to the CMVP Tracking DataBase and billing information, if applicable, will be sent to NIST billing. The weekly Modules-In-Process listing will be generated based on this information provided.

G.3 Partial Validations and Not Applicable Areas of FIPS 140-2

Applicable Levels:	*All*
Original Publishing Date:	*02/25/1997*
Effective Date:	*02/25/1997*
Last Modified Date:	*01/21/2005*
Relevant Assertions:	*General*
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Question/Problem

Can a cryptographic module be validated only for selected areas of Section 4 of FIPS 140-2? Which areas of Section 4 of FIPS 140-2 can be marked *Not Applicable*?

Resolution

NIST and CSEC will not issue a validation certificate unless the cryptographic module meets at least the Security Level 1 requirements for each area in Section 4 of FIPS 140-2 that cannot be designated as *Not Applicable* according to the following:

- **Section 4.5**, Physical Security may be designated as *Not Applicable* if the cryptographic module is a software-only module and thus has no physical protection mechanisms;

- **Section 4.6**, Operational Environment may be designated as *Not Applicable* depending on the module implementation (e.g. if the operational environment for the cryptographic module is a limited operational environment); and

- **Section 4.11**, Mitigation of Other Attacks may be designated as *Not Applicable* if the vendor has made no claim that the cryptographic module provides such protection mechanisms.

The CST laboratory must provide in the validation test report the rationale for marking sections as *Not Applicable*.

Additional Comments

If a section is *Not Applicable*, it will be marked N/A on the module validation certificate. If Section 4.6 is N/A, depending on the module implementation, configuration information may still be required on the module validation certificate (e.g. a *firmware* module must provide the tested configuration)

G.4 Design and testing of cryptographic modules

Applicable Levels:	*All*
Original Publishing Date:	*11/12/1997*
Effective Date:	*11/12/1997*
Last Modified Date:	*04/28/2000*
Relevant Assertions:	*General*
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Question/Problem

What activities may CST laboratories perform, regarding the design and testing of cryptographic modules?

Resolution

The following information is supplemental to the guidance provided by NVLAP, and further defines the separation of the design, consulting, and testing roles of the laboratories. CMVP policy in this area is as follows:

1. A CST Laboratory *may not* perform validation testing on a module for which the laboratory has:

    a. designed any part of the module,

    b. developed original documentation for any part of the module,

    c. built, coded or implemented any part of the module, or

    d. any ownership or vested interest in the module.

2. Provided that a CST Laboratory has met the above requirements, the laboratory *may* perform validation testing on modules produced by a company when:

    a. the laboratory has no ownership in the company,

b. the laboratory has a completely separate management from the company, and

c. business between the CST Laboratory and the company is performed under contractual agreements, as done with other clients.

3. A CST Laboratory may perform consulting services to provide clarification of 140-2, the Derived Test Requirements, and other associated documents at any time during the life cycle of the module.

Additional Comments

Item 3 in the Resolution references "other associated documents". Included in this reference are:

- Documents developed by the CMVP staff for the Cryptographic Module testing program (e.g., Implementation Guidance, CMVP Policy, Handbook 150-17, *Cryptographic Module Testing*); and
- Implementation Guidance and Policy associated with 140-2, *Security Requirements for Cryptographic Modules*.

Also see IG G.9, regarding FSM and Security Policy consolidation and formatting.

G.5 Maintaining validation compliance of software or firmware cryptographic modules

Applicable Levels:	*All*
Original Publishing Date:	*11/21/1997*
Effective Date:	*11/21/1997*
Last Modified Date:	*03/10/2009*
Relevant Assertions:	*General*
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Question/Problem

For a validated software or firmware cryptographic module, how may such a module be implemented so that compliance with the validation is maintained?

Resolution

The tested/validated module version, operational environment upon which it was tested, and the originating vendor are stated on the validation certificate. The certificate serves as the benchmark for the module-compliant configuration.

This guidance addresses two separate scenarios: actions a vendor can affirm or change to maintain a module’s validation and actions a user can affirm to maintain a module’s validation.

This guidance is *not applicable* for validated modules when **Section 4.5 Physical Security** has been validated at **Levels 2** or higher. Therefore this guidance is only applicable at Level 1 for *firmware* or *hybrid* modules.

Vendor

1. A vendor may perform post-validation recompilations of a software or firmware module and affirm the modules continued validation compliance provided the following is maintained:

a) Software modules that do not require any source code modifications (e.g., changes, additions, or deletions of code) to be recompiled and ported to another operational environment must:

   i) For **Level 1 Operational Environment**, a software cryptographic module will remain compliant with the FIPS 140-2 validation when operating on any general purpose computer (GPC) provided that the GPC uses the specified single user operating system/mode specified on the validation certificate, or another compatible single user operating system, and

   ii) For **Level 2 Operational Environment**, a software cryptographic module will remain compliant with the FIPS 140-2 validation when operating on any GPC provided that the GPC incorporates the specified CC evaluated EAL2 (or equivalent) operating system/mode/operational settings or another compatible CC evaluated EAL2 (or equivalent) operating system with like mode and operational settings.

b) Firmware modules (i.e. Operational Environment is *not applicable)* that do not require any source code modifications (e.g., changes, additions, or deletions of code) to be recompiled and its identified unchanged tested operating system (i.e. same version or revision number) may be ported together from one GPC or platform to another GPC or platform while maintaining the module's validation.

c) Hybrid modules (i.e. Operational Environment may or may not be applicable depending if the controlling component is software or firmware) that do not require any of the following:

   i) software or firmware source code modifications (e.g., changes, additions, or deletions of code) to be recompiled and its identified unchanged tested operating system (i.e. same version or revision number)

   ii) hardware components utilized by the controlling software or firmware is not modified (e.g. changes, additions, or deletions)

d) may be ported together from one GPC or platform to another GPC or operating platform while maintaining the module's validation

The CMVP allows vendor porting and re-compilation of a validated software, firmware or hybrid cryptographic module from the operational environment specified on the validation certificate to an operational environment which was not included as part of the validation testing as long as the porting rules are followed. The validation status of the cryptographic module is maintained without the cryptographic module being retested in the new operational environment. However, the CMVP makes no statement as to the correct operation of the module when so ported if the specific operational environment is not listed on the validation certificate.

The vendor may provide a new security policy which would affirm and include references to the new operational environment(s), GPC(s) or platform(s).

2. Software or firmware modules that require non-security relevant source code modifications (e.g., changes, additions, or deletions of code) to be recompiled and ported to another hardware or operational environment must be reviewed by a CST laboratory and revalidated per **FIPS 140-2 IG G.8 (1)** to ensure that the module does not contain any operational environment-specific or hardware environment-specific code dependencies.

3. If the new operational environment and/or platform is requested to be updated on the validation certificate, the CST laboratory **shall** follow the requirements for non-security relevant changes in **FIPS 140-2 IG G.8 (1)** and in addition, perform the regression test suite of operational tests included in **FIPS 140-2 IG G.8** Table G.8.1 – Regression Test Suite. Underlying algorithm validations must meet requirements specified in **FIPS 140-2 IG 1.4** *Binding of Cryptographic Algorithm Validation Certificates.*

   Upon re-testing and validation, the CMVP provides the same assurance as the original operational environment(s) as to the correct operation of the module when ported to the newly listed OS(s) and/or operational environment(s) which would be added to the modules validation web entry.

The vendor must meet all applicable requirements in **FIPS 140-2 Section 4.10**.

This policy only addresses the operational environment under which a software, firmware or hybrid module executes and does not affect requirements of the other sections of FIPS 140-2. A module must meet all requirements of the level stated.

FIPS 140-2 IG 1.3 *Firmware Designation* describes the difference in terminology between a *software* and a *firmware* module.

FIPS 140-2 IG 1.9 *Hybrid Designation* describes the attributes and definition of a hybrid module.

User

A user may not modify a validated module. Any user modifications invalidate a modules validation. **Note 1**

A user may perform post-validation porting of a module and affirm the modules continued validation compliance provided the following is maintained:

1. For **Level 1 Operational Environment**, a software, firmware or hybrid cryptographic module will remain compliant with the FIPS 140-2 validation when operating on any general purpose computer (GPC) or platform provided that the GPC for the software module, or software controlling portion of the hybrid module, uses the specified single user operating system/mode specified on the validation certificate, or another compatible single user operating system, or that the GPC or platform for the firmware module or firmware controlling portion of the hybrid module, uses the specified operating system on the validation certificate, and

2. For **Level 2 Operational Environment**, a software cryptographic module will remain compliant with the FIPS 140-2 validation when operating on any GPC provided that the GPC incorporates the specified CC evaluated EAL2 (or equivalent) operating system/mode/operational settings or another compatible CC evaluated EAL2 (or equivalent) operating system with like mode and operational settings.

The CMVP allows user porting of a validated software, firmware or hybrid cryptographic module to a operational environment which was not included as part of the validation testing. The validation status is maintained in the new operational environment without retesting in the new operational environment as long as the porting rules are followed. However, the CMVP makes no statement as to the correct operation of the module when executed in an operational environment not listed on the validation certificate.

Additional Comments

Users include third party integrators or any entity that is the not originating vendor as specified on the validation certificate.

Note 1: A user may post-validation recompile a module if the unmodified source code is available and the module's Security Policy provides specific guidance on acceptable recompilation methods to be followed as a specific exception to this guidance. The methods in the Security Policy must be followed without modification to maintain validation under this guidance.

G.6 Modules with both a FIPS mode and a non-FIPS mode

Applicable Levels:	*All*
Original Publishing Date:	*03/11/1998*
Effective Date:	*03/11/1998*
Last Modified Date:	*04/02/1998*
Relevant Assertions:	*General*
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Question/Problem

How can a module be defined, when it includes both FIPS-approved and non-FIPS approved security methods?

Resolution

A module that contains both FIPS-approved and non-FIPS approved security methods **shall** have at least one "FIPS mode of operation" - which *only* allows for the operation of FIPS-approved security methods. This means that when a module is in the "FIPS mode", a non-FIPS approved method **shall not** be used in lieu of a FIPS-approved method (For example, if a module contains both MD5 and SHA-1, then when hashing is required in the FIPS mode, SHA-1 shall be used.). The operator must be made aware of which services are FIPS 140-2 compliant.

The FIPS 140-2 validation certificate will identify the cryptographic module's "FIPS mode" of operation.

The selection of "FIPS mode" does not have to be restricted to any particular operator of the module. However, each operator of the module must be able to determine whether or not the "FIPS mode" is selected.

There is no requirement that the selection of a "FIPS mode" be permanent.

Additional Comments

G.7 Relationships Among Vendors, Laboratories, and NIST/CSEC

Applicable Levels:	*All*
Original Publishing Date:	*04/14/1998*
Effective Date:	*04/14/1998*
Last Modified Date:	*04/14/1998*
Relevant Assertions:	*General*
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Question/Problem

What is the Cryptographic Module Validation Program policy regarding the relationships among vendors, testing laboratories, and NIST/CSEC?

Resolution

The CST laboratories are accredited by NVLAP to perform cryptographic module validation testing to determine compliance with FIPS 140-2. NIST/CSEC rely on the CST laboratories to use their extensive validation testing experience and expertise to make sound, correct, and independent decisions based on 140-2, the Derived Test Requirements, and Implementation Guidance. Once a vendor is under contract with a laboratory, NIST/CSEC will only provide official guidance and clarification for the vendor's module through the point of contact at the laboratory.

In a situation where the vendor and laboratory are at an irresolvable impasse over a testing issue, the vendor may ask for clarification/resolution directly from NIST/CSEC. The vendor should use the format required by Implementation Guidance G.1 and the point of contact at the laboratory ***shall*** be carbon copied. All correspondence from NIST/CSEC to the vendor on the issue will be issued through the laboratory point of contact.

Additional Comments

G.8 Revalidation Requirements

Applicable Levels:	*All*
Original Publishing Date:	*08/17/2001*
Effective Date:	*08/17/2001*
Last Modified Date:	*08/02/2010*
Relevant Assertions:	*General*
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Question/Problem

What is the Cryptographic Module Validation Program (CMVP) policy regarding revalidation requirements and validation of a new cryptographic module that is significantly based on a previously validated module?

Resolution

An updated version of a previously validated cryptographic module can be considered for a *revalidation* rather than a *full validation* depending on the extent of the modifications from the previously validated version of the module. (Note: the updated version may be, for example, a new version of an existing crypto module or a new model based on an existing model.)

A cryptographic module that is changed under change scenarios 1, 2 and 4 below, must meet ALL standards, implementation guidance and algorithm testing that were met at the time of original validation. A module does not need to continue to meet requirements that were removed or added since the time of original validation.

A cryptographic module that is changed under change scenarios 3 and 5 below, must meet ALL standards, implementation guidance and algorithm testing in effect at the time of module report submission to the CMVP. The CST laboratory is responsible for requesting from the vendor all the documentation necessary to determine whether the cryptographic module meets the current standards and IGs. This is particularly important for features/services of the cryptographic module that required a specific ruling from NIST and CSEC.

For example, a cryptographic module may have been validated with an implementation of Triple-DES that has not been tested. If the same cryptographic module is later submitted for revalidation under scenarios 3 and 5, this Triple-DES implementation must be tested and validated against FIPS 46-3, and the cryptographic module must meet the applicable FIPS 140-2 requirements, e.g., self-tests.

There are five possible **change** scenarios:

1. Modifications are made to hardware, software or firmware components **that do not affect any FIPS 140-1 or FIPS 140-2 security relevant items**. The vendor is responsible for providing the applicable documentation to the CST laboratory, which identifies the modification(s). Documentation may include a previous validation report, design documentation, source code, etc. The CST laboratory **shall** review the vendor-supplied documentation and identify any additional documentation requirements. The CST laboratory **shall** also determine additional testing as required to confirm that FIPS 140-1 or FIPS 140-2 security relevant items have not been affected by the modification.

   Upon successful review and applicable testing as required, the CST laboratory **shall** submit a signed explanatory letter that contains a description of the modification(s) and lists the affected TEs and their associated laboratory assessment. The assessment **shall** include the analysis performed by the laboratory that confirms that no security relevant items were affected. The letter **shall** also indicate whether the modified cryptographic module replaces the previously validated module or adds to the latter. If new algorithm certificates were obtained, they **shall** be listed.

   Upon a satisfactory review by NIST and CSEC, the updated version or release information will be posted on the *Validated FIPS 140-1 and FIPS 140-2 Cryptographic Module List* web site entry associated with the original cryptographic module. A new certificate will not be issued.

   It is *strongly* encouraged that a new security policy be provided for posting that updates the module version number with the new version number if applicable.

   The submission at a minimum shall consist of an encrypted ZIP file containing the unsigned letter <PDF>, image of the signed letter <PDF> and the _vendor.txt file. The ZIP file and files within the ZIP file **shall** follow all programmatic naming conventions and submitted to the CMVP using the specified encryption methods.

   Please refer to CMVP FAQ Section 5.8 for other non-security relevant change requests.

2. No modifications are made to any hardware, software or firmware components of the cryptographic module. All version information is unchanged. Post validation, Approved security relevant functions or services for which testing was not available at the time of validation, or security relevant functions or services that were not tested during the original validation, are now tested and are being submitted for inclusion as a FIPS Approved function or service. The CST laboratory is responsible for identifying the documentation that is needed to determine whether a revalidation is sufficient and the vendor is responsible for submitting the requested documentation to the CST laboratory. Documentation may include a previous validation report and applicable NIST and CSEC rulings, design documentation, source code, etc.

   The CST laboratory **shall** identify the assertions affected and **shall** perform the tests associated with those assertions. This will require the CST laboratory to:

   a. Review the COMPLETE list of assertions for the module embodiment and security level;
   b. Identify, from the previous validation report, the assertions that are newly tested;
   c. Identify additional assertions that were previously tested but should now be re-tested; and
   d. Review assertions where specific Implementation Guidance (IG) was provided at the time of the original validation to confirm that the IG is still applicable.

   The CST laboratory does not need to perform the regression test suite of operational tests since there is no change to the module.

   The CST laboratory **shall** document the test results in the associated assessments and all affected TEs **shall** be annotated as “re-tested.” The CST laboratory **shall** submit a test report as specified in G.2 describing the modification and highlighting those assertions that have been newly tested and retested (selecting the re-tested option in CRYPTIK). A new security policy **shall** be provided for posting that updates the new services or functions that are now included in an Approved mode of operation. Upon a satisfactory review by NIST and CSEC, the updated security policy and information will be posted on the *Validated FIPS 140-1 and FIPS 140-2 Cryptographic Module List* web site entry associated with the original cryptographic module. If new algorithm certificates were obtained, they **shall** be listed. A new certificate will not be issued.

3. Modifications are made to hardware, software or firmware components **that affect some of the FIPS 140-2 security relevant items**. An updated cryptographic module can be considered in this scenario if it is similar to the original module with only minor changes in the security policy and FSM, and less than 30% of the modules security relevant features[2]. The CST laboratory is responsible for identifying the documentation that is needed to determine whether a revalidation is sufficient and the vendor is responsible for submitting the requested documentation to the CST laboratory. Documentation may include a previous validation report and applicable NIST and CSEC rulings, design documentation, source code, etc.

   The CST laboratory **shall** identify the assertions affected by the modification and **shall** perform the tests associated with those assertions. This will require the CST laboratory to:

   a. Review the COMPLETE list of assertions for the module embodiment and security level,
   b. Identify, from the previous validation report, the assertions that have been affected by the modification,
   c. Identify additional assertions that were NOT previously tested but should now be tested due to the modification, and
   d. Review assertions where specific Implementation Guidance (IG) was provided to confirm that the IG is still applicable.

   For example, a revision to a firmware component that added security functionality may require a change to assertions in Section 1.

   In addition to the tests performed against the affected assertions, the CST laboratory **shall** also perform the regression test suite of operational tests included in Table G.8.1 – Regression Test Suite.

   When a cryptographic module is tested for revalidation from FIPS 140-1 to FIPS 140-2, the CST laboratory may re-use information contained in the FIPS 140-1 test report for the preparation of the FIPS 140-2 test report. The table found in Mapping FIPS 140-2 to FIPS 140-1 can be used to guide the tester.

   Note: Included in the table are the ASs, TEs, VEs (AS2 for FIPS 140-2 and AS1 for FIPS 140-1, etc.), security level(s), single chip (S), multi chip embedded (ME), multi chip standalone (MS), operational test (Op - x is used for the operational tests, r is used for regression test), applicable to FIPS 140-2 (M - match), and comment (describes the applicability of FIPS 140-1 results to FIPS 140-2, and may include info on the FIPS 140-2 requirement). The CST laboratory **shall** perform all the operational tests (TEs labeled with an x and an r in the Op field).

   The CST laboratory must provide a summary of the changes and rationale of why this meets the <30% guideline. The CMVP upon review, may determine that the changes are >30% and **shall** be submitted as a full report. The CST laboratory **shall** document the test results in the associated assessments and all affected TEs **shall** be annotated as "re-tested." The CST laboratory **shall** submit a test report as specified in G.2 describing the modification and highlighting those assertions that have been modified and retested (selecting the re-tested option in CRYPTIK). Upon a satisfactory review by NIST and CSEC, the updated version will be revalidated to FIPS 140-2. A new certificate will be issued.

[2] For example, security relevant features may include addition/deletion/change of minor components and their composition, addition/deletion of ports and interfaces, addition/delete/modification of security functions, modification of the physical boundary and protection mechanisms. These changes may affect many TE's yet be considered a minor change (<30%), or affect few TE's yet be a gross change (>30%).

4. Modifications are made only **to the physical enclosure of the cryptographic module that provides its protection and involves no operational changes to the module**. The CST laboratory is responsible for ensuring that the change only affects the physical enclosure (integrity) and has no operational impact on the module. The CST laboratory must also fully test the physical security features of the new enclosure to ensure its compliance to the relevant requirements of the standard. The CST laboratory must then submit a letter to NIST and CSEC that:

   a. Describes the change (pictures may be required),
   b. States that it is a security relevant change,
   c. Provides sufficient information supporting that the physical only change has no operational impact,
   d. Describes the tests performed by the laboratory that confirm that the modified enclosure still provides the same physical protection attributes as the previously validated module. For security levels 2, 3 and 4, the submission of an updated Physical Security Test Report is mandatory.

   Each request will be handled on a case-by-case basis. The CMVP will accept such letters against cryptographic modules already validated to FIPS 140-1 and FIPS 140-2. A new certificate will not be issued[3].

   It is *strongly* encouraged that a new security policy be provided for posting that updates the module version number with the new version number.

   The submission at a minimum shall consist of an encrypted ZIP file containing the unsigned letter <PDF>, image of the signed letter <PDF> and the _vendor.txt file. The ZIP file and files within the ZIP file **shall** follow all programmatic naming conventions and submitted to the CMVP using the specified encryption methods.

   An example of such a change could be the plastic encapsulation of the Level 2 token which has been reformulated or colored. Therefore the molding or cryptographic boundary has been modified. This change is security relevant as the encapsulation provides the opacity and tamper evidence requirements. But this can be handled as a letter only change with evidence that the new composition has the same physical security relevant attributes as the prior composition.

5. If modifications are made to hardware, software, or firmware components **that do not meet the above criteria**, then the cryptographic module will be considered a new module and must undergo a full validation testing by a CST laboratory. The CST laboratory **shall** submit a test report as specified in G.2.

If the overall Security Level of the crypto module changes or if the physical embodiment changes, e.g., from multi-chip standalone to multi-chip embedded, then the cryptographic module will be considered a new module and must undergo full validation testing by a CST laboratory.

Table G.8.1 – Regression Test Suite

Regression Testing Table					
AS	**TE**	**Security Level**			
		1	**2**	**3**	**4**
Section 1 - Cryptographic Module Specification					
AS01.03	TE.01.03.02	x	x	x	x
Section 2 - Cryptographic Module Ports and Interfaces					
AS02.06	TE02.06.02	x	x	x	x

[3] A certificate may be issued on a case by case basis.

	TE02.06.04	x	x	x	x
AS02.13	TE02.13.03	x	x	x	x
AS02.14	TE02.14.02	x	x	x	x
AS02.16	TE02.16.02			x	x
AS02.17	TE02.17.02			x	x
Section 3 - Roles, Services and Authentication					
AS03.02	TE03.02.02	x	x	x	x
	TE03.02.03	x	x	x	x
AS03.12	TE03.12.03	x	x	x	x
AS03.13	TE03.13.02	x	x	x	x
AS03.14	TE03.14.02	x	x	x	x
AS03.15	TE03.15.02	x	x	x	x
AS03.17	TE03.17.02		x		
AS03.18	TE03.18.02		x		
AS03.19	TE03.19.02			x	x
	TE03.19.03			x	x
AS03.21	TE03.21.02	x	x	x	x
AS03.22	TE03.22.02		x	x	x
AS03.23	TE03.23.02	x	x	x	x
Section 4 - Finite State Model					
AS04.03	TE.04.03.01	x	x	x	x
AS04.05	TE04.05.08	x	x	x	x
Section 5 - Physical Security					
	NONE				
Section 6 - Operational Environment					
AS06.05	TE06.05.01	x			
AS06.06	TE06.06.01	x			
AS06.07	TE06.07.01	x	x	x	x
AS06.08	TE06.08.02	x	x	x	x
AS06.11	TE06.11.02		x	x	x
	TE06.11.03		x	x	x
AS06.12	TE06.12.02		x	x	x
	TE06.12.03		x	x	x
AS06.13	TE06.13.02		x	x	x
	TE06.13.03		x	x	x
AS06.14	TE06.14.02		x	x	x
	TE06.14.03		x	x	x
AS06.15	TE06.15.02		x	x	x
AS06.16	TE06.16.02		x	x	x
AS06.17	TE06.17.02		x	x	x
AS06.22	TE06.22.02			x	x
	TE06.22.03			x	x
AS06.24	TE06.24.02			x	x
	TE06.24.03			x	x
AS06.25	TE06.25.02			x	x

Section 7 - Cryptographic Key Management					
AS07.01	TE07.01.02	x	x	x	x
AS07.02	TE07.02.02	x	x	x	x
AS07.15	TE07.15.02	x	x	x	x
	TE07.15.03	x	x	x	x
	TE07.15.04	x	x	x	x
AS07.25	TE07.25.02	x	x	x	x
AS07.27	TE07.27.02	x	x	x	x
AS07.28	TE07.28.02	x	x	x	x
AS07.29	TE07.29.02	x	x	x	x
AS07.31	TE07.31.04			x	x
AS07.39	TE07.39.02	x	x	x	x
AS07.41	TE07.41.02	x	x	x	x
Section 8 - EMI / EMC					
	As Required				
Section 9 - Self Tests					
AS09.04	TE09.04.03	x	x	x	x
AS09.05	TE09.05.03	x	x	x	x
AS09.09	TE09.09.02	x	x	x	x
AS09.10	TE09.10.02	x	x	x	x
AS09.12	TE09.12.02	x	x	x	x
AS09.22	TE09.22.07	x	x	x	x
AS09.35	TE09.35.05	x	x	x	x
AS09.40	TE09.40.03	x	x	x	x
	TE09.40.04	x	x	x	x
AS09.45	TE09.45.03	x	x	x	x
AS09.46	TE09.46.03	x	x	x	x
Section 10 - Design Assurance					
AS10.03	TE10.03.02	x	x	x	x
Section 11 - Mitigation of Other Attacks					
	NONE				
Appendix C - Cryptographic Module Security Policy					
	As Required				

Additional Comments

G.9 FSM, Security Policy, User Guidance and Security Officer Guidance Documentation

Applicable Levels:	*All*

Original Publishing Date:	*05/29/2002*
Effective Date:	*05/29/2002*
Last Modified Date:	*05/29/2002*
Relevant Assertions:	*General*
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Question/Problem

May a CST laboratory create original documentation specified in FIPS 140-2? The specific documents in question are the FSM, Security Policy, User Guidance and Security Officer Guidance.

Resolution

<u>FSM and Security Policy:</u>

A CST laboratory may take existing vendor documentation for an existing cryptographic module (post-design and post-development) and consolidate or reformat the existing information (from multiple sources) into a set format. If this occurs, NIST and CSEC **shall** be notified of this when the validation report is submitted. Additional details for the individual documents are provided below.

FSM: The vendor-provided documentation must readily provide a finite set of states, a finite set of inputs, a finite set of outputs, a mapping from the sets of inputs and states into the set of states (i.e., state transitions), and a mapping from the sets of inputs and states onto the set of outputs (i.e., an output function).

Security Policy: The vendor-provided documentation must readily provide a precise specification of the security rules under which a cryptographic module must operate, including the security rules derived from the requirements of FIPS 140-2 and the additional security rules imposed by the vendor.

In addition, a CST laboratory must be able to show a mapping from the consolidated or reformatted FSM and/or Security Policy back the original vendor source documentation. The mapping(s) must be maintained by the CST laboratory as part of the validation records.

Consolidating and reforming are defined as follows:

- The original source documents were prepared by the vendor (or a subcontractor to the vendor) and submitted to the CST laboratory with the cryptographic module.

- The CST laboratory extracts applicable technical statements from the original source documentation to be used in the FSM and/or Security Policy. The technical statements may **only** be reformatted to improve readability of the FSM and/or Security Policy. The content of the technical statements must not be altered.

- The CST laboratory may develop transitional statements in the FSM and/or Security Policy to improve readability. These transitional statements **shall** be specified as developed by the CST laboratory in the mapping.

User Guidance and Security Officer Guidance:

A CST laboratory may create User Guidance, Security Officer Guidance and other non-design related documentation for an existing cryptographic module (post-design and post-development). If this occurs, NIST and CSEC **shall** be notified of this when the validation report is submitted.

Additional Comments

G.10 Physical Security Testing for Re-validation from FIPS 140-1 to FIPS 140-2

Applicable Levels:	*All*
Original Publishing Date:	*03/29/2004*
Effective Date:	*03/29/2004*
Last Modified Date:	*03/29/2004*
Relevant Assertions:	*General*
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Background

FIPS 140-2 IG G.2 specifies that all report submissions must include a separate physical security test report section for Levels 2, 3 or 4.

Question/Problem

Questions have been asked regarding re-validation test reports where a previous separate physical security test report may not have existed or evidence such as images, etc. had not been provided with the original validation test report. What should the CST laboratory provide if the physical security requirements have not changed?

Resolution

If a previous *separate* physical security test report did not exist for the module undergoing re-validation testing and the physical security features of the module have not changed, the CST laboratory must compile the physical security test evidence that has been maintained from their records from the original tested module and create and submit a new *separate* physical security test report. If the records no longer exist because they were generated outside the period of the CST laboratories record retention period specified in the quality manual, then re-testing **shall** be required to provide such evidence. It is not required that a CST laboratory perform re-testing simply to create new photographic images that may not have been saved or generated during the original testing

Additional Comments

If the CST laboratory was not the original testing laboratory and therefore does not have access to the previous test records, then the module **shall** be re-tested to be able to provide such evidence. Without the prior records, the new CST laboratory cannot make a determination that the physical security has or has not changed.

G.11 Testing using Emulators and Simulators

Applicable Levels:	*All*
Original Publishing Date:	*09/12/2005*
Effective Date:	*09/12/2005*
Last Modified Date:	*09/12/2005*
Relevant Assertions:	*General*

Relevant Test Requirements:	
Relevant Vendor Requirements:	

Background

Vendors of cryptographic modules use independent, accredited Cryptographic and Security Testing (CST) laboratories to have their modules tested for conformance to the requirements of FIPS 140-2. Organizations wishing to have testing performed would contract with the laboratories for the required services. The Derived Test Requirements (DTR) document describes the methods that will be used by accredited laboratories to test whether the cryptographic module conforms to the requirements of FIPS 140-2. It includes detailed procedures, inspections, documentation and code reviews, and operational and physical tests that the tester must follow, and the expected results that must be achieved for the cryptographic module to satisfy its conformance to the FIPS PUB 140-2 requirements. These detailed methods are intended to provide a high degree of objectivity during the testing process and to ensure consistency across the accredited testing laboratories.

Definitions:

An **emulator** attempts to “model” or “mimic” the behavior of a cryptographic module. The correctness of the emulators' behavior is dependant on the inputs to the emulator and how the emulator was designed. It is not guaranteed that the actual behavior of the cryptographic module is identical, as many other variables may not be modeled correctly or with certainty.

A **simulator** exercises the actual module source code (e.g., VHDL code) prior to physical entry into the module (e.g., an FPGA or custom ASIC). From a behavioral perspective, the behavior of the source code within the simulator may be logically identical when placed into the module or instantiated into logic gates. However, many other variables exist that may alter the actual behavior (e.g. path delays, transformation errors, noise, environmental, etc). It is not guaranteed that the actual behavior of the cryptographic module is identical, as many other variables may not be identified with certainty.

Question/Problem

May a CST laboratory tester use module emulation and/or simulation methods to perform cryptographic module testing?

Resolution

There are three broad areas of focus during the testing of a cryptographic module: operational testing of the module at the defined boundary of the module, algorithm testing and operational fault induction error testing.

1. Operational Testing

   Emulation or simulation is prohibited for the operational testing of a cryptographic module. Actual testing of the cryptographic module must be performed utilizing the defined ports and interfaces and services that a module provides.

2. Operational Fault Induction

   An emulator or simulator may be utilized for fault induction to test a cryptographic module’s transition to error states as a complement to the already allowed source code review. Rationale must be provided for the applicable TE why a method does not exist to induce the actual module into the error state for testing.

3. Algorithm Testing

   Algorithm testing utilizing the defined ports and interfaces and services that a module provides is the preferred method. This method most clearly meets the requirements of FIPS 140-2 IG 1.4.

   If this preferred method is not possible where the module's defined set of ports and interfaces and services do not allow access to internal algorithmic engines, two alternative methods may be utilized:

   a. A module may be modified by the CST laboratory for testing purposes to allow access to the algorithmic engines (e.g. test jig, test API), or

   b. A module simulator may be utilized.

   When submitting the algorithm test results to the CAVP, the actual operational environment on which the testing was performed must be specified (e.g. including modified module identification or simulation environment). When submitting the module test report to the CMVP, **AS01.12** must include rationale explaining why the algorithm testing was not conducted on the actual cryptographic module.

   An emulator may not be used for algorithm testing.

Additional Comments

G.12 Post-Validation Inquiries

Applicable Levels:	*All*
Original Publishing Date:	*01/26/2007*
Effective Date:	*01/26/2007*
Last Modified Date:	*01/26/2007*
Relevant Assertions:	*General*
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Background

FIPS 140-2 conformance testing that is performed by the accredited Cryptographic and Security Testing (CST) laboratories and validation of those test results by NIST and CSEC provide a level of assurance that a module conforms to the requirements of FIPS 140-2 and other underlying standards.

Once a module is validated and posted on the NIST CMVP web site, many parties review and scrutinize the merits of the validation. These parties may be potential procurers of the module, competitors, academics or others.

If a party performing a post-validation review believes that a conformance requirement of FIPS 140-2 has not been met and was not determined during testing or subsequent validation review, the party may submit a inquiry to the CMVP for review.

Question/Problem

What is the procedure and process for submitting an inquiry for review and how is the review performed? If a review is determined to have merit, what actions may be taken regarding the module's validation status?

Resolution

An *Official Request* must be submitted to the CMVP in writing with signature following the guidelines in FIPS 140-2 IG G.1. If the requestor represents an organization, the official request must be on the organization's letterhead. The assertions must be objective and not subjective. The module must be identified by reference to the validation certificate number(s). The specific technical details must be identified and the relationship to the specific FIPS 140-2 Derived Test Requirements assertions must be identified. The request must be non-proprietary and not prevent further distribution by the CMVP.

The CMVP will distribute the unmodified official request to the CSTL that performed the conformance testing of the identified module. The CSTL may choose to include participation of the vendor of the identified module during its determination of the merits of the inquiry. Once the CSTL has completed its review, it will provide to the CMVP a response with rationale on the technical validity regarding the merits of the official request. The CSTL will state its position whether its review of the official request regarding the module:

1. is without merit and the validation of the module is unchanged.
2. has merit and the validation of the module is affected. The CSTL will further state its recommendations regarding the impact to the validation.

The CMVP will review the CSTLs position and rationale supporting its conclusion.

If the CMVP concurs that the official request is without merit, no further action is taken.

If the CMVP concurs that the official request has merit, a security risk assessment will be performed regarding the non-conformance issue.

Additional Comments

G.13 Instructions for completing a FIPS 140-2 Validation Certificate

Applicable Levels:	*All*
Original Publishing Date:	*06/28/2007*
Effective Date:	*06/28/2007*
Last Modified Date:	*06/10/2010*
Relevant Assertions:	*General*
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Question/Problem

How are the various fields on a FIPS 140-2 validation certificate presented and provided to the CMVP for validation?

Resolution

During the pre-validation testing by an accredited cryptographic module testing laboratory, the CMVP supplied CRYPTIK tool is used to create the pre-validation draft certificate. The information to be presented on the validation certificate is entered through the Module Information screen. This draft certificate is presented to the CMVP for review and validation along with the other report components identified in G.2.

These instructions describe the presentation of the information on the certificate via entry in CRYPTIK.

File Naming: The name of the file, which contains the draft validation certificate, must be re-named from the CRYPTIK output (certificate.rtf) to the following format specified in the *CMVP Convention for E-mail Submittal*:

TID-nn[4]-nnnn[5]-nnnn[6]-140crtxxxx[7].doc

Front of the Validation Certificate

1. [**CRYPTO MODULE NAME**] - the complete name of the cryptographic module. Do not include the version number with the name. The name of the cryptographic module **shall** be consistent with IG 1.1 and the name found in the security policy and test report. Include all necessary ™, ® and © symbols. CRYPTIK may not accept or pass the special symbols and therefore they may need to be added manually into the .doc file.

   Examples: **Crypto Acceleration Token**
   Secure Cryptographic ToolKit™
   Best Crypto©

   If the test report represents multiple modules, list all module names. If this requires the use of the word "and", then the word "and" **shall** be italicized. CRYPTIK cannot output italicized fonts, so this must be performed manually into the exported .doc file.

   Examples: **Crypto Sensor AM-5000 *and* AM-5010**
   Crypto 8000 PCI, Crypto 9000 PCI *and* Crypto Plus++ PCI

2. [***by* Vendor Name**] - the name of the vendor (including Corp., Inc., Ltd., etc) that developed the cryptographic module.

   Examples: ***by* AcmeSecurity, Inc.**
   ***by* Acmeproducts, Ltd.**
   ***by* AcmeSecurity, Inc. *and* Acmeproducts, Ltd.**

   Note the italicized words "by" and "and": CRYPTIK cannot output italicized fonts, so this must be performed manually after exporting from CRYPTIK.

3. [**(applicable caveat)**] - This caveat may be modified or expanded by the CMVP during the validation process.

   Cryptographic modules may not have a caveat if the module only has a single FIPS Approved mode of operation.

[4] nn is the 2-digit CST laboratory code
[5] nnnn is the CST laboratory assigned TID
[6] nnnn is the CSEC TID
[7] xxxx is the assigned certificate number (if available)

Examples: **When operated in FIPS mode**
Added if the module can also operate in a non-FIPS mode.

When operated in FIPS mode with module [module name] validated to FIPS 140-2 under Cert. #xxxx operating in FIPS mode
Added if the module's validation is bound to another validated cryptographic module.

> **Example:** 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.

For services provided by the FIPS-Approved algorithms listed on the reverse

The <tamper evident seals> and <security devices> installed as indicated in the Security Policy

When operated in FIPS mode and initialized to Overall Level 2 per Security Policy
Added if the module can be initialized to different overall levels.

> **Example:** A module can be initialize to either support Level 2 role-based authentication or initialized to support only Level 3 identity-based authentication.

This module contains the embedded module [module name] validated to FIPS 140-2 under Cert. #xxxx operating in FIPS mode
Added if the module incorporates an embedded validated cryptographic module.

> **Example:** 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:** A hardware cryptographic module which has embedded within its physical boundary a validated cryptographic module.

Back of the Validation Certificate

4. [**CRYPTO MODULE NAME *by* Vendor Name**] - the name of the cryptographic module and the vendor that developed the module. The complete name of the vendor **shall** be used (e.g., Corp., Inc., LTD.) This information **shall** match the information listed in items 1 and 2 above. Note the italic font between the module name and the vendor name. If there is more than one module name on the certificate that requires the use of the word "and", then the word "and" is also italicized. CRYPTIK cannot output italicized fonts, so this must be performed manually after exporting from CRYPTIK.

   The FIPS 140-1 and FIPS 140-2 Vendor Listing is an alphabetical list of vendors who have implemented validated cryptographic modules. It is desirable that the vendor name be consistent on validation certificates issued for modules from the same vendor. The listing can be found at: http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401vend.htm

5. [(***Version No. nnn***;] - the version number of the crypto module. This number **shall** be of sufficient level such that updates/upgrades/changes **shall** be reflected in a version change. For example, version 4 may not be sufficient if the releases are numbered 4.0, 4.1, 4.2, etc. The version number may also include letters, for example, 4.0a, 4.0b, 4.0c, etc. This **shall** include the version numbers for each element; hardware, software, and firmware, if applicable. Each elements version number (e.g. hardware, firmware, software) **shall** be separated by a semi-colon. If a module does not include an element, leave the field blank; do not enter "NA". The version numbers **shall** be the same as the ones found in the security policy. For example, hardware version: 4.2; software version: 4.0a. If there are several version numbers, the word "version" **shall** be pluralized. Note the italic font for the version numbers in the examples found in section 6 below. CRYPTIK cannot output italicized fonts, so this must be performed manually after exporting from CRYPTIK.

   If there are multiple modules listed on the certificate, or if there are multiple part numbers with different versions of firmware for example, brackets **shall** be used to clearly indicate the pairings.

6. [**module type)**] - the module type is one of the following: **Hardware, Firmware, Software**, **Software-Hybrid** or **Firmware-Hybrid**. If a module is hardware with embedded software and/or firmware, the modules type is simply labeled Hardware. Note the non-italicized font of the module type.

   Examples: **(*Hardware Version: 4.2; Software Version: 4.0a*; Hardware)**
   Hardware module with software embedded within it.

   (*Hardware Versions: 5.2 and 5.3, Build 3; Firmware Version: 2.45*; Hardware)
   Two different hardware modules, each with the same embedded firmware.

   (*Hardware Versions: 5.2 [1] and 5.3 [2], Build 3; Firmware Versions: 2.45 [1] and 2.50 [2]*; Hardware)
   Two different hardware modules each with the specified version of embedded firmware.

   (*Hardware Version: 88X8868; Software Version: 1.0*; Software-Hybrid)
   Software hybrid module referencing the hardware and disjoint software components.

   (*Hardware Version: BN45; Firmware version 1.0; Software Version 2.0;* Software-Hybrid)
   Software hybrid module referencing the hardware and disjoint software versions. The hardware component also has firmware embedded within it.

   (*Hardware Version: 88X8686; Firmware Version 1.4;* Firmware-Hybrid)
   Firmware hybrid module referencing both the hardware and disjoint firmware versions.

   Note the use of the comas, semi-colons and colons.

7. [**(applicable NPIVP Cert. #)**] When a module implements a validated PIV application, the application validation certificate type and number **shall** be indicated under the module version number and module type line as:

   (PIV Card Application: Cert. #nnn)

8. [***Lab Name,***] - the name of the CST laboratory.

9. ***NVLAP LAB CODE*** [*999999-9*] - the code assigned by NVLAP to the CST laboratory

10. **CRYPTIK Version** [**x.xx**] - the version of the CRYPTIK tool used to create the report

11. **Level [n]** - for each of the 11 areas, include the specific level. For FIPS 140-2, the Operating System Security Level, the Physical Security level and Mitigation of Other Attacks level may not be applicable

and if so, **shall** be marked as N/A.

If a module meets Level 3 Physical Security and also has been tested for EFP and/or EFT, this **shall** be annotated on the certificate as: **Level 3 +EFP** or **+EFT** or **+EFP/EFT**

12. [***(embodiment type)***] - the cryptographic module **shall** be specified as one of the three types: **Multi-chip Standalone**, **Multi-chip Embedded**, or **Single-chip**, in this format.

13. **tested in the following configuration(s)**: - the specific configuration(s) that was(were) used during testing by the lab. This **shall** match the information in the test report in **AS01.08**.

    For a *software* cryptographic module at Security Level 1, the test platform does not need to be specified but the caveat "(single-user mode) must be included. For a *software* cryptographic module at Security Level 2, the test platform needs to be specified. For Java applets, the Java environment (JRE, JVM) version needs to be specified for all Security Levels. For multiple operating environment entries, separate each with a semi-colon; do not use "and".

    Examples: **Microsoft Windows XP with SP2 (single-user mode)**
    Sun Solaris Version 2.6SE running on a Sun Ultra SPARC-1 workstation
    Microsoft Windows XP with SP2; HP-UX 11.23 (single user mode)

    The following example for a *firmware* cryptographic module; the certificate **shall** specify the hardware platform and operating system that was used for testing:

    Example: **BlackBerry® 7230 with BlackBerry OS® Versions 3.8, 4.0 and 4.1**

    If the *firmware* module's physical security meets FIPS 140-2 Section 4.5 Levels 2, 3 or 4, the hardware platform shall include applicable specific versioning information.

    Example: **Crypto Unit (Hardware Version: 1.0) with Little OS® Version 3.7b**

    The following example for a *software-hybrid* cryptographic module; the certificate **shall** specify the hardware platform and operating system that was used for testing:

    Example: **Debian GNU/Linux 4.0 (Linux kernel 2.6.17.13) running on 4402-A ViPr Desktop Terminal (single-user mode)**

    The following example for a *firmware-hybrid* cryptographic module; the certificate **shall** specify the hardware platform and operating system that was used for testing:

    Example: **BlackBerry 8700c with BlackBerry OS Version 4.2**

    If this field is not applicable, mark the field as N/A.

14. **The following FIPS Approved Cryptographic Algorithms are used**: - the Approved security functions included in the cryptographic module and utilized by the modules callable services or internal functions. The security function is listed and then the applicable algorithm Certificate number in parentheses. Do NOT include the modes or key lengths, e.g., ECB, CBC; 128 bits. All algorithm entries must be separated by semi-colons.

    If a module contains within it an already validated embedded cryptographic module, all Approved security functions that are used by the modules callable services and internal functions **shall** be annotated on the certificate (both those within the embedded module and in addition to the embedded module). Algorithms that are either in "dead code" or in the embedded module that are never called shall not be listed on the certificate.

The algorithm must meet all three (3) conditions to be listed as FIPS Approved:

1. Must be an Approved security function as specified in FIPS 140-2 Annex A;
2. Must meet all requirements of FIPS 140-2 (KAT, etc); and
3. Must be used in at least one FIPS Approved cryptographic function or service for that cryptographic algorithm in a FIPS Approved mode of operation.

Examples: **Triple-DES (Certs. #78 and #122); Triple-DES MAC (Triple-DES Cert. #78, vendor affirmed); SHS (Cert. #23); HMAC (Cert. #23); CCM (Cert. #3); KAS[8] (Cert. #3); KAS[9] (SP 800-56A, vendor affirmed, key agreement); DRBG[10] (Cert. #12); RNG[11] (Cert. #45); DSA[12] (Cert. #200); DSA[13] (FIPS 186-3, vendor affirmed); DSA (Cert. #200[12] and FIPS 186-3, vendor affirmed[13]); RSA[14] (Cert. #133); RSA[15] (FIPS 186-3, vendor affirmed); RSA (Cert. #133[14] and FIPS 186-3, vendor affirmed[15]); ECDSA[14] (Cert. #100); ECDSA[15] (FIPS 186-3, vendor affirmed); ECDSA (Cert. #100[14] and FIPS 186-3, vendor affirmed[15]); AES[16] (XTS-AES: AES (Cert. #500, vendor affirmed)**

For MAC, the certificate number must specify the underlying algorithm certificate and the “vendor affirmed” caveat.

For multiple certificate entries, the term “Cert” shall be pluralized (i.e., Certs), an “and” shall be placed between the last two certificate numbers and there shall be a “#” in front of each number.

Examples: **AES (Cert. #11); Triple-DES (Certs. #118 and #133); DSA (Cert. #132); SHS (Certs. #103, #115 and #119); RSA (Cert. #24); HMAC (Cert. #52); RNG (Cert. #33)**

15. **non-FIPS Approved algorithms**: - the non-FIPS Approved cryptographic algorithms implemented in the cryptographic module. Non-FIPS Approved algorithms may be *allowed* in a FIPS Approved mode of operation and will be identified in the module Security Policy. A non-Approved implementation may exist for what appears to be an Approved algorithm where a CAVP validation or the requirements of FIPS 140-2 (e.g. self-test) are not met. These non-Approved implementations will include the caveat “*non-compliant*” so that it is clear the algorithm implementation shall not be used in an Approved mode of operation.

    For DES and DES MAC, after May 19, 2007, these shall be listed as non-Approved without any additional caveat.

[8] Key Agreement Scheme
[9] Vendor Affirmed per IG D.1
[10] DRBG references compliance to NIST SP 800-90
[11] RNG references compliance to legacy RNGs (e.g. X9.31, FIPS 186-2)
[12] If *not* supporting the generation and validation of provably prime domain parameters p and q and canonical generation and validation of domain parameter g.
[13] If supporting only the generation and validation of provably prime domain parameters p and q and canonical generation and validation of domain parameter g.
[14] FIPS 186-2
[15] Vendor Affirmed per IG A.6
[16] Vendor Affirmed per IG A.7

Examples: **DES; MD5[17]; RC4; Blowfish; Diffie-Hellman[18]; Diffie-Hellman[19] (key agreement); EC Diffie-Hellman[19] (key agreement); AES[20] (non-compliant); DSA[21] (FIPS 186-3, non-compliant)**

For the non-FIPS Approved Diffie-Hellman and EC Diffie-Hellman examples: these examples are valid for legacy implementations; implementations that do not implement a KDF specified in NIST SP 800-56A but specified in IG 7.1 and meet AS.07.19; and where only the SP 800-56A DLC primitive is implemented.

For algorithms that are used both Approved and non-Approved (e.g. RSA), then it only needs to be listed once on the FIPS Approved line. The Security Policy **shall** indicate all uses of the algorithm. Exceptions are cases where there are caveats highlighting weaknesses in the use of algorithms.

Examples: **RSA (encrypt/decrypt)**
In this example, RSA is implemented and *only* used for encryption/decryption.

AES (Cert. nnn; non-compliant)
In this example, AES is implemented, has an algorithm certificate, but the KAT was not implemented and fails the FIPS 140-2 requirements.

AS.07.19 requires that the wrapping key used in key transport be equal or of greater strength than the wrapped key. If the strength of the largest key that can be established by a cryptographic module is greater than the comparable strength of the implemented key establishment method, then the module certificate and security policy **shall** be annotated with, in addition to the other required caveats, the caveat "**(key establishment methodology provides xx bits of encryption strength)**" for that key establishment method as allowed in IG 7.5 – *Strength of Key Establishment Methods*. No caveat is required if the wrapping key used in key transport be equal or of greater strength than the wrapped key.

If the module supports, for a particular key establishment method, a single strength, then the caveat **shall** state the strength provided by the keys.

Examples: **Diffie-Hellman (key agreement; key establishment methodology provides 80 bits of encryption strength)**

RSA (key wrapping; key establishment methodology provides 80 bits of encryption strength)

Triple-DES (Cert. #114, key wrapping; key establishment methodology provides 80 bits of encryption strength)

AES (Cert. #300, key wrapping; key establishment methodology provides 192 bits of encryption strength)

KAS (SP 800-56A, vendor affirmed; key establishment methodology provides 112 bits of encryption strength)

If a module *only* implements the 1024-bit and 2048-bit Diffie-Hellman then:

[17] May be allowed in an Approved mode of operation when used as part of an approved key transport scheme (e.g. SSL v3.1) where no security is provided by the algorithm
[18] If only the NIST SP 800-56A DLC primitive is implemented – allowed in an Approved mode of operation
[19] Allowed in an Approved mode of operation
[20] Not validated by the CAVP or the requirements of FIPS 140-2 are not met (e.g. self-test)
[21] Is not vendor affirmed to IG A.6.

Diffie-Hellman (key agreement; key establishment methodology provides 80 or 112 bits of encryption strength)

If a module implements several key sizes between 1024-bit and 15,360-bit Diffie-Hellman, then only the range end points are indicated:

Diffie-Hellman (key agreement; key establishment methodology provides between 80 and 256 bits of encryption strength)

If a module implements several key sizes between 1024-bit and 15,360-bit Diffie-Hellman, and also less than 80-bits of strength, then only the range end points are indicated and a caveat regarding the strength less than 80-bits:

Diffie-Hellman (key agreement; key establishment methodology provides between 80 and 256 bits of encryption strength; non-compliant less than 80-bits of encryption strength)

If a module implements only a key size less than 80-bits of strength (for example 56-bits), then only the caveat regarding the non-compliant strength less than 80-bits is provided:

Diffie-Hellman (non-compliant key agreement; key establishment methodology provides 56 bits of encryption strength)

If AES MAC is implemented for OTAR, it **shall** be specified as:

AES MAC (AES Cert. #2, vendor affirmed; P25 AES OTAR)

If AES MAC is implemented and not used for OTAR, it **shall** be specified as:

AES MAC (AES Cert. #2; non-compliant)

Note: In all cases, the CMVP report reviewer must ascertain the correctness of the added caveat(s) and the most accurate wording and the best interpretation to give to the Federal users.

If this field is not applicable, mark the field as N/A.

For non-Approved algorithms that have names similar to Approved security functions, the caveat "(non-compliant)" must be appended to alleviate misinterpretation.

Example: **AES (non-compliant)**
In this example, AES stands for Accelerated Encryption Scheme which is not AES specified in FIPS 197.

16. ***Overall Level Achieved*: [n]** – the overall level of the crypto module. This value is the *lowest* value of the individual levels.

Additional Comments

Section 1 - Cryptographic Module Specification

1.1 Cryptographic Module Name

Applicable Levels:	*All*
Original Publishing Date:	*02/27/2004*
Effective Date:	*02/27/2004*
Last Modified Date:	*02/27/2004*
Relevant Assertions:	*AS01.05, AS01.08 and AS01.09*
Relevant Test Requirements:	*TE01.08.03,04 and 05 and TE01.09.01 and 02*
Relevant Vendor Requirements:	*VE01.08.03 and VE01.09.01*

Question/Problem

How shall the name of a cryptographic module relate to the defined cryptographic boundary?

Resolution

The provided name of the cryptographic module (which will be on the validation certificate) **shall** be consistent with the defined cryptographic boundary as defined in the test report.

It is not acceptable to provide a module name that represents a module that has more components than the modules defined boundary. If it is desired to have a name that does represent a larger entity, then the cryptographic boundary must be consistent. All components residing within the cryptographic boundary must either be included (**AS.01.08**) or excluded (**AS.01.09**) in the test report.

Additional Comments

Example: The provided name of a cryptographic module is the *Crypto Card*. However, the defined cryptographic boundary in the test report is a small black encapsulated component placed in one corner of the card. The named card also has additional components that were not referenced (e.g. batteries, connectors). If the defined boundary in the test report specifies *ONLY* the black encapsulated component, it is clearly NOT the *Crypto Card*. A unique different name **shall** be provided to be consistent with the defined boundary. To represent the entire card, the boundary must be redefined and must include all the components and address them properly (include/exclude).

1.2 FIPS Approved Mode of Operation

Applicable Levels:	*All*
Original Publishing Date:	*03/15/2004*
Effective Date:	*03/15/2004*
Last Modified Date:	*09/12/2005*
Relevant Assertions:	*AS01.02, AS01.03 and AS01.04*
Relevant Test Requirements:	*TE01.03.01-02 and TE01.04.01-12*
Relevant Vendor Requirements:	*VE01.03.01-02 and VE01.04.01-02*

Definition

Approved mode of operation: a mode of the cryptographic module that employs only Approved security functions (not to be confused with a specific mode of an Approved security function, e.g., AES CBC mode).

Question/Problem

Are there any operational requirements when switching between modes of operation, either from an Approved mode of operation to a non-Approved mode of operation, or vice versa?

Resolution

In addition to the requirements specified in AS01.02, AS.01.03 and AS.01.04, a module **shall** not share CSPs between an Approved mode of operation and a non-Approved mode of operation.

Additional Comments

This separation mitigates the risk of untrusted handling of CSPs generated in an Approved mode of operation. Examples:

- a module may not generate keys in a non-Approved mode of operation and then switch to an Approved mode of operation and use the generated keys for Approved services. The keys may have been generated using non-Approved methods and their integrity and protection cannot be assured.
- a module **shall** not electronically import keys in plain text in a non-Approved mode of operation and then switch to an Approved mode of operation and use those keys for Approved services.
- a module may not generate keys in an Approved mode of operation and then switch to a non-Approved mode of operation and use the generated keys for non-Approved services. The integrity and the protection of the Approved keys cannot be assured in the non-Approved mode of operation.

1.3 Firmware Designation

Applicable Levels:	*All*
Original Publishing Date:	*04/28/2004*
Effective Date:	*04/28/2004*
Last Modified Date:	*06/10/2010*
Relevant Assertions:	*AS01.01*
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Background

Cryptographic module: the set of hardware, software, and/or firmware that implements Approved security functions (including cryptographic algorithms and key generation) and is contained within the cryptographic boundary.

Firmware: the programs and data components of a cryptographic module that are stored in hardware (e.g., ROM, PROM, EPROM, EEPROM or FLASH) within the cryptographic boundary and cannot be dynamically written or modified during execution.

The *operational environment* of a cryptographic module refers to the management of the software, firmware, and/or hardware components required for the module to operate. The operational environment can be non-modifiable (e.g., firmware contained in ROM, or software contained in a computer with I/O devices disabled), or modifiable (e.g., firmware contained in RAM or software executed by a general purpose computer).

A *limited operational environment* refers to a static non-modifiable virtual operational environment (e.g., JAVA virtual machine on a non-programmable PC card) with no underlying general purpose operating system upon which the operational environment uniquely resides.

If the operational environment is a limited operational environment, the operating system requirements in Section 4.6.1 do not apply.

Question/Problem

How shall a *software* cryptographic module running on a limited operational environment be designated as?

Resolution

If the Operational Environment is a limited operational environment, and is indicated as NA on the certificate, then the cryptographic module **shall** be designated as a ***firmware*** module.

Additional Comments

- The reference tested OS must be indicated on the validation certificate for all software and firmware cryptographic modules. It will be referenced on the CMVP validation list web page as follows:
  - If the Operational Environment is applicable: -*Operational Environment: Tested as meeting Level x with ...*
  - If the Operational Environment is NA: -*Tested: ...*
- For an overall Level 2, 3, or 4 module or where FIPS 140-2 Section 4.5 Physical Security is Level 2, 3 or 4, the reference hardware platform with appropriate specific versioning information used during operational testing **shall** also be listed. The certificate caveat shall minimally indicate: *When operated only on the specific platforms specified on the reverse*
- For JAVA applets, the tested JAVA environment (JRE, JVM) and operating system need to be specified for all Security Levels.

Per FIPS 140-2 IG G.5, porting of software modules is only applicable to modules operating on a General Purpose Computer (GPC) and when the Operational Environment is applicable. The module's validation will be maintained if no changes are made to underlying source code.

If the operational environment is not applicable, a firmware module at overall Level 1 (with FIPS 140-2 Section 4.5 Physical Security at Level 1) and its identified tested OS together may be ported from one platform to another platform while maintaining the module's validation (IG G.5). For firmware module's that are JAVA applets, the firmware module, its identified tested OS, and the tested JAVA environment (JRE, JVM) must be moved together when porting from one platform to another platform in order to maintain the module's validation.

For all other cases, the validation of the cryptographic module is not maintained if ported.

1.4 Binding of Cryptographic Algorithm Validation Certificates

Applicable Levels:	*All*
Original Publishing Date:	*01/21/2005*
Effective Date:	*01/21/2005*
Last Modified Date:	*01/21/2005*
Relevant Assertions:	*AS01.12*
Relevant Test Requirements:	*TE01.12.01*
Relevant Vendor Requirements:	*VE01.12.01*

Background

Cryptographic algorithm implementations are tested and validated under the Cryptographic Algorithm Validation Program (CAVP). The cryptographic algorithm validation certificate states the name and version number of the validated implementation, and the tested operational environment.

Cryptographic modules are tested and validated under the Cryptographic Module Validation Program (CMVP). The cryptographic module validation certificate states the name and version number of the validated cryptographic module, and the tested operational environment.

The validation certificate serves as a benchmark for the configuration and operational environment used during the validation testing.

Question/Problem

What are the configuration control and operational environment requirements for the cryptographic algorithm implementation(s) embedded within a cryptographic module when the latter is undergoing testing for compliance to FIPS 140-2?

Resolution

For a validated cryptographic algorithm implementation to be embedded within a software, firmware or hardware cryptographic module that undergoes testing for compliance to FIPS 140-2, the following requirements must be met:

1. the implementation of the validated cryptographic algorithm has not been modified upon integration into the cryptographic module undergoing testing; and

2. the operational environment under which the validated cryptographic algorithm implementation was tested by CAVS must be identical to the operational environment that the cryptographic module is being tested under by the CST laboratory.

Additional Comments

1.5 moved to A.1

1.6 moved to A.2

1.7 Multiple *Approved* Modes of Operation

Applicable Levels:	*All*
Original Publishing Date:	*09/12/2005*
Effective Date:	*09/12/2005*
Last Modified Date:	*09/12/2005*
Relevant Assertions:	*AS01.03 and AS01.04*
Relevant Test Requirements:	*TE01.03.01-02 and TE01.04.01-02*
Relevant Vendor Requirements:	*VE01.03.01-02 and VE01.04.01-02*

Background

Section 4.1 of FIPS PUB 140-2 does not preclude a vendor from implementing more than one Approved mode of operation in a cryptographic module. An example of multiple Approved modes of operation may be a module where all modes may not have the same set of services.

Question/Problem

May a module implement more than one Approved mode of operation? What are the requirements for a module to implement more than one Approved mode of operation?

Resolution

A cryptographic module may be designed to support multiple Approved modes of operation.

For a cryptographic module to implement more than one Approved mode of operation, the following **shall** apply:

- the overall security level can not be changed when configured for different Approved modes of operation;
- the security policy **shall** describe each Approved mode of operation implemented in the cryptographic module and how each one is configured;
- upon re-configuration from one Approved mode of operation to another, the cryptographic module **shall** reinitialize and perform a power on self-test;
- power on self-tests **shall** be performed for all Approved security functions used in the selected Approved mode of operation; and
- if re-configuration changes the physical security level of the module, upon re-configuration the cryptographic module **shall** perform a zeroization of all CSPs within the module.

To confirm the correct operation of the several modes of operation, the tester **shall**:

- verify the documentation describing each Approved mode of operation;
- use the vendor provided instructions described in the non-proprietary security policy to invoke each Approved mode of operation;
- verify that, for each Approved mode of operation, only the security functions implemented for that mode are accessible and that security functions not implemented for that mode are not;
- verify that the aforementioned requirements are met for each Approved mode of operation;
- verify that the requirements of AS.01.03 and/or AS.01.04 are met for each Approved mode of operation; and
- verify that CSPs are not shared between the multiple Approved modes of operation.

Additional Comments

1.8 Listing of DES Implementations

Applicable Levels:	*All*
Original Publishing Date:	*11/23/2005*
Effective Date:	*05/19/2007*
Last Modified Date:	*01/16/2008*
Relevant Assertions:	*AS01.12*
Relevant Test Requirements:	*TE01.12.01*
Relevant Vendor Requirements:	*VE01.12.01*

Background

DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
[Docket No. 040602169-5002-02]

Announcing Approval of the Withdrawal of Federal Information Processing Standard (FIPS) 46-3, Data Encryption Standard (DES); FIPS 74, Guidelines for Implementing and Using the NBS Data Encryption Standard; and FIPS 81, DES Modes of Operation.

Question/Problem

With the withdrawal of the DES cryptographic algorithm, how does the DES and DES MAC algorithms get listed on the FIPS 140-2 validation certificate?

Resolution

The DES transition period ended on May 19, 2007. DES and DES MAC are no longer Approved security functions and **shall** be listed on the FIPS 140-2 certificate as non-Approved algorithms.

Additional Comments

1.9 Definition and Requirements of a Hybrid Cryptographic Module

Applicable Levels:	*Level 1*
Original Publishing Date:	*03/10/2009*
Effective Date:	*03/10/2009*
Last Modified Date:	*03/19/2010*
Relevant Assertions:	*AS01.01 and AS01.08*
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Background

Cryptographic module: the set of hardware, software, and/or firmware that implements Approved security functions (including cryptographic algorithms and key generation) and is contained within the cryptographic boundary.

Software: the programs and data components within the cryptographic boundary, usually stored on erasable media (e.g., disk), that can be dynamically written and modified during execution.

Firmware: the programs and data components of a cryptographic module that are stored in hardware (e.g., ROM, PROM, EPROM, EEPROM or FLASH) within the cryptographic boundary and cannot be dynamically written or modified during execution.

Firmware Designation: FIPS 140-2 IG 1.3:

Question/Problem

Define what a ***hybrid*** cryptographic module is and specify the requirements applicable to this module type?

Resolution

A ***hybrid*** cryptographic module is a special type of software or firmware cryptographic module that, as part of its composition, utilizes disjoint special purpose cryptographic hardware[22] components installed within the physical boundary of the GPC or operating environment. A hybrid cryptographic module implemented as disjoint hardware and software components is defined as a Software-Hybrid. A hybrid cryptographic module implemented as disjoint hardware and firmware components is defined as Firmware-Hybrid.

In addition to the requirements applicable to a software or firmware cryptographic module, the following requirements are also applicable to the additional cryptographic hardware of the ***hybrid*** cryptographic module:

- Cryptographic Module Specification: All the components of the ***hybrid*** cryptographic module must be fully specified by type, part numbers and version numbers;
    - Manufacturer and model of the special purpose hardware component(s) and platform(s) on which testing was performed;
    - Operating system(s) on which testing was performed; and
        - *If Software-Hybrid*: modifiable operating system
        - *If Firmware-Hybrid*: the limited or non-modifiable operating system
    - All additional special purpose hardware and firmware components as applicable
- Cryptographic Module Ports and Interfaces: By policy, all status and control ports and interfaces of the hybrid cryptographic module shall be directed through the software component logical interface if a software module (controlling component), and through the firmware interface if a firmware module (controlling component);
- Roles, Services and Authentication: All the services provided by the composite of the ***hybrid*** cryptographic module must be specified;
- Physical Security: Section 5 – Physical Security is applicable for a ***hybrid*** module since a hardware component is specified as part of the hybrid composite.
- Cryptographic Key Management: Key exchanged within the boundary of the GPC or operating platform and between two or more components of the ***hybrid*** cryptographic module may be transferred in plaintext;
- Self-Tests: Self-tests requirements are applicable to all components of the ***hybrid*** cryptographic module;

[22] e.g. cryptographic hardware accelerator cards, cryptographic hardware chip(s), , etc.

  - A strong integrity test shall be performed on the software component,

  - A firmware integrity test (AS09.22) shall be performed on any applicable special purpose firmware component, and

  - All other applicable power-up or conditional tests are applicable to all components as required.

- Security Policy: The security policy must specify all the components of the ***hybrid*** cryptographic module by type, part numbers and version numbers. The security policy must contain a picture of the hardware components of the module. The security policy must specify all the services and sub-services provided by each component of the ***hybrid*** cryptographic module.

- Operational Environment: Section 6 – The operating system requirements may be applicable for a ***hybrid*** module.

  - If the module is a Software-Hybrid module; this section is applicable; or

  - If the module is a Firmware-Hybrid module; this section is not applicable.

FIPS 140-2 IG G.13 provides information guidance on how to complete the FIPS certificate for a hybrid module.

Additional Comments

Hybrid cryptographic modules shall be only applicable at FIPS 140-2 Level 1.

The hybrid cryptographic module may be ported to other compatible environments per IG G.5.

Changes to *any* component of the ***hybrid*** cryptographic module require the re-validation of the complete module as per IG G.8 – *Revalidation Requirements.*

The hardware components and applicable firmware components of the ***hybrid*** module are considered an extension of the software or firmware module to perform or accelerate cryptographic operations. In a ***hybrid*** module, the hardware components can only exchange CSPs and control information with the controlling software or firmware component of the module.

1.10 moved to A.3

1.11 moved to D.1

1.12 moved to C.1

1.13 moved to A.4

1.14 moved to A.5

1.15 moved to A.6

Section 2 – Cryptographic Module Ports and Interfaces

Section 3 – Roles, Services, and Authentication

3.1 Authorized Roles

Applicable Levels:	*All*
Original Publishing Date:	*05/29/2002*
Effective Date:	*05/29/2002*
Last Modified Date:	*06/14/2007*
Relevant Assertions:	
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Question/Problem

An operator is not required to assume an authorized role to perform services where cryptographic keys and CSPs are not modified, disclosed, or substituted (e.g., show status, self-tests, or other services that do not affect the security of the module).

Authentication mechanisms may be required within a cryptographic module to authenticate an operator accessing the module, and to verify that the operator is authorized to assume the requested role and perform the services within the role.

Resolution

An operator **shall** assume an authorized role for all services utilizing Approved security functions with the following exceptions if cryptographic keys and CSPs are not created, modified, disclosed, or substituted:

- The Secure Hash Algorithms (SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512) which are specified in *Secure Hash Standard*, FIPS 180-2 with Change Notice 1 dated February 25, 2004;
- The deterministic Random Number Generators which are specified in National Institute of Standards and Technology, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators (Revised)*, NIST Special Publication 800-90, March 2007. If the RNG service is provided to an operator who is not required to assume an authorized role, the entropy source and seeding of the RNG **shall** be completely contained within the boundary of the cryptographic module and not subject to manipulation by any operator or service of the module;
- Processes used for authentication (e.g., symmetric algorithm secret sharing, asymmetric algorithms for authentication). The completion of the authentication mechanism **shall** be enforced (e.g., the module will cease to function, even after power up) until the authentication is completed before any generalized authenticated role for any services utilizing Approved security functions is allowed; and
- Show status, self-tests, zeroization or other services that do not affect the security of the module.

Additional Comments

3.2 Bypass Capability in Routers

Applicable Levels:	ALL
Original Publishing Date:	04/01/2009
Effective Date:	04/01/2009
Last Modified Date:	04/01/2009
Relevant Assertions:	AS03.12 and AS03.13
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Background

A router is a particular type of cryptographic module where bypass is typically applicable but has some unique attributes. Typically, a router has an internal IP address table that contains entries for known addresses as well as instructions specifying routing destinations and whether the packets are to be encrypted or passed in plaintext. In addition, if an unknown IP address is found, a router may “drop” the incoming packet or pass it to a predetermined address unchanged (e.g. default gateway).

Question/Problem

Is the cryptographic module subject to the bypass requirements of FIPS 140-2 if packets with an unknown IP address are either dropped or re-directed to a predetermined address (e.g. default gateway)?

Resolution:

The bypass requirements of FIPS 140-2 are not applicable if packets with an unknown IP address are dropped unprocessed.

Packets with an unknown IP address that are re-directed to a predetermined address (e.g. default gateway) are bypassing the module’s encryption and the bypass requirements of FIPS 140-2 are applicable.

This IG is also applicable to cryptographic modules that are offering an exclusive bypass capability or no bypass capability at all.

Additional Comments

Section 4 - Finite State Model

Section 5 - Physical Security

5.1 Opacity and Probing of Cryptographic Modules with Fans, Ventilation Holes or Slits at Level 2

Applicable Levels:	*Level 2*
Original Publishing Date:	*02/10/2004*
Effective Date:	*02/10/2004*
Last Modified Date:	*02/10/2004*
Relevant Assertions:	*AS05.49*
Relevant Test Requirements:	*TE05.49.01*
Relevant Vendor Requirements:	*VE05.49.01*

Background

Cryptographic modules typically require the use of heat dissipation techniques that can include the use of fans, ventilation holes or slits. The size of these openings in the modules' enclosure, or the spacing between fan blades, may allow the viewing or possible probing of internal components and structures within the cryptographic module.

Question/Problem

How do the opacity requirements of FIPS 140-2 affect the design of the heat dissipation techniques on those cryptographic modules at Security Level 2? Should the cryptographic module prevent probing through the ventilation holes or slits at Security Level 2?

Resolution

The following are the physical security requirements for multi-chip stand-alone module at Security Level 2 pertaining to opacity and probing:

- the embodiments that are entirely contained within a metal or hard plastic production-grade enclosure that may include doors or removable covers (Security Level 1 requirement); and
- the enclosure of the cryptographic module **shall** be opaque within the visible spectrum.

Probing Requirements

Probing is not addressed at Security Level 2. Probing through ventilation holes or slits is addressed at Security Level 3 (AS.05.21).

Opacity Requirements

The purpose of the opacity requirement is to deter direct observation of the cryptographic module's internal components and design information to prevent a determination of the composition or implementation of the module.

A module is considered "opaque" only if it cannot be determined by visual inspection within the visible spectrum using artificial light sources shining through the enclosure openings or translucent surfaces, the manufacturer and/or model numbers of internal components (such as specific IC types) and/or design and composition information (such as wire traces and interconnections).

Component outlines may be visible from the enclosure openings or translucent surfaces as long as the component's manufacturer and/or model numbers, and/or composition and information about the module's design cannot be determined.

All components within the boundary of the cryptographic module must meet the opacity requirements of the standard. Excluded non-security relevant components do not have to meet these requirements.

Additional Comments

Note: Visible light is defined as light within a wavelength range of 400nm to 750nm.

5.2 Testing Tamper Evident Seals

Applicable Levels:	*Levels 2, 3 and 4*
Original Publishing Date:	*09/12/2005*
Effective Date:	*09/12/2005*
Last Modified Date:	*09/12/2005*
Relevant Assertions:	*AS.05.16, AS.05.35, AS.05.36, AS.05.37, AS.05.48, AS.05.50*
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Question/Problem

What level of testing and scope of testing should be applied when testing tamper evident seals?

Resolution

If a module uses tamper evident labels, it **shall** not be possible to remove or reapply a label without tamper evidence. For example, if the label can be removed without tamper evidence, and the same label can be re-applied without tamper evidence, the assertion fails.

Conversely, if any attempt to remove the label leaves evidence, or removal and re-application leaves evidence, or the label is destroyed during removal, the assertion passes. This means that the CST laboratory **shall** have to use creative ways (e.g. chemically, mechanically, thermally) to remove a label without evidence and without destroying the original label, and be able to re-apply the removed label in a manner that does not leave evidence.

Additional Comments

It is out-of-scope for an attacker to introduce new materials to cover up evidence of the attack.

5.3 Physical Security Assumptions

Applicable Levels:	*ALL*
Original Publishing Date:	*03/10/2009*
Effective Date:	*03/10/2009*
Last Modified Date:	*03/10/2009*
Relevant Assertions:	
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Background

Extracted from FIPS 140-2 Section 1 – OVERVIEW:

> FIPS 140-1 was developed by a government and industry working group composed of both operators and vendors. The working group identified requirements for four security levels for cryptographic modules to provide for a wide spectrum of data sensitivity (e.g., low value administrative data, million dollar funds transfers, and life protecting data) and a diversity of application environments (e.g., a guarded facility, an office, and a completely unprotected location). Four security levels are specified for each of 11 requirement areas. Each security level offers an increase in security over the preceding level. These four increasing levels of security allow cost-effective solutions that are appropriate for different degrees of data sensitivity and different application environments. FIPS 140-2 incorporates changes in applicable standards and technology since the development of FIPS 140-1 as well as changes that are based on comments received from the vendor, laboratory, and user communities.
>
> The use of a validated cryptographic module in a computer or telecommunications system is not sufficient to ensure the security of the overall system. The overall security level of a cryptographic module must be chosen to provide a level of security appropriate for the security requirements of the application *and environment* in which the module is to be utilized and for the security services that the module is to provide. The responsible authority in each organization should ensure that their computer and telecommunication systems that utilize cryptographic modules provide an acceptable level of security for the given application and environment.
>
> The importance of security awareness and of making information security a management priority should be communicated to all users. Since information security requirements vary for different applications, organizations should identify their information resources and determine the sensitivity to and the potential impact of losses. Controls should be based on the potential risks and should be selected from available controls, including administrative policies and procedures, physical and environmental controls, information and data controls, software development and acquisition controls, and backup and contingency planning.

FIPS 140-2 does not specify the required strength of the Approved security functions that may be implemented within a cryptographic module at each security level. Allowable strengths are addressed in IG 7.5. Therefore a Level 1 module may implement the same security strength of an encryption function as a Level 4 module.

The four physical security levels of FIPS 140-2 are focused on the protection of the modules CSPs by the module itself independent of the environment the module is deployed. Therefore selection of a security level is greatly influenced by the environment the module is to be deployed. At a Level 1 security level, which does not itself provide physical security protection, in the right environment, may be an acceptable solution because the environment provides the required physical security protection features.

A software cryptographic module is not subject to the physical security requirements of this standard. The following resolution assumes the host platform is not subject to the physical security requirements of FIPS 140-2.

Question/Problem

What are the assumptions that have defined the protection, attack types and operator roles in the FIPS 140-2 physical security requirements for which a cryptographic module itself provides at each security level?

Resolution:

Level 1

Protection Provided:

No physical protection of CSPs; access assumed

Hardware: probing and observation of components assumed.
Software: access to operating environment, applications and data assumed.

User Assumptions:

Correct operation of the *Approved* cryptographic services and security functions.
All attacks result in access to CSPs and data (plaintext and ciphertext) held within the module.
Operator is responsible for the physical protection of the module.

*Value or sensitivity of data protected by the module is assumed negligible in an unprotected environment.

Attack Type:

Passive attack to gain immediate access to CSPs and data held by the module.

Attack Characterization/Testing Assumptions:

No prior access to the module is assumed.
No tools and materials are assumed needed.

Value:

The module provides correct operation of security functions and services. Protection of the plaintext CSPs and data held within the module is provided by the operator of the module (e.g. the environment the module may be used). If the module is used in an unprotected environment, then the module should not hold or maintain unprotected plaintext CSPs or data.

Level 2

Protection Provided:

Observable evidence of tampering.
Physical boundary of the module is opaque to prevent direct observation of internal security components.
Hardware: probing is assumed.
Software: logical access protection of the cryptographic modules unprotected CSPs and data is provided by the evaluated operating system at EAL2.

User Assumptions:

Correct operation of the *Approved* cryptographic services and security functions.
All attacks result in access to CSPs and data (plaintext and ciphertext) held within the module.
Operator is responsible for the physical protection of the module.

*Value or sensitivity of data protected by the module is assumed low in an unprotected environment.

Attack Type:

Active attack to gain immediate access to CSPs and data held by the module.

Attack Characterization/Testing Assumptions:

No prior access to the module is assumed.
Readily available low cost tools and materials which are on hand at time of attack.
Attack time is assumed to be low.

Value:

The module provides correct operation of security functions and services. Protection of the plaintext CSPs and data held within the module is provided by the operator of the module (e.g. the environment the module may be used). The operator of the module is aware by tamper evidence that internal information may be compromised. If the module is used in an unprotected environment, then the module should not hold or maintain unprotected plain-text CSPs or data which have a moderate or high value.

Level 3

Protection Provided:

Observable evidence of tampering.
Physical boundary of the module is opaque to prevent direct observation of internal security components.
Direct entry/probing attacks prevented.
Strong tamper resistant enclosure or encapsulation material.
If applicable, active zeroization if covers or doors opened.
Software: logical access protection of the cryptographic modules unprotected CSPs and data is provided by the evaluated operating system at EAL3.

User Assumptions:

Correct operation of the *Approved* cryptographic services and security functions.
Non-direct attacks result in access to CSPs and data (plaintext and ciphertext) held within the module.

*Value of data protected by the module is assumed moderate in an unprotected environment.

Attack Type:

Moderately aggressive attack to gain immediate access to to CSPs and data held by the module.

Attack Characterization/Testing Assumptions:

Prior access to or basic knowledge of the module is assumed.
Readily available tools and materials.
Actual attack time is assumed to be moderate (this does not include time spend gaining prior access or basic knowledge of module).

Value:

The module provides correct operation of security functions and services. Protection of the plaintext CSPs and data held within the module is provided by the operator of the module (e.g. the environment the module may be used) and by the physical protection mechanisms of the module (e.g. strong enclosure, tamper response for covers and doors, deterrent of probing). The operator of the module is aware by tamper evidence that internal information may be compromised. An attack is pre-meditated but will be of moderate difficulty. If the module is used in an unprotected environment, then the module should not hold or maintain unprotected plain-text CSPs or data which have a high value.

Level 4

Protection Provided:

Observable evidence of tampering.
Physical boundary of the module is opaque to prevent direct observation of internal security components.
Direct entry/probing attacks prevented.
Strong tamper resistant enclosure or encapsulation material.
If applicable, active zeroization if covers or doors opened.
A complete envelope of protection around the module preventing unauthorized attempts at physical access.
Penetration of the module's enclosure from any direction had a very high probability of being detected resulting in immediate zeroization of plaintext CSPs or severe damage to the module rendering it inoperable.
Non-direct attacks prevented.
Software: logical access protection of the cryptographic modules unprotected CSPs and data is provided by the evaluated operating system at EAL4.

User Assumptions:

Correct operation of the *Approved* cryptographic services and security functions.
Module is tamper resistant against all physical attacks defined in the standard.

*Value of data protected by the module is assumed high in an unprotected environment.

Attack Type:

Aggressive attack to gain immediate access to to CSPs and data held by the module.

Attack Characterization/Testing Assumptions:

Prior access to or advanced knowledge of the module is assumed.
Specialized tools and materials.
Temperature and voltage attacks.
No time restriction on attack.

Value:

The module provides correct operation of security functions and services. Protection of the plaintext CSPs and data held within the module is provided by the operator of the module (e.g. the environment the module may be used) and by the physical protection mechanisms of the module (e.g. strong enclosure, tamper response for covers and doors, complete envelope of protection and penetration detection resulting in immediate zeroization of plaintext CSPs, voltage and temperature assurance). The operator of the module is aware by tamper evidence that the module was attached. The module shall zeroize all unprotected CSPs before an attacker can compromise the module. An attack is pre-meditated, well funded, organized and determined.

Additional Comments

*Discussion of the value of the data protected by the module does not consider physical protection provided by the operator to supplement the minimum physical security requirements of each level in FIPS 140-2. As an example, a user of Level 1 module may add "guards, guns, vaults and gates" surrounding the module and therefore may be comfortable in protecting more valuable information.

Attack times of low and moderate are subjective and depend on the experience and skill of an attacker and techniques employed. FIPS 140-2 Derived Test Requirements and FIPS 140-1 and FIPS 140-2 Implementation Guidance provide further guidance for the tester for each security level.

5.4 Level 3: Hard Coating Test Methods

Applicable Levels:	*Level 3*
Original Publishing Date:	*01/27/2010*
Effective Date:	
Last Modified Date:	*06/15/2010*
Relevant Assertions:	*AS05.28, AS05.39 and AS05.52*
Relevant Test Requirements:	*TE05.28.02, TE05.39.06 and TE05.52.02*
Relevant Vendor Requirements:	

Background - References

AS05.28: (Single-Chip - Levels 3 and 4) Either the cryptographic module shall be covered with a hard opaque tamper-evident coating (e.g., a hard opaque epoxy covering the passivation).

TE05.28.02: The tester shall verify that the coating cannot be easily penetrated to the depth of the underlying circuitry, and that it leaves tamper evidence. The inspection must verify that the coating completely covers the module, is visibly opaque, and deters direct observation, probing, or manipulation.

AS05.39: (Multiple-Chip Embedded - Levels 3 and 4) the multiple-chip embodiment of the circuitry within the cryptographic module shall be covered with a hard coating or potting material (e.g., a hard epoxy material) that is opaque within the visible spectrum.

TE05.39.06: (Option 1 - Utilize a hard opaque material) The tester shall verify by inspection and from vendor documentation that the module is covered with a hard opaque material. The documentation shall specify the material that is used. The tester shall verify that it cannot be easily penetrated to the depth of the underlying circuitry. The tester shall verify that the material completely covers the module and is visibly opaque within the visible spectrum.

AS05.52: (Multiple-Chip Standalone – Levels 3 and 4) the multiple-chip embodiment of the circuitry within the cryptographic module shall be covered with a hard potting material (e.g., a hard epoxy material) that is opaque within the visible spectrum.

TE05.52.02: (Option 1 – Covered with a hard opaque potting material) Encapsulate within a hard, opaque potting material. The tester shall verify from vendor documentation and by inspection, if internal access is possible, that the circuitry within the module is covered with a hard opaque potting material. The documentation shall specify which potting material is used and its hardness characteristics.

Question/Problem

What kind of testing is expected to be performed at Level 3 to verify that the hard coating or potting material that encapsulates the circuitry is *hard*?

Resolution

Within the scope of FIPS 140-2, the term *hard* is defined as:

Hard / hardness: the relative resistance of a metal or other material to denting, scratching, or bending; physically toughened; rugged, and durable. The relative resistances of the material to be penetrated by another object.

Test methods **shall** be consistent with *FIPS 140-2 Implementation Guidance* IG 5.3 that addresses a *moderately aggressive attack* at Level 3.

The test methods **shall** at a minimum address the hardness characteristics of the epoxy or potting material as follows:

1. Attempts to penetrate the material by an instrument (e.g. awl, pointed handheld tool, etc.) using a *moderately aggressive* amount of force to the depth of the underlying circuitry. The use of a drilling or grinding motion is out-of-scope.

2. The use of an instrument with a *moderately aggressive* amount of force to pry or break the material away from the underlying circuitry (e.g. insert a pry instrument at the boundary of the epoxy or potting material and another material/component (e.g. PCB board)).

3. The use of a *moderately aggressive* amount of flexing or bending force to crack or break the material away from or expose the underlying circuitry.

During testing the module should be consistently assessed to determine if serious damage has occurred (i.e. the module will either cease to function or the module is unable to function).

The manufacturing method which is used to apply the epoxy or potting material **shall** be reviewed to determine if voids or pockets may exist that could create an exposure or weakness. The above testing **shall** exploit those areas.

Module hardness testing **shall** be performed at the vendors specified nominal operating temperature for the module and at the vendors specified lowest and highest temperature that the module will not be damaged (e.g. during storage, transportation/shipping, etc.). If no specification is provided, hardness testing **shall** be performed by the laboratory at ambient temperature.

The Security Policy **shall** (AS14.05) specify the nominal and high/low temperature range that the module hardness testing was performed. If the module hardness testing was only performed at a single temperature (e.g. vendor provided only a nominal temperature or the vendor did not provide a specification), the Security Policy **shall** clearly state that the module hardness testing was only performed at a single temperature and no assurance is provided for Level 3 hardness conformance at any other temperature.

At Level 3, testing methods at all embodiments (single-chip, multi-chip embedded and multi-chip standalone) **shall not** consist of drilling, milling, cutting, burning, melting, grinding or dissolving the epoxy or potting material, in order to gain access to the underlying circuitry. These types of "attacks" are addressed by Level 4 physical security and are consistent with *FIPS 140-1 Implementation Guidance* IG 5.7.

Additional Comments

While the above test methods may be applicable at Physical Security Level 3 for a module which is protected by a strong enclosure or includes doors or removable covers, this IG does not specifically address those test methods.

Section 6 – Operational Environment

6.1 Single Operator Mode and Concurrent Operators

Applicable Levels:	*All*
Original Publishing Date:	*03/10/2003*
Effective Date:	*03/10/2003*
Last Modified Date:	*04/24/2003*
Relevant Assertions:	*AS06.04*
Relevant Test Requirements:	*TE06.04*
Relevant Vendor Requirements:	*VE06.04*

Background

Historically, for a FIPS 140-1 and FIPS 140-2 validated software cryptographic module on a server to meet the single user requirement of Security Level 1, the server had to be configured so that only *one* user at a time could access the server. This meant configuring the server Operating System (OS) so that only a single user at a time could execute processes (including cryptographic processes) on the server. Consequently, servers were not being used as intended.

Question/Problem

AS06.04 states: "(Level 1 Only) The operating system **shall** be restricted to a single operator mode of operation (i.e., concurrent operators are explicitly excluded)". What is the definition of concurrent operators in this context? Specifically, may Level 1 software modules be implemented on a server and achieve FIPS 140-2 validation? (Note: this question is also applicable to VPN, firewalls, etc.)

Resolution

Software cryptographic modules implemented in client/server architecture are intended to be used on both the client and the server. The cryptographic module will be used to provide cryptographic functions to the client and server applications. When a crypto module is implemented in a server environment, the server application is the user of the cryptographic module. The server application makes the calls to the cryptographic module. Therefore, the server application is the single user of the cryptographic module, even when the server application is serving multiple clients

Additional Comments

This information must be included in the non-proprietary security policy.

6.2 Applicability of Operational Environment Requirements to JAVA Smart Cards

Applicable Levels:	*All*
Original Publishing Date:	*04/08/2003*
Effective Date:	*04/08/2003*
Last Modified Date:	*09/11/2003*
Relevant Assertions:	*AS06.01*

Relevant Test Requirements:	
Relevant Vendor Requirements:	

Background

FIPS 140-2 states (Section 4.6 Operational Environment) “A limited operational environment refers to a static non-modifiable virtual environment (e.g., a JAVA virtual machine on a non-programmable PC card) with no underlying general purpose operating system upon which the operational environment uniquely resides.”

Question

Does the FIPS 140-2 statement mean that a smart card implementing a non-modifiable operating system (e.g., like the ones currently used today in most smart cards) that accept and run JAVA applets (whether validated or not) is a limited operational environment?

Resolution

The CMVP cannot issue a general statement that applies to all JAVA card modules since functionality and design can vary greatly from module to module. The determination is left to the CST laboratories, which have the complete module documentation available to them. In general**,** however, a JAVA smart card module with the ability to load unvalidated applets post-validation is considered to have a *modifiable* operational environment and the Operational Environment requirements of FIPS 140-2 are applicable.

A JAVA smart card module having a modifiable operational environment which either:

a) is configured such that the loading of any applets is not possible, or

b) loads only applets that have been tested and validated to either FIPS 140-1 or FIPS 140-2,

could be considered to have a *limited* operational environment and have the FIPS 140-2 Operational Environment requirements section of the module test report marked as *Not Applicable*.

The validated JAVA smart card cryptographic module must use an Approved authentication technique on all loaded applets. The module **shall** also meet, at a minimum, the requirements of AS09.34, AS09.35, AS10.03 and AS10.04, as well as any other applicable assertions. Validation of the cryptographic module is maintained through the loading of applets that have either been tested and validated during the validation effort of the smart card itself or through an independent validation effort (i.e., the applet itself has its own validation certificate number).

The security policy of the validated smart card module must state whether:

- The module can load applets post-validation, validated or not (Note: if the module can load non-validated applets post-validation, the security policy must clearly indicate that the module’s validation to FIPS 140-1 or FIPS 140-2 is no longer valid once a non-validated applet is loaded);

- Any applets are contained within the validated cryptographic module and, if so, must list their name(s) and version number(s).

Additional Comments

The name(s) and version number(s) of all applets contained within a validated cryptographic module **shall** be listed on the module’s certificate and CMVP website entry.

6.3 Correction to Common Criteria Requirements on Operating System

Applicable Levels:	*All*
Original Publishing Date:	*03/29/2004*
Effective Date:	*03/29/2004*
Last Modified Date:	*03/29/2004*
Relevant Assertions:	*AS06.10, AS06.21 and AS06.27*
Relevant Test Requirements:	*TE06.10, TE06.21 and TE06.27*
Relevant Vendor Requirements:	*VE06.10, VE06.21 and VE06.27*

Background

Depending on how assertions AS.06.10, AS.06.21 and AS.06.27 are read, they could be interpreted as the OS upon which the module is running on has to meet ALL of the listed PPs in Annex B at EAL2, EAL3 and EAL4 respectively. This is because of the plural at the end of the "Protection Profile**s**".

Question/Problem

Must the OS upon which the module is running on has to meet ALL of the listed PPs in Annex B at EAL2, EAL3 and EAL4 respectively?

Resolution

No, the requirements should be interpreted to read as follows:

- For **AS.06.10**:

  an operating system that meets the functional requirements specified in **a** Protection Profile listed in Annex B and is evaluated at the CC evaluation assurance level EAL2

- For **AS.06.21**, the first sentence:

  an operating system that meets the functional requirements specified in **a** Protection Profile listed in Annex B.

- For **AS.06.27**, the first sentence:

  an operating system that meets the functional requirements specified in **a** Protection Profile listed in Annex B.

Additional Comments

6.4 Approved Integrity Techniques

Applicable Levels:	*All*
Original Publishing Date:	*01/21/2005*
Effective Date:	*01/21/2005*
Last Modified Date:	*01/21/2005*
Relevant Assertions:	*AS06.08*
Relevant Test Requirements:	*TE06.01.01-02*

Relevant Vendor Requirements:	*VE06.08.01*

Background

Section 4.6.1 of FIPS 140-2 states that "A cryptographic mechanism using an Approved integrity technique (e.g. Approved message authentication code or digital signature algorithm) **shall** be applied to all cryptographic software and firmware components within the cryptographic module."

Question/Problem

What is an *Approved integrity technique*, as specified in AS06.08, and when must be it performed?

Resolution

An *Approved integrity technique* is a keyed cryptographic mechanism that uses an Approved and validated cryptographic security function. This includes a digital signature scheme, an HMAC or a MAC. Approved security functions are listed in FIPS 140-2 Annex A.

The Approved integrity technique is considered a *Power-Up Test* and **shall** meet all power-up test requirements.

Additional Comments

Section 7 – Cryptographic Key Management

7.1 moved to D.2

7.2 Use of IEEE 802.11i Key Derivation Protocols

Applicable Levels:	*All*
Original Publishing Date:	*01/21/2005*
Effective Date:	*01/21/2005*
Expiration Date:	
Last Modified Date:	*01/27/2010*
Relevant Assertions:	*AS07.17*
Relevant Test Requirements:	*TE07.17.01-02*
Relevant Vendor Requirements:	*VE07.17.01*

Background

FIPS 140-2 Annex D provides a list of the FIPS Approved key establishment techniques applicable to FIPS PUB 140-2.

The commercially available schemes referred to in FIPS 140-2 Annex D are concerned with the derivation of a shared secret, or, as it is sometimes called, "the keying material." The IEEE 802.11i standard describes how to derive keys from a secret shared between two parties. It does not specify how to establish this commonly shared secret.

Question/Problem

Assuming that the shared secret is established using a key establishment technique specified in Annex D, can a cryptographic module use the 802.11i key derivation techniques to derive a data protection key, a key encryption key and other keys for use in a FIPS Approved mode of operation?

Resolution

Implementations of the IEEE 802.11i protocol operating in a FIPS approved mode of operation must meet the following requirements:

1. To derive a data protection key, a key encryption key and other keys for use in a FIPS Approved mode of operation, the following requirements **shall** be met:

   a) the shared secret (the keying material) **shall** be established using a FIPS Approved method specified in FIPS 140-2 Annex D; AND

   b) the key derivation function **shall** be implemented as defined **IG 7.10**.

2. The data protection method defined in the 802.11i protocol **shall** be AES CCM, which is an Approved security function for use in a FIPS Approved mode of operation as specified in FIPS 140-2 Annex A.

3. The keying material may be established via manual methods as specified in FIPS 140-2. The key derivation function as defined in **IG 7.10** may then be applied.

Additional Comments

References

Amendment 6: IEEE 802.11Medium Access Control (MAC) Security Enhancements, IEEE P802.11i/D10.0, April 2004. Section 8.5.1.2. Pairwise Key Hierarchy.

7.3 moved to C.2

7.4 Zeroization of Power-Up Test Keys

Applicable Levels:	*All*
Original Publishing Date:	*09/12/2005*
Effective Date:	*09/12/2005*
Last Modified Date:	*02/23/2007*
Relevant Assertions:	*AS07.41*
Relevant Test Requirements:	*TE07.41.01-04*
Relevant Vendor Requirements:	*VE07.41.01*

Background

Section 4.7.6 of FIPS 140-2 states that "*The cryptographic module **shall** provide methods to zeroize all Plaintext secret and private cryptographic keys and CSPs within the module.*"

Question/Problem

Are cryptographic keys used by a module ONLY to perform Section 4.9.1 Power-Up Tests (e.g. cryptographic algorithm Known Answer Tests (KAT) or software/firmware integrity tests) considered CSPs and is zeroization required under Section 4.7.6?

Resolution

Cryptographic keys used by a cryptographic module ONLY to perform Section 4.9.1 Power-Up Tests are not considered CSPs and therefore do not need to meet the Section 4.7.6 zeroization requirements.

Additional Comments

7.5 Strength of Key Establishment Methods

Applicable Levels:	*All*
Original Publishing Date:	*11/23/2005*

Effective Date:	*06/29/2005*
Last Modified Date:	*06/10/2010*
Relevant Assertions:	*AS07.19*
Relevant Test Requirements:	*TE07.19.01-02*
Relevant Vendor Requirements:	*VE07.19.01*

Background

NOTE: NIST SP 800-131, *Recommendation for the Transitioning of Cryptographic Algorithms and Key Sizes*, when published, will address new transition dates and migration from legacy algorithms. The latest draft can be found at: http://csrc.nist.gov/publications/PubsDrafts.html Once NIST SP 800-131 is published, the CMVP will provide updated transition guidance related to module validations and review.

FIPS 140-2 AS.07.19 states that "Compromising the security of the key establishment method (e.g., compromising the security of the algorithm used for key establishment) **shall** require as many operations as determining the value of the cryptographic key being transported or agreed upon. "

NIST Special Publication 800-57, *Recommendation for Key Management – Part 1: General (Revised)* (March 2007), Section 5, Sub-Section 5.6.1, Comparable Algorithm Strength, contains Table 2, which provides comparable security strengths for the Approved algorithms.

Table 2: Comparable strengths

Bits of security	**Symmetric key algorithms**	**FFC (e.g., DSA, D-H)**	**IFC (e.g., RSA)**	**ECC (e.g., ECDSA)**
80	2TDEA[18]	L = 1024 N = 160	k = 1024	f = 160-223
112	3TDEA	L = 2048 N = 224	k = 2048	f = 224-255
128	AES-128	L = 3072 N = 256	k = 3072	f = 256-383
192	AES-192	L = 7680 N = 384	k = 7680	f = 384-511
256	AES-256	L = 15,360 N = 512	k = 15,360	f = 512+

[18] The 80-bit security of 2TDEA is based on the availability of 2^{40} matched plaintext and ciphertext blocks to an attacker (see [ANSX9.52], Annex B).

1. Column 1 indicates the number of bits of security provided by the algorithms and key sizes in a particular row. Note that the bits of security is not necessarily the same as the key sizes for the algorithms in the other columns, due to attacks on those algorithms that provide computational advantages.
2. Column 2 identifies the symmetric key algorithms that provide the indicated level of security (at a minimum), where 2TDEA and 3TDEA are specified in [SP800-67], and AES is specified in [FIPS197]. 2TDEA is TDEA with two different keys; 3TDEA is TDEA with three different keys.
3. Column 3 indicates the minimum size of the parameters associated with the standards that use finite field cryptography (FFC). Examples of such algorithms include DSA as defined in [FIPS186-3] for digital signatures, and Diffie-Hellman (DH) and MQV key agreement as defined in [ANSX9.42] and [SP800-56]), where L is the size of the public key, and N is the size of the private key.
4. Column 4 indicates the value for k (the size of the modulus n) for algorithms based on integer factorization cryptography (IFC). The predominant algorithm of this type is the RSA algorithm. RSA is specified in [ANSX9.31] and [PKCS#1]. These specifications are referenced in [FIPS186-3] for digital signatures. The value of k is commonly considered to be the key size.

5. Column 5 indicates the range of f (the size of n, where n is the order of the base point G) for algorithms based on elliptic curve cryptography (ECC) that are specified for digital signatures in [ANSX9.62] and adopted in [FIPS186-3], and for key establishment as specified in [ANSX9.63] and [SP800-56]. The value of f is commonly considered to be the key size.

For example, if a 256-bit AES is to be transported utilizing RSA, then k=15,360 for the RSA key pair. A 256-bit AES key transport key could be used to wrap a 256-bit AES key.

For key strengths not listed in Table 2 above, the correspondence between the length of an RSA or a Diffie-Hellman key and the length of a symmetric key of an identical strength can be computed as:

> If the length of an RSA key L (this is the value of k in the fourth column of Table 2 above), then the length x of a symmetric key of approximately the same strength can be computed as:
>
> $$x = \frac{1.923 \times \sqrt[3]{L \times \ln(2)} \times \sqrt[3]{[\ln\,(L \times \ln(2))]^2} - 4.69}{\ln(2)} \qquad (1)$$
>
> If the lengths of the Diffie-Hellman public and private keys are L and N, correspondingly, then the length y of a symmetric key of approximately the same strength can be computed as:
>
> $$y = \min(x, N/2)\,, \qquad (2)$$
>
> where x is computed as in formula (1) above.

Question/Problem

What does FIPS 140-2 assertion AS.07.19 mean in the context of NIST Special Publication 800-57?

Resolution

The requirement applies to the key establishment methods found in Section 4.7.

If a key is established via a key agreement or key transport method, the transport key or key agreement method **shall** be of equal or greater strength than the key being transported or established. For example, it is acceptable to have a two-key Triple-DES key (80-bit strength) transported using a 2048-bit RSA key (112-bit strength).

If the apparent strength of the largest key (taken at face value) that can be established by a cryptographic module is greater or equal than the largest comparable strength of the implemented key establishment method, then the module certificate and security policy will be annotated with, in addition to the other required caveats, the caveat "(Key establishment methodology provides xx bits of encryption strength)" for that key establishment method. For example, if a 256-bit AES is to be transported utilizing RSA with a value of k=1024 for the RSA key pair, the caveat would state "RSA (PKCS#1, key wrapping, key establishment methodology provides 80 bits of encryption strength)".

Furthermore, if the module supports, for a particular key establishment method, several key strengths, then the caveat will state either the choice of strengths provided by the keys while operated in FIPS mode, if there are only two possible effective strengths, or a range of strengths if there are more than two possible strengths. For example, if a module implements 512 and 1024-bit public key Diffie-Hellman with the private keys of 112 and 160 bits then the caveat would state "Diffie-Hellman (key agreement; key establishment methodology provides 56 or 80 bits of encryption strength)". If, on the other hand, a module implements, in support of a key wrapping protocol, the RSA encryption/decryption with the RSA keys of 1024, 2048, 4096 and 15360 bits, then the caveat would say "RSA (key wrapping; key establishment methodology provides between 80 and 256 bits of encryption strength)". These caveats provide clarification to Federal users on the actual strength the module is providing even though Table 4 below states that the strength is sufficient.

Additional Comments

NIST Special Publication 800-57, *Recommendation for Key Management – Part 1: General (Revised)* (March 2007) also provides the following information in Section 5.6.2:

Table 4 provides recommendations that may be used to select an appropriate suite of algorithms and key sizes for Federal Government unclassified applications. A minimum of eighty bits of security **shall** be provided until 2010. Between 2011 and 2030, a minimum of 112 bits of security **shall** be provided. Thereafter, at least 128 bits of security **shall** be provided.

1. Column 1 indicates the estimated time periods during which data protected by specific cryptographic algorithms remains secure. (i.e., the algorithm security lifetimes).
2. Column 2 identifies appropriate symmetric key algorithms and key sizes: 2TDEA and 3TDEA are specified in [SP800-67], the AES algorithm is specified in [FIPS197], and the computation of Message Authentication Codes (MACs) using block ciphers is specified in [SP800-38].
3. Column 3 indicates the minimum size of the parameters associated with FFC, such as DSA as defined in [FIPS186-3].
4. Column 4 indicates the minimum size of the modulus for IFC, such as the RSA algorithm specified in [ANSX9.31] and [PKCS#1] and adopted in [FIPS186-3] for digital signatures.
5. Column 5 indicates the value of *f* (the size of *n*, where *n* is the order of the base point *G*) for algorithms based on elliptic curve cryptography (ECC) that are specified for digital signatures in [ANSX9.62] and adopted in [FIPS186-3], and for key establishment as specified in [ANSX9.63] and [SP800-56]. The value of *f* is commonly considered to be the key size.

Table 4: Recommended algorithms and minimum key sizes

Algorithm security lifetimes	Symmetric key Algorithms (Encryption & MAC)	FFC (e.g., DSA, D-H)	IFC (e.g., RSA)	ECC (e.g., ECDSA)
Through 2010 (min. of 80 bits of strength)	2TDEA[21] 3TDEA AES-128 AES-192 AES-256	Min.: $L = 1024$; $N = 160$	Min.: k=1024	Min.: f=160
Through 2030 (min. of 112 bits of strength)	3TDEA AES-128 AES-192 AES-256	Min.: $L = 2048$ $N = 224$	Min.: k=2048	Min.: f=224
Beyond 2030 (min. of 128 bits of strength)	AES-128 AES-192 AES-256	Min.: $L = 3072$ $N = 256$	Min.: k=3072	Min.: f=256

[21] The 80-bit security of 2TDEA is based on the availability of 2^{40} matched plaintext and ciphertext blocks to an attacker (see [ANSX9.52], Annex B).

The algorithms and key sizes in the table are considered appropriate for the protection of data during the given time periods. Algorithms or key sizes not indicated for a given range of years **shall** not be used to protect information during that time period. If the security life of information extends beyond one time period specified in the table into the next time period (the later time period), the algorithms and key sizes specified for the later time **shall** be used. The following examples are provided to clarify the use of the table:

a. If information is encrypted in 2005 and the maximum expected security life of that data is only five years, any of the algorithms or key sizes in the table may be used. But if the information is protected in 2005 and the expected security life of the data is six years, then 2TDEA would not be appropriate.

b. If a CA signature key and all certificates issued under that key will expire in 2005, then the signature and hash algorithm used to sign the certificate needs to be secure for at least five years. A certificate issued in 2005 using 1024 bit DSA and SHA-1 would be acceptable.
c. If information is initially signed in 2009 and needs to remain secure for a maximum of ten years (i.e., from 2009 to 2019), a 1024 bit RSA key would not provide sufficient protection between 2011 and 2019 and, therefore, it is not recommended that 1024-bit RSA be used in this case. It is recommended that the algorithms and key sizes in the "Through 2030" row (e.g., 2048-bit RSA) should be used to provide the cryptographic protection. In addition, the signature must be generated using a hash algorithm of comparable or greater strength, such as SHA-224 or SHA-256.

7.6 RNGs: Seeds, Seed Keys and Date/Time Vectors

Applicable Levels:	*All*
Original Publishing Date:	*11/16/2007*
Effective Date:	*11/16/2007*
Last Modified Date:	*11/16/2007*
Relevant Assertions:	*AS07.09*
Relevant Test Requirements:	*TE07.09.01*
Relevant Vendor Requirements:	*VE07.09.01*

Background

An RNG may employ a seed and seed key and a Date/Time vector for its operation. FIPS 140-1 IG **8.7** provides a basis for the requirements related to the ANSI X9.31 RNG **seed**, **seed key** and Date/Time vector. The document titled *NIST Recommended Random Number Generator based on ANSI X9.31 Appendix A.2.4 using the 3-Key Triple DES and AES Algorithms* allows for the use of Triple-DES and AES.

Questions/Problems

1. In the case where an RNG employs a **seed** and **seed key**, how does AS07.09 apply?

2. In the case where an RNG employs a Date/Time vector, what, if any, additional attributes apply?

Resolution

1. **AS.07.09** of FIPS 140-2 specifies that the seed and seed key **shall** not have the same value.

   During initialization of the **seed** or **seed key**, the initialization data provided for one, **shall** not be provided as initialization data to the other. The **seed** or **seed key** or both may be re-initialized prior to each call for a random data value.

2. The Date/Time vector **shall** be updated on each iteration or call to the RNG. In lieu of a Date/Time vector, an incrementer may be used. The Date/Time vector or incrementer **shall** be a non-repeating value during each instance of the module's power-on state.

Additional Comments

ANSI X9.31 specifies that the **seed shall** also be kept secret. As such, the **seed** is considered a CSP and **shall** meet all the requirements pertaining to CSPs.

FIPS 140-2 AS07.14 and AS07.23 are applicable to the **seed key**.

The seed key is sometimes referred as the RNG key; the key used by the underlining encryption algorithm(s) implemented by the RNG.

7.7 Key Establishment and Key Entry and Output

Applicable Levels:	*All*
Original Publishing Date:	*01/24/2008*
Effective Date:	*01/24/2008*
Last Modified Date:	*01/24/2008*
Relevant Assertions:	*General*
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Question/Problem

Given different configurations of cryptographic modules, how can a modules key establishment and key entry and output states be easily mapped to the FIPS 140-2 requirements for Cryptographic Module Ports and Interfaces (Section 4.2), Key Establishment (Section 4.7.3) and Key Entry and Output (Section 4.7.4)?

Resolution

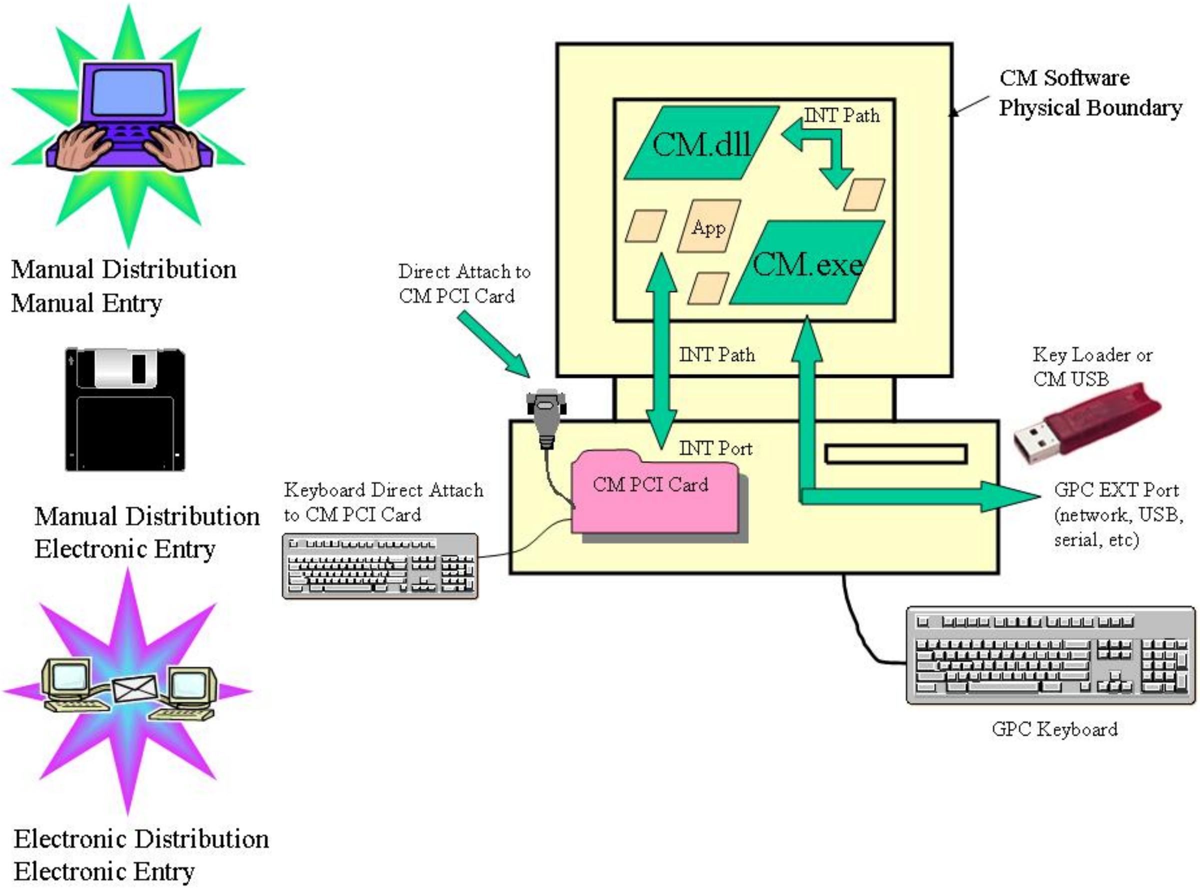
Using the following guidelines, first determine how keys are established to a module. Once the establishment method is determined, the Key Entry format table will indicate the requirements on how keys **shall** be entered or output. The following is based on the requirements found in FIPS 140-2 in Sections 4.2 and 4.7.

CM: a FIPS 140-1 or FIPS 140-2 *validated* Cryptographic Module
GPC: General Purpose Computer
EXT: a *validated* Cryptographic Module which lies *External* or outside of the boundary in regard to the reference diagrams CM software physical boundary. This also includes a standalone CM.
INT: a *validated* Cryptographic Module which lies *Internal* or inside of the boundary in regard to the reference diagrams CM software physical boundary.
App: a non-validated non-crypto general purpose software application operating inside of the boundary in regard to the reference diagrams CM software physical boundary.

Key Establishment – Table 1	
MD: Manual Distribution	**ME: Manual Entry (Input / Output)**
ED: Electronic Distribution	**EE: Electronic Entry (Input / Output)**
CM Software[1] from GPC Keyboard	**MD / ME**
CM Software[1] to/from GPC Key Loader (e.g., diskette, USB token, etc)	**MD / EE**
CM Software[1] to/from GPC EXT Ports (e.g., network port)	**ED / EE**
CM Software[1] to/from CM Software[1] via GPC INT Path	NA
CM Software[1] to/from App Software via GPC INT Path	**NA**
CM Software[1] to/from INT CM Hardware via GPC INT Path	**NA**
CM Software[1] to/from EXT CM Hardware running on a non-networked GPC (key loader)	**MD / EE**
CM Software[1] to/from EXT CM Hardware running on a networked GPC	**ED / EE**
INT CM Hardware to/from App Software via GPC INT Path	**ED / EE**

INT CM Hardware to/from GPC EXT Ports via GPC INT Path	**ED / EE**
INT CM Hardware from GPC Keyboard via GPC INT Path	**ED / EE**
INT CM Hardware to/from direct attach key loader	**MD / EE**
INT CM Hardware from direct attach keyboard	**MD / ME**
EXT CM Hardware to/from networked GPC	**ED / EE**
EXT CM Hardware to/from directly attached key loader (a non-networked GPC could be considered and used as a key loader)	**MD / EE**
EXT CM Hardware from direct attach keyboard	**MD / ME**
[1] **Must meet requirements of AS.06.04, AS.06.05 and AS.06.06**	

The following illustration provides reference to the above Key Establishment table.



Key Entry Format – Table 2

		Distribution (Establishment)				
		Manual				**Electronic**
Entry (Input / Output)	**Manual**	Keyboard, Thumbwheel, Switch, Dial				
		1	**2**	**3**	**4**	

		P/E	P/E	E/SK	E/SK				
	Electronic	Smart Cards, Token, Diskettes and Key Loaders				Key Transport or Key Agreement			
		1	**2**	**3**	**4**	**1**	**2**	**3**	**4**
		P/E	P/E	E/SK	E/SK	E	E	E	E

Legend:
P/E: May be Plaintext or Encrypted
E: Encrypted
E/SK: Encrypted or Plaintext Split Knowledge (via separated physical ports or via trusted path)

At Levels 3 and 4, plaintext key components may be entered either via separate physical ports or logically separated ports using a trusted path. Manual entry of plaintext keys must be entered using split knowledge procedures. Keys may also be entered encrypted manually. If automated methods, they must be encrypted.

Additional Comments

This IG reaffirms that keys established using *manual transport methods* and *electronically input or output* to a cryptographic module may be input or output in plaintext at Levels 1 and 2.

Level 1 Software – General Purpose Operational Environment

AS06.04: (Level 1 Only) The operating system shall be restricted to a single operator mode of operation (i.e., concurrent operators are explicitly excluded).

AS06.05: (Level 1 Only) The cryptographic module shall prevent access by other processes to plaintext private and secret keys, CSPs, and intermediate key generation values during the time the cryptographic module is executing/operational. Processes that are spawned by the cryptographic module are owned by the module and are not owned by external processes/operators.

AS06.06: (Leve1 1 Only) Non-cryptographic processes shall not interrupt the cryptographic module during execution.

A Software Cryptographic Module (SCM) requires the use of an underlying General Purpose Computer (GPC) and Operational Environment (OE) to execute/operate. A SCM is conceptually comprised of two sub-elements: a Physical Cryptographic Module (PCM) and the Logical Cryptographic Module (LCM) boundary. The LCM is executes/operates within the PCM. The LCM is the collection of executable code that encompasses the cryptographic functionality of the SCM (e.g., .dll’s, .exe’s). Other general-purpose application software (App) (e.g., word processors, network interfaces, etc) may reside within the PCM. Therefore the PCM encompasses the following elements: GPC, OE, LCM and App. The LCM relies on the OE and GPC for memory management, access to ports and interfaces, and other services such as the requirements of AS06.04, AS06.05 and AS06.06. The LCM has no operational control over other App elements within the PCM of the SCM. The SCM, which is comprised of all the various sub-elements (GPC, OE, LCM and App), is restricted to a single operator mode of operation, such that the single operator has a level of confidence in the SCM environment as a whole. The CMVP views the non-LCM elements (GPC, OE and App) as implicitly excluded.

Example: If the LCM generates keys, it must use a FIPS Approved RNG. That key may be stored within the PCM but must meet **AS06.05** unless the LCM wishes the key to be exported. If exported, refer to Table 1 for the key establishment and key entry requirements. If a key is generated outside of the LCM, then the generation method is out-of-scope but the key must be imported per Table 1 requirements.

It is the burden of the operator of the SCM to understand the environment the SCM is running. If that environment is not acceptable, then there are alternative solutions (hardware cryptographic modules and/or Level 2, 3 or 4 software cryptographic modules) that should be considered.

If the operating system requirements of AS06.04, AS06.05 and AS06.06 cannot be met, then the SCM cannot be validated at Level 1. The vendor provided documentation shall indicate how these requirements are met (AS14.02).

7.8 Key Generation Methods Allowed in FIPS Mode

Applicable Levels:	*All*
Original Publishing Date:	*03/10/2009*
Effective Date:	*03/10/2009*
Last Modified Date:	*03/10/2009*
Relevant Assertions:	*AS07.11 and AS07.16*
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Background

Section 4.7.2 of FIPS 140-2 states that "… *Approved key generation methods are listed in Annex C to this standard. If an Approved key generation method requires input from an RNG, then an Approved RNG that meets the requirements specified in Section 4.7.1* ***shall*** *be used.*"

Question/Problem

FIPS 140-2 Annex C, like all other Annexes to FIPS 140-2, exists in a draft form to allow updating as necessary. While the quote from FIPS 140-2 states that the Approved key generation methods are listed in Annex C, the annex itself lists the approved Random Number Generators (RNGs) and not the methods to derive a key from the generated random bits. How can this be reconciled and what additional processing may be applied within the cryptographic boundary of the module to the output of an RNG before this output becomes a cryptographic key?

Resolution

FIPS 140-2 Annex C is concerned with approved RNGs. Key generation is addressed in this Implementation Guidance (IG).

The term "key generation" applies to the generation of secret and private keys to be used by the cryptographic algorithms. Many algorithms that are either approved or allowed in the FIPS-approved Mode of Operations, such as AES, Triple-DES, Skipjack, DSA, ECDSA, Diffie-Hellman (DH) and the Elliptic Curve Diffie-Hellman (ECDH) (including their MQV versions, MQV and ECMQV) require a secret or private key. The same is true for the RSA Signature and the RSA key wrapping algorithms, but the generation of keys is more complicated and will be defined in the FIPS 186-3 standard; therefore, the generation of RSA keys will not be discussed in this Implementation Guidance. However, the prime generation seeds that will be required by FIPS 186-3 standard (when this standard becomes effective) to produce the secret primes *p* and *q* for RSA **shall** be generated according to this IG.

“Key generation” should not be confused with “key establishment”, which is discussed in IG 7.1. It is also different from using a key that has been entered into the module. Key generation refers to the generation of a cryptographic key or key pair “locally” within a module; the secret key of a symmetric algorithm or the public key of an asymmetric (public key) algorithm may subsequently be distributed to other parties, as appropriate. Key generation involves generating a random and/or unpredictable bit string and performing various operations that will turn it into the secret key or private key. While these operations could use certain values sent to the cryptographic module by another entity, the generating module is solely responsible for the key generation process, and, once generated, no other module knows the value of the key. (The module may later wrap this newly generated key and send it to another cryptographic module. This is outside the scope of this IG.)

To summarize, this Implementation Guidance is only concerned with the generation of a secret value K that will be used as 1) a secret key for a symmetric algorithm, such as AES or Triple DES, 2) a private key for an asymmetric (public key) algorithm, such as DSA, ECDSA, DH, ECDH, MQV or ECMQV, or 3) a prime generation seed for RSA. The secret value is sufficiently random for its use, although it may not be the direct output from a random number generator (see below).

To be used in FIPS mode, a secret value K can be any value of the form:

$$K = U \text{ XOR } V, \qquad (1)$$

where the components U and V are of the same length as K[23], are “independent” of each other, and U is derived, possibly using a *qualified post-processing* (see below), from the output of an approved RNG in the module that is generating K. In addition, each component may be a function of other values (e.g., U = F(U′), or V = F(V′)).

The security strength of the generated value K is equal to the larger of the security strengths of U and V. In general, the security strength of K is determined by the security strength of U, and the security strength of U is the minimum of the length of U (and K) and the security strength of the RNG used to generate U. Therefore, the length of U (and K), and the security strength of the RNG used to generate U **shall** meet or exceed the security strength required for K. However, a vendor can claim that the security strength of the generated value K is determined by the security strength of V if it can be demonstrated that V has a higher security strength than U.

The independence required for U and V is interpreted in the statistical sense; that is, knowing one of the values yields no information that can be used to derive the other one. The following are some examples of independent values. Note that the U component is determined by an approved RNG in all of these examples.

1. U is an output of an approved RNG within this module, and V is a constant. (Note, that if V is a string of binary zeroes, then K = U, i.e., the output of an approved RNG.)

2. U is an output of an approved RNG within this module, and V is produced by an approved or allowed key agreement scheme between this module and another module. Any seed used to instantiate an RNG in one module **shall** not intentionally be the same as the seed used in the other module. If the seeds are allowed to be the same, then a situation could occur, repeatedly, when the value of U is equal to that of V, and K would be equal to 0, each time.

3. U is an output of an approved RNG within this module, and V is a key wrapped (i.e, encrypted) by another module using an approved or allowed key wrapping algorithm, and received and unwrapped by this module.

[23] If U and V are of different length, one can be padded with a string of 0’s of the appropriate length to make them equal in length so that the XOR operation becomes meaningful.

4. U is an output of an approved RNG within this module. V′ is either 1) a constant, 2) a value produced by an approved key agreement scheme between this module and another module, or 3) a key wrapped by another module using an approved or allowed key wrapping algorithm, and is received and unwrapped by this module. V is produced by hashing V′ using an approved hash function (i.e., V = H(V′)).

5. U′ is an output of an approved RNG within this module. V is either 1) a constant, 2) a value produced by an approved key agreement scheme between this module and another module, or 3) a key wrapped by another module using an approved or allowed key wrapping algorithm and received and unwrapped by this module. U is produced by hashing U′ using an approved hash function (i.e., U = H(U′)). Note that in this case, the length of U is the length of the output of the hash function, and the security strength of U is the minimum of the security strength of U′ and the length of the output of the hash function.

6. U′ is either 1) an output of an approved RNG within this module, or 2) the output of a hash function as specified in example 5 (i.e., U′ = H(U″)). V′ is either 1) a constant, 2) a value produced by an approved key agreement scheme between this module and another module, 3) a key wrapped by another module using an approved or allowed key wrapping algorithm and received and unwrapped by this module, or 4) the output of a hash function as specified in example 4 (i.e., V′ = H(V″)). However, both U′ and V′ **shall** not be the output of a hash function (i.e., the case where U = H(U″) and V = H(V″) is not allowed).

   Either U′ or V′ or both of these values are truncated to produce the corresponding U or V value (i.e., U = U′ and V = T(V′); or U = T(U′) and V = V′; or U = T(U′) and V = T(V′)). The truncation may be performed either by dropping a certain number of the leftmost bits or a certain number of the rightmost bits from the bit strings that represents U′ or V′. Dropping bits on both sides or dropping any bits "in the middle" of the U′ or V′ strings is not permitted. The security strength of a truncated U′ value **shall** meet or exceed the security strength requirement for K. If the length of U′ is *n* bits, and it is truncated to *k* bits, the resulting security strength for U (the truncated U′ value) is the (original security strength of U′)*(*k*/*n*).

   NOTE. The security strength of U may, in some rare cases, be higher, than what is calculated at the end of Example 6. However, if a vendor wants to claim a higher security strength for U, it is their responsibility to provide to the Security Technology Group at NIST the proof of their claim.

Finally, if K1 and K2 are two keys produced by formula (1) above, the module may derive a cryptographic key by concatenating K1 and K2:

$$K = K1 \parallel K2. \qquad (2)$$

If K1 and K2 are calculated independently, then the security strength of K can be claimed to be the sum of the entropies of K1 and K2.

Qualified Post-Processing Algorithms

The U component described above uses the output of an approved RNG as an input parameter. As explained earlier, this RNG output may be further modified by applying a qualified post-processing algorithm *before* it is used to compute the secret value K. When post-processing is performed on RNG output, the output of the post-processing operation **shall** be used in place of any use of the RNG output.

Let *M* be the length of the output requested from the RNG by a consuming application, and let R_M be the set of all bit strings of length *M*. When the output is to be used for keys, *M* is typically a multiple of 64; however, these algorithms are flexible enough to cover any output size. Let R_N be the set of all bit strings of length *N*, and let F: $R_N \rightarrow \{0,1, \ldots, k\text{-}1\}$ be a function on *N*-bit strings with integer output in the range 1 to *k*, where *k* is an arbitrary positive integer. Let $\{P_1, P_2, \ldots, P_k\}$ be a set of permutations (one-to-one functions) from R_M back to

R_M. The P_j's may be fixed, or they may be generated using a random seed or secret value. Examples of F and P_i are given below.

Let r_1 be randomly selected from the set R_N (i.e., r_1 is a random N-bit value), and let r_2 be randomly selected from the set R_M (i.e., r_2 is a random M-bit value). Both r_1 and r_2 **shall** be outputs from an approved RNG, such that $N \le M$. (The case $r_1 = r_2$ is permissible.) The post processor's output is the M-bit string $P_{\mathrm{F}(r_1)}(r_2)$.

NOTE. Although some security strength is lost during post-processing, the loss is small enough to be ignored for the purposes of FIPS 140-2 validation.

The apparent complexity of this post-processing should not be of any concern to vendors and testing laboratories. The identical permutation (that is, no post-processing at all) is perfectly acceptable.

Examples of F(r_1) used for Post Processing

The function F may be simple or fairly complex.

Let k be the number of desired permutations, and let r_1 represent an N-bit output of an approved RNG. Two examples are provided:

1. A very simple example of a suitable F is the following, where k is assumed to be an integer in the range 1 to 2^N:

   $$\mathrm{F}(r_1) = r_1 \bmod k.$$

   Here, r_1 is interpreted as an integer represented by the bit string r_1.

2. A more complex example is:

   $$\mathrm{F}(r_1) = \mathrm{HMAC}(key,\ r_1) \bmod k\ ,$$

   using a hashing algorithm and a fixed key in the HMAC computation. In this case, k could be as large as 2^{outlen}, or as small as 1, where *outlen* is the length of the hash function output in bits. (Having a single permutation, while permitted, would certainly not require the use of a keyed hash to "choose" it. On the other hand $k = 2$ might make sense in the right application.)

Note that in both of these examples, the k permutations are selected with (nearly) equal probability, but this is not a requirement imposed by this post-processing algorithm.

Examples of P$_i$ used for Post-Processing.

Depending on the requirements of the application, the P_i may be very simple or quite complex. The security of the key generation method depends on the P_i being *permutations*.

1. An example of a very simple permutation P_i is bitwise XOR with a fixed mask A_i: $P_i(r_2) = (r_2 \text{ XOR } A_i)$, where r_2 and A_i are M-bit vectors. Continuing this example, if there are four such masks ($k = 4$), the simple function F(r_1) that maps r_1 into an integer represented by the two rightmost bits of r_1 (say, '01' corresponds to 1, '02' corresponds to 2, '03' corresponds to 3, and '00' corresponds to 4) could be used to choose among them. Then the post-processor's output $P_{\mathrm{F}(r_1)}(r_2)$ would be r_2 XOR $A_{F(r_1)}$.

   Note that in this example, $2 \le N \le M$, where N is the length of r_1, and M is the length of r_2.

   [This should not be confused with the XORing defined in equation (1) above. The equation in (1) is applied after each of the U and V values is calculated, including any qualified post-processing, if applicable.]

2. A more complex example would be the use of a codebook to effect a permutation. For example, $P_i(r_2)$ = Triple-DES(key_i, r_2) could be used on an RNG whose outputs were 64-bit strings. Similarly, $P_i(r_2)$ = AES(key_i, r_2) could be used to effect permutations on an RNG with 128-bit outputs.

   Suppose that there are ten 256-bit AES keys (k = 10). Let F(r_1) = SHA256(r_1) mod 10. Then the post-processed output $P_{F(r_1)}(r_2)$ would be AES($key_{SHA256(r1) \bmod 10}$, r_2) . Note that in this case, $4 \le N \le M$, where N is the length of r_1, and M is the length of r_2 (the minimum length of r_1 is determined by the modulus value 10, which is represented in binary as 4 bits).

   A similar example, but one with a *much* larger value for k, (e.g., $k = 2^{128}$), might use key_i = SHA256(128-bit representation of i). Let F(r_1) = SHA256(r_1). The output $P_{F(r_1)}(r_2)$ of the post-processor would be AES(SHA256(r_1), r_2). Note that is this case, $N = M = 128$.

3. An example of a permutation somewhere between these extremes of complexity is a byte-permutation '$SBOX_i$', which will be applied to each byte of input, with the final output being the concatenation of the individually permuted bytes:

   $$P_i(B_1||B_2|| \ldots||B_{M/8}) = SBOX_i(B_1)||SBOX_i(B_2)||\ldots||SBOX_i(B_{M/8})$$

   For specificity, suppose that M = 128; there are just 2 byte permutations to choose from, $SBOX_0$ and $SBOX_1$; and F maps 8-bit strings to their parity: F(r_1) = 0 if r_1 has an even number of 1's, and F(r_1) = 1 if r_1 has an odd number of 1's. Note that in this case, N = 8.

   The post-processor's output $P_{F(r_1)}(r_2)$, on the input pair r_1 and $r_2 = B_1||B_2|| \ldots||B_{16}$ would be $SBOX_{parity(r1)}(B_1)$ || $SBOX_{parity(r1)}(B_2)$ ||…|| $SBOX_{parity(r1)}(B_{16})$. To complete the example, suppose that the two byte permutations are specified as: $SBOX_0$ = the AES SBOX, and $SBOX_1$ is the inverse permutation to the same AES SBOX.

Additional Comments

1. The concatenation step (formula (2)) must be performed last. An example of the danger of performing the concatenation earlier in the process, followed by other operations is the following: Let U be an n-bit-long output of the RNG with a security strength of n bits. Let V be an n-bit-long publically-known constant. Compute W as W = U || V. W is 2n bits long and has a security strength of n bits. Now, truncate the leftmost n bits of W to obtain key X (i.e., X now consists of the rightmost n bits, which is V, the constant). What we get is a constant. The module should not end up with a known constant and certainly the claim of the security strength of X = (security strength of W) * (n/2n) = n/2 bits would not apply to this constant.
2. The processes described in the "Qualified Post-Processing Algorithms" section must be performed prior to the operations performed individually on U and V in examples 1 through 6 in the Resolution section of this Implementation Guidance, since the latter processes may result in a change in the length of the processed value. The permutations must be applied first.
3. This Implementation Guidance only addresses key generation based, at least partially, on the output value from an approved RNG. It does not address the derivation (i.e., generation) of keys from other keys; this topic is addressed in SP 800-108. The CMVP will issue separate guidance for using SP 800-108.
4. The CMVP does not encourage the use of the Qualified Post-Processing Algorithms. In the vast majority of cases, the methodology shown in examples 1 through 6 should be sufficient to generate a secret value (e.g., a cryptographic key). However, post-processing, as described in this IG, is permitted.

5. It is the vendor's responsibility to demonstrate how their key generation method satisfies the requirements of this Implementation Guidance. The best way to do this is to map their method into one of the examples shown in this Implementation Guidance.

6. In order to make the language of this Implementation Guidance consistent with that of NIST Special Publication 800-90, the IG discusses the security strength (rather than the entropy) of the generated secret value K. The vendor is responsible for demonstrating that the Random Number Generator used in the generation of K received sufficient entropy for the purposes of its applications.

Test Requirements

Code review, vendor documentation review, and mapping of the module's key generation procedures into the methods described in this Implementation Guidance.

7.9 Procedural CSP Zeroization

Applicable Levels:	*All*
Original Publishing Date:	*03/24/2009*
Effective Date:	*03/24/2009*
Last Modified Date:	*03/24/2009*
Relevant Assertions:	*AS07.41, AS07.42*
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Background

FIPS 140-2 Section 4.7.6 states "A cryptographic module shall provide methods to zeroize all plaintext secret and private cryptographic keys and CSPs within the module."

Question/Problem

A module shall provide methods to zeroize all plaintext permanent, temporary and ephemeral CSPs within the module. These methods may be operational (i.e. a callable service invoked by the operator of a module), or methods commonly referred to as *procedural zeroization* methods. What are acceptable methods?

Resolution

The zeroization methods required in AS07.41 are operational or procedural methods that will provide an operator of a module a method to zeroize all permanent, temporary and ephemeral plaintext CSPs. This **shall** be done with a level of assurance that the CSPs can not be easily recovered. However this **shall** not include methods of recovery that require substantial skill and methods that may be employed by governmental or other well funded institutions. As an operational or procedural method, the time necessary to perform the zeroization **shall** be reasonable based on the method employed.

- For software modules, a procedural method may include the uninstallation of the cryptographic module application, *and* reformatting of and overwriting, at least once, the platform's hard drive or other permanent storage media. Only performing the procedural uninstallation of the cryptographic module application is not an acceptable method.

- For space-based modules, a procedural method that relies on the de-orbit destruction is acceptable *only if* the vendor of the module provides analysis that indicates the components where plaintext CSPs may reside have a high probability of destruction and non-recovery.

- All procedural or operational zeroization methods **shall** be performed by the operator of the module while the operator is in control of the module (i.e. present to observe the method has completed successfully or controlled via a remote management session). If the method is not under the direct control of the operator, then rationale **shall** be provided on how the zeroization method(s) are employed such that the secret and private cryptographic keys and other CSPs within the module cannot be obtained by an attacker.

- Except for space-based modules, physical destruction of the module is not considered an acceptable zeroization method.

Additional Comments

TE07.41.03 is revised as follows**:**

TE07.41.03: The tester **shall** initiate zeroization and verify the key destruction method is performed in a sufficient time that an attacker can not access plaintext secret and private cryptographic keys and other plaintext CSPs while under the direct control of the operator of the module (i.e. present to observe the method has completed successfully or controlled via a remote management session).. If the method is not under the direct control of the operator, then rationale shall be provided on how the zeroization method(s) are employed such that the secret and private cryptographic keys and other CSPs within the module cannot be obtained by an attacker.

7.10 Using the SP 800-108 KDFs in FIPS Mode

Applicable Levels:	*All*
Original Publishing Date:	*10/22/2009*
Effective Date:	*10/22/2009*
Last Modified Date:	*10/22/2009*
Relevant Assertions:	*AS07.11 and AS07.16*
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Background

When a key is shared between two entities, it may be necessary to derive additional keying material using the shared key. NIST SP 800-108 provides Key Derivation Functions (KDFs) for deriving keys from a shared key; in NIST SP 800-108, the shared key is called a pre-shared key. The shared key may have been generated, entered or established using any method approved or allowed in FIPS mode.

Note that IG D.2 contains key establishment methods, and includes KDFs that are used during key agreement to derive keying material from a shared secret, which is the result of applying a Diffie-Hellman or MQV primitive. The keying material may be used as a key directly or to derive further keying material.

IG 7.2 defines IEEE 802.11i KDFs that may be used to derive further keying material.

Question/Problem

Where do the KDFs from NIST SP 800-108 fit in the key establishment process, and under what conditions can these KDFs be used in FIPS mode? Are there any other allowed methods for deriving additional keys from a pre-shared key?

Resolution

All key derivation methods listed in NIST SP 800-108 will be allowed in FIPS mode if the Key Derivation Key K_I, as introduced in Section 5 of NIST SP 800-108 has been generated, entered or established using any method approved or allowed in FIPS mode.

Note that the KDFs described IG 7.2 are included in SP 800-108, thus making IG 7.2 obsolete.

Other KDFs that are allowed for key derivation from shared keying material are:

1. The KDF specified in the Secure Real-time Transport Protocol (SRTP) defined in RFC 3711.

Additional Comments

A key hierarchy as specified in Section 6 of NIST SP 800-108 may be used.

Section 8 – Electromagnetic Interference/Electromagnetic Compatibility (EMI/EMC)

Section 9 – Self-Tests

9.1 Known Answer Test for Keyed Hashing Algorithm

Applicable Levels:	*All*
Original Publishing Date:	*02/10/2004*
Effective Date:	*02/10/2004*
Last Modified Date:	*09/22/2004*
Relevant Assertions:	*AS09.07*
Relevant Test Requirements:	*TE09.07.01*
Relevant Vendor Requirements:	*VE09.07.01*

Background

Several keyed hashing algorithms are FIPS-approved (e.g. HMAC-SHA-1, HMAC-SHA-2) and have different levels of complexity that determine the power-on Know-Answer-Test (KAT) requirements.

Question/Problem

What are the KAT requirements when implementing keyed hashing algorithms in FIPS mode?

Resolution

The following table summarizes the minimal KAT requirements:

KAT Requirements	Keyed Hashing algorithm	Underlying algorithm
Triple-DES MAC	No	Yes
HMAC-SHA-1	Yes	No
HMAC-SHA-224	Yes	No
HMAC-SHA-256	Yes	No
HMAC-SHA-384	Yes	No
HMAC-SHA-512	Yes	No

Rationale

Triple-DES MAC algorithms do not include much additional complexity over the underlying algorithmic engine (e.g. Triple-DES). However, keyed hashing algorithms such as HMAC-SHA-1 have additional complexity over the underlying algorithmic engine (e.g. SHA-1). A KAT performed on the Triple-DES algorithms adequately verifies their associated hashing algorithm. This is not the case for the keyed hashing algorithm using a SHS algorithm which implements several other functions in addition to the underlying SHS algorithm.

Additional Comments

As discussed in FIPS 140-2 IG 9.3, if HMAC-SHA-1 is used as the Approved integrity technique to verify the software or firmware components as specified in AS.06.08, a KAT is not required for either the HMAC-SHA-1 or the underlying SHA-1 algorithm.

9.2 Known Answer Test for Embedded Cryptographic Algorithms

Applicable Levels:	*All*
Original Publishing Date:	*02/10/2004*
Effective Date:	*02/10/2004*
Last Modified Date:	*08/19/2004*
Relevant Assertions:	*AS09.19*
Relevant Test Requirements:	*TE09.19.01-03*
Relevant Vendor Requirements:	*VE09.19.01-02*

Background

Core cryptographic algorithms are often embedded into other higher cryptographic algorithms for their operation in FIPS mode (e.g. SHA-1 algorithm embedded into HMAC-SHA-1 and DSA, Triple-DES into RNGs). FIPS 140-2 requires that cryptographic modules that implement FIPS-approved algorithms used in FIPS mode perform a Known-Answer-Test (KAT) as part of their power-up self-tests. This requirement is also valid for the core cryptographic algorithm implementation. However, when the cryptographic module performs the KAT on the higher cryptographic algorithm, the embedded core cryptographic algorithm may also be self-tested.

Question/Problem

If an embedded core cryptographic algorithm is self-tested during the higher cryptographic algorithm KAT, is it necessary for the cryptographic module to implement a KAT for the already self-tested core cryptographic algorithm implementation?

Resolution

It is acceptable for the cryptographic module not to perform a KAT on the embedded core cryptographic algorithm implementation if;

1. the higher cryptographic algorithm uses that implementation,

2. the higher cryptographic algorithm performs a KAT at power-up and,

3. all cryptographic functions within the core cryptographic algorithm are tested (e.g. encryption and decryption for Triple-DES).

Additional Comments

If the cryptographic module contains several core cryptographic algorithm implementations (e.g., several different implementations of SHA-1 algorithm) and some are not used by other higher FIPS-approved cryptographic algorithms (and are therefore not self-tested), then the cryptographic module must perform a KAT at power-up for each of those implementations.

Implementation of Triple-DES within an RNG such as ANSI X9.31 does not meet bullet #3 above since not all the Triple-DES cryptographic functions are tested (e.g. encrypt is performed in the RNG generation, not decrypt)

Implementation of SHA-1 within the FIPS 186-2 random number generation algorithms does not meet bullet #3 above since the hashing function is not completely performed

9.3 KAT for Algorithms used in an Integrity Test Technique

Applicable Levels:	*All*
Original Publishing Date:	*02/10/2004*
Effective Date:	*02/10/2004*
Last Modified Date:	*02/10/2004*
Relevant Assertions:	*AS06.08 and AS09.16*
Relevant Test Requirements:	*TE06.08.01-02 and TE09.16.01-02*
Relevant Vendor Requirements:	*VE06.08.01 and VE09.16.01*

Background

AS06.08 requires that a cryptographic mechanism using an Approved integrity technique **shall** be applied to all cryptographic software and firmware components within the cryptographic module. AS09.16 requires that a cryptographic algorithm test using a Known-Answer-Test (KAT) **shall** be conducted for all cryptographic functions of each Approved cryptographic algorithm implemented by the cryptographic module and used in FIPS mode of operation.

Question/Problem

Must a cryptographic module implement a separate KAT for the underlying cryptographic algorithm used in the Approved integrity technique?

Resolution

A cryptographic module may not implement a separate KAT for the underlying cryptographic algorithm used for the Approved integrity technique if all the cryptographic functions of the underlying cryptographic algorithm are tested (e.g. encryption and decryption for Triple-DES).

Rationale

The software/firmware integrity check using an Approved integrity technique is considered a KAT since the cryptographic module uses itself as an input to the algorithm and a known answer as the expected output.

EX: If HMAC-SHA-1 is used as the Approved integrity technique to verify the software or firmware components, a KAT is not required for either the HMAC-SHA-1 or the underlying SHA-1 algorithm.

EX: If Triple-DES MAC is used as the Approved integrity technique to verify the software or firmware components, a KAT is still required for the underlying Triple-DES as the integrity checking may not use both the Triple-DES encrypt and decrypt functions.

EX: If RSA is used to verify the signature of the software or firmware components, a KAT is still required for the underlying RSA as the integrity checking would not use the RSA signature generation function. However, a KAT for the underlying SHA-1 hashing function is not required.

Additional Comments

9.4 Cryptographic Algorithm Tests for SHS Algorithms and Higher Cryptographic Algorithms Using SHS Algorithms

Applicable Levels:	*All*
Original Publishing Date:	*08/19/2004*
Effective Date:	*08/19/2004*
Last Modified Date:	*01/16/2008*
Relevant Assertions:	*AS09.16*
Relevant Test Requirements:	*TE09.16.01*
Relevant Vendor Requirements:	*VE09.16.01*

Background

Cryptographic algorithm test. A cryptographic algorithm test using a known answer **shall** be conducted for all cryptographic functions (e.g., encryption, decryption, authentication, and random number generation) of each Approved cryptographic algorithm implemented by a cryptographic module. A known-answer test involves operating the cryptographic algorithm on data for which the correct output is already known and comparing the calculated output with the previously generated output (the known answer). If the calculated output does not equal the known answer, the known-answer test **shall** fail.

Cryptographic algorithms whose outputs vary for a given set of inputs (e.g., the Digital Signature Algorithm) **shall** be tested using a known-answer test or **shall** be tested using a pair-wise consistency test (specified below).

Each algorithm implementation to be used in a FIPS Approved mode of operation must implement a cryptographic algorithm test. The cryptographic algorithm test is a *health check* of the algorithm implementation performed at power-up or on demand.

Question/Problem

What are the minimum requirements placed on Known Answer Tests (KATs) for SHS algorithms and higher cryptographic algorithms implementing SHS algorithms so that they can be used in FIPS Approved mode of operation? What are the minimum requirements placed on a pair-wise consistency test (for public and private keys) if performed at power-up or on demand?

Resolution

Following is a subset of algorithm KAT specific implementation guidance:

- the following are minimal requirements for SHS algorithms:
  - a KAT for SHA-1 (if applicable) is required;
  - a KAT for SHA-256 (if applicable) is required;
  - a KAT for SHA-224 (if applicable) is required if SHA-224 is implemented without SHA-256;
  - a KAT for SHA-512 (if applicable) is required; and,
  - a KAT for SHA-384 (if applicable) is required if SHA-384 is implemented without SHA-512.

- a KAT or pair-wise consistency for DSA and RSA (if applicable) is required and **shall** be performed on:
  - at minimum, the smallest NIST-Recommended modulus size or DSA prime that is supported by the module; and,
  - at minimum, any one of the implemented underlying SHS algorithms used by the higher cryptographic algorithm.

- an RSA KAT **shall** be performed using both the public and private exponents (*e* and *d*) and the two exponents **shall** correspond [that is, $d * e = 1 \pmod{(p-1)(q-1)}$].

- a KAT or pair-wise consistency for ECDSA (if applicable) is required and **shall** be performed at a minimum, on:
    - any one of the implemented curves in each of the implemented two types of fields (i.e., prime field where *GF(p)*, and binary field where $GF(2^m)$); and
    - any one of the implemented underlying SHS algorithms used by the higher cryptographic algorithm.

- a KAT for HMAC (if applicable) is required and **shall** be performed at minimum, on any one of the implemented underlying SHS algorithms.

Additional Comments

FIPS 140-2 IG 9.2 *Known Answer Test for Embedded Crypto Algorithms* applies.

This IG is consistent with FIPS 140-2 IG 9.1 *Known Answer Test For Keyed Hashing Algorithm*.

Rationale: The purpose of a KAT is to perform a health-check of the cryptographic module to identify catastrophic failures or alterations of the module between power cycles and not that the implementation is correct. The implementation verification is performed during the cryptographic algorithmic testing and validation.

9.5 Module Initialization during Power-Up

Applicable Levels:	ALL
Original Publishing Date:	04/01/2009
Effective Date:	04/01/2009
Last Modified Date:	04/01/2009
Relevant Assertions:	*AS.09.08, AS.09.09, AS.09.10, AS.09.11*
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Background

Power-up tests **shall** be performed by a cryptographic module when the module is powered up. All data output via the data output interface **shall** be inhibited when the power-up tests are performed.

Question/Problem

What is the *initialization period* and what module activities are allowed to occur during that period?

Resolution

The *initialization period* is the period between the time power is applied to the module (after being powered off, reset, rebooted, instantiated, etc), and the time the module completes the power-up tests and outputs status (success or failure) indicating that the module is ready or not to perform operational cryptographic functions and services. The module may perform many activities during this period (i.e. before, during or after the power-up tests are performed) prior to the output of status and the module becoming operational. The cryptographic module is not considered to be in a FIPS Approved mode of operation during the *initialization period*.

During the initialization period, the module:

- **shall** perform all the power-up tests required by Section 4.9. When completed, the results (i.e. indications of success or failure) **shall** be output via the "status output" interface; (status output may be implicit or explicit);

- **shall** perform all the necessary internal services required to properly initialize or instantiate the module in conjunction with performing the power up self-tests;

- may receive data and control input via the *data input interface* or *control input interface* (e.g. may receive data and control requests for Approved services that the module may act upon once the initialization period is completed);

- **shall** inhibit all data output via the data output interface *except*:

  – the module is allowed to output, when requested, non-security relevant module identification information, or module identification information. The module **shall** prevent the output of any plaintext secret and private cryptographic keys or CSPs that are contained within the module.

If applicable, the security policy **shall** describe the outputted information and the services performed during the *initialization period*.

Once the initialization period is completed (which includes the power-up tests), the module would transition to the operational state and may start providing Approved cryptographic functions and services (if operating in an Approved mode of operation).

Additional Comments

Rationale: One can consider the services performed to properly initialize or instantiate the module and the exchange of non-security relevant information in conjunction with the power-up tests to be part of the power-up initialization sequence (e.g. a modules handshake during the powering sequence).

9.6 Self-Tests When Implementing the SP 800-56A Schemes

Applicable Levels:	*All*
Original Publishing Date:	*10/21/2009*
Effective Date:	*10/21/2009*
Last Modified Date:	*10/21/2009*
Relevant Assertions:	*AS09.01*
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Background

Section 4.9 of FIPS 140-2 states that "… *A cryptographic module* ***shall*** *perform power-up self-tests and conditional self-tests to ensure that the module is functioning properly. Power-up self-tests* ***shall*** *be performed* *when the cryptographic module is powered up. Conditional self-tests **shall** be performed when an applicable security function or operation is invoked (i.e., security functions for which self-tests are required).*"

FIPS 140-2 Implementation Guidance (IG) D.1, *CAVP Requirements for Vendor Affirmation of NIST SP 800-56A*, states March 24, 2009 as the "Transition End Date." After this date, new FIPS 140-2 report submissions to the CMVP implementing the key agreement schemes based on the discrete logarithm problem in the multiplicative group of a finite field (FFC schemes) or the Elliptic Curve discrete logarithm problem (ECC schemes) **shall** either be compliant with SP 800-56A or (until the end of 2010 only – subject to change) they may implement other methods as described in IG D.2 to be used in a FIPS-approved mode of operation. Key agreement schemes described in IG D.2 that are *allowed* in FIPS mode but not fully compliant with SP 800-56A will be listed on the non-FIPS Approved line of the module's validation certificate. IG G.13 provides examples how the module's validation certificate can be completed.

Question/Problem

What self-tests are required when a cryptographic module implements an Approved SP 800-56A-compliant scheme?

Resolution

When an algorithm such as AES or DSA becomes FIPS-Approved, the module implementing this algorithm is required to perform various self-tests if used in a FIPS-Approved mode of operation: the power-up tests and, if applicable, the conditional tests. While the key agreement methodologies described in SP 800-56A are technically not *algorithms* but *schemes* that will be used in protocols, for FIPS 140-2 they **shall** be considered as cryptographic algorithms and the following described self-tests **shall** be performed.

1. **Power-up Tests.**

   The power-up test requires that the cryptographic module has a set of domain parameters and a key pair that will only be used for a power-up test. At the time that the power-up tests are performed per Section 4.9.1 of FIPS 140-2, the module **shall** both test the consistency of the domain parameters and the correct implementation of a key derivation function defined in SP 800-56A. To verify the consistency of the domain parameters,

   FFC schemes: the module **shall** check that $g^x = y \pmod{p}$;

   ECC schemes: the module **shall** check that $Q = dG$.

   To verify the correct implementation of a key derivation function (either the Concatenation Key Derivation Function specified in Section 5.8.1 or the ASN.1 Key Derivation Function specified in Section 5.8.2 of SP 800-56A), the module **shall** start with an (artificial) shared key *Z* and an *OtherInput* value that is consistent with the definition of the appropriate key derivation function, and compute the *DerivedKeyingMaterial* bit string. This result **shall** then be compared to a previously stored pre-computed value.

   If the test fails the requirements of Section 4.9 of FIPS 140-2, describing what the module does when a self-test fails **shall** apply.

   When performing power-up self-tests for the key derivation functions defined in SP 800-56A, the module can choose any values for the fields included in the *OtherInput* input parameter, as long as they are consistent with the definitions in SP 800-56A. The value of the shared secret *Z* **shall** be non-trivial, i.e., its length **shall** be equal to one of the shared secret lengths supported by the module, and not all of the bits in *Z* can be 0.

2. **Conditional Tests.**

   a. **Pair-Wise Consistency Tests**

The *pair-wise consistency* test for a module implementing the SP 800-56A-compliant schemes **shall** be performed by the cryptographic module when a (private, public) key pair, either static or ephemeral, is generated or received by the cryptographic module. No such test is required by an entity that does not have the private key of the key pair (e.g., a recipient in a key agreement transaction). See Appendix B of FIPS 186-3 for the explanation of all parameters.

FFC schemes: For the domain parameters (p, q, g) and the private and public key pair (x, y), the module **shall** test that

$$g^x = y \,(\mathrm{mod}\, p). \qquad [1]$$

ECC schemes: For the domain parameters (q, a, b, G, n, h) and the private and public key pair (d, Q), the module **shall** test that

$$Q = dG. \qquad [2]$$

If the test fails, the requirements of Section 4.9 of FIPS 140-2 describing what the module does when a self-test fails, **shall** apply.

Since the pair-wise consistency test consists of recomputing the public key from the private key and the domain parameters, the pair-wise consistency test **shall** be implemented as a different routine from the key generation routine. This justifies the apparent overhead of having two implementations of the same routine. Since there is no control over how and when the key pair was first calculated, (at a minimum) the test will obtain assurance that the private and public keys are consistent.

b. Public Key Validation Tests

The recipient of a public key needs to obtain assurance of the *validity* of that key, i.e., confidence that the public key(s) of the other party in a key agreement transaction is (are) arithmetically correct, given the set of domain parameters. Note that the recipient may be either an initiator or a responder in a key agreement transaction, depending on the scheme. For example, when the dhHybridOneFlow scheme is used, the initiator is the recipient of the responder's static public key, and the responder is the recipient of the initiator's static public key and ephemeral public key.

According to SP 800-56A, there are three ways for a recipient to obtain assurance of public key validity for the public key owned by the other party in a key agreement transaction: 1) the recipient performs a public key validation test, 2) a trusted third party (trusted by the recipient) performed a successful public key validation test, or 3) a trusted third party (trusted by the recipient) generated the key pair (i.e., the public key and the associated private key) and provided it to the other party .

For static public key agreement keys, at least one of the following two conditions **shall** be satisfied:

1. The cryptographic module's Security Policy states that, in the Approved mode of operation and when acting as the recipient, the cryptographic module always uses static public key agreement keys associated with the other party in a key agreement scheme that have been generated or validated by a trusted third party (trusted by the recipient) -for example, the public key could have been validated by a CA that is trusted by the recipient. In this case, no public key validation test is required to validate the other party's public key.

2. The module **shall** perform the appropriate public key validity test specified in SP 800-56A. For FFC schemes, the test is specified in Section 5.6.2.4; for ECC schemes, the test is specified in Section 5.6.2.5.

For key agreement schemes using ephemeral keys provided by the other party, the module **shall** perform the appropriate test as specified in SP 800-56A, with the exceptions identified in IG 7.10.

For FFC schemes, the public key validation test is specified in Section 5.6.2.4; for ECC schemes, the test is specified in either Section 5.6.2.5 or 5.6.2.6.

Additional Comments

1. There is no difference between tests for the Diffie-Hellman and the MQV schemes. The self-tests are health checks only. They only indicate that keys appear to be consistent, not that the key agreement scheme itself is correctly implemented. The latter verification is performed during the validation testing by an accredited testing laboratory.

2. No domain parameter validation (such as the (*p*, *q*, *g*) triplet for an FFC scheme) will be required for self testing.

3. Tests [1] and [2] defined above make sense both as power-up tests and as pair-wise consistency tests, since they represent the simplest way to check that the underlying arithmetic works properly and that the set of generated parameters is consistent. However, for a recipient of a public key, where the private key is not available, the module needs to verify that the received public key is valid, as specified in SP 800-56A, Sections 5.6.2.4, 5.6.2.5, or 5.6.2.6.

4. No power-up self-tests are required at this time for the key derivation functions not defined in SP 800-56A.

5. When the Security Policy states that the static public keys associated with the other party in a key agreement transaction have been generated or validated by a trusted third party, it is not required that the trusted third party is identified in the Security Policy. The user of the cryptographic module will ultimately decide which trusted third party to trust. An example of an acceptable scenario that provides assurance of public key validity without an explicit public key validation by the recipient is the case where a CA that is trusted by the recipient performs a public key validation when certifying the public key of the other party in a key agreement transaction. Another scenario is the case where a trusted party (e.g., NIST) generates static key pairs for its employees, and NIST is trusted to generate the key pairs correctly by parties both inside and outside of NIST.

Test Requirements

The vendor and tester evidence **shall** be provided under FIPS 140-2 DTR AS09.16.

Section 10 – Design Assurance

Section 11 – Mitigation of Other Attacks

Section 12 – Appendix A: Summary of Documentation Requirements

Section 13 – Appendix B: Recommended Software Development Practices

Section 14 – Appendix C: Cryptographic Module Security Policy

14.1 Level of Detail When Reporting Cryptographic Services

Applicable Levels:	*All*
Original Publishing Date:	*11/15/2001*
Effective Date:	*11/15/2001*
Last Modified Date:	*11/15/2001*
Relevant Assertions:	*AS01.02, AS01.03, AS01.12, AS01.16, AS03.14, AS10.06, AS14.02, AS14.03, AS14.04, AS14.06, AS14.07*
Relevant Test Requirements:	*TE01.03.01, TE01.03.02, TE01.16.01, TE03.14.01, TE10.06.01, TE14.07.01, TE14.07.02*
Relevant Vendor Requirements:	*VE01.03.01, VE01.03.02, VE01.16.01, VE03.14.01, VE03.14.02, VE10.06.01, VE14.07.01, VE14.07.02, VE14.07.03*

Question/Problem

What is the level of detail that the non-proprietary security policy must contain in order to describe the cryptographic service(s) implemented by a cryptographic module?

Resolution

When presenting information in the non-proprietary security policy regarding the cryptographic services that are included in the module validation, the security policy **shall** include, at a minimum, the following information **for each service**:

- The service name
- A concise description of the service purpose and/or use (the service name alone may, in some instances, provide this information)
- A list of Approved security functions (algorithm(s), key management technique(s) or authentication technique) used by, or implemented through, the invocation of the service.
- A list of the cryptographic keys and/or CSPs associated with the service or with the Approved security function(s) it uses.
- For each operator role authorized to use the service:
  - Information describing the individual access rights to all keys and/or CSPs
  - Information describing the method used to authenticate each role.

The presentation style of the documentation is left to the vendor. FIPS 140-2, Appendix C, contains tabular templates that provide non-exhaustive samples and illustrations as to the kind of information to be included in meeting the documentation requirements of the Standard.

Additional Comments

FIPS 140-2 requires information to be included in the module security policy which:

- Allows a user (operator) to determine when an approved mode of operation is selected (**AS01.06, AS01.16**).
- Lists all security services, operations or functions, both Approved and non-Approved, that are provided by the cryptographic module and available to operators (**AS01.12, AS03.07, AS03.14, AS14.03**).
- Provides a correspondence between the module hardware, software, and firmware components (**AS10.06**)
- Provides a specification of the security rules under which the module **shall** operate, including the security rules derived from the requirements of FIPS 140-2. (**AS14.02**)
- For each service, specifies a detailed specification of the service inputs, corresponding service outputs, and the authorized roles in which the service can be performed. **(AS03.14, AS14.03)**

See also the definitions of ***Approved mode of operation*** and ***Approved security function*** in FIPS 140-2.

14.2 Level of Detail When Reporting Mitigation Of Attacks

Applicable Levels:	*All*
Original Publishing Date:	*11/15/2001*
Effective Date:	*11/15/2001*
Last Modified Date:	*11/15/2001*
Relevant Assertions:	*AS14.09*
Relevant Test Requirements:	*TE14.09.01*
Relevant Vendor Requirements:	*VE14.09.01*

Question/Problem

What is the level of detail that the non-proprietary security policy must contain that describes the security mechanism(s) implemented by the cryptographic module to mitigate other attacks?

Resolution

The level of detail describing the security mechanism(s) implemented by the cryptographic module to mitigate other attacks required to be contained in the security policy must be similar to what is found on advertisement documentation (product glossies).

Additional Comments

14.3 Logical Diagram for Software, Firmware and Hybrid Modules

Applicable Levels:	*All*
Original Publishing Date:	*07/03/2007*
Effective Date:	*07/03/2007*

Last Modified Date:	*07/03/2007*
Relevant Assertions:	*AS14.01*
Relevant Test Requirements:	*TE14.01.01*
Relevant Vendor Requirements:	*VE14.01.01*

Background

FIPS 140-2 DTR VE.14.01.01 specifies the requirement for the vendor to provide in the security policy a diagram or image of the physical cryptographic module.

While the requirement is vague when applied to a software, firmware or hybrid cryptographic module, it is intended as well to clearly illustrate the *logical boundary* of the module as well as the other logical objects and the operating environment with which the module executes with.

Question/Problem

For a software, firmware or hybrid cryptographic module, what are the requirements of the *logical diagram* contained in the security policy as specified in VE.14.01.01?

Resolution

The *logical diagram* must illustrate:

- the logical relationship of the software, firmware or hybrid module with respect to the operating environment. This **shall** include, as applicable, references to any operating system, hardware components (i.e. hybrid) other supporting applications, and illustrate the physical boundary of the platform. All the logical and physical layers between the logical object and the physical boundary **shall** be clearly defined.

Additional Comments

The *logical diagram* must convey basic information to the operator of the cryptographic module about its relationship respective to the operating environment.

The *logical diagram* could be a subset of the block diagram specified in AS.01.13.

14.4 Operator Applied Security Appliances

Applicable Levels:	*Level 2, 3 or 4*
Original Publishing Date:	*01/27/2010*
Effective Date:	
Last Modified Date:	*11/25/2009*
Relevant Assertions:	*AS05.15, AS05.26, AS05.35, AS05.49, AS10.04, AS10.22, AS14.01 and AS14.08*
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Background

FIPS 140-2 Section 4.5, *Physical Security*, addresses specific requirements at Level 2. This IG addresses the following two requirements:

1. a module **shall** be constructed in a manner to provide tamper evidence, and
2. a module **shall** have an opaque tamper evident coating or enclosure.

FIPS 140-2 **IG 5.1** provides guidance on opacity and **IG 5.2** on testing of tamper evident seals. Many module implementations are constructed in a manner where the operator of the module is required to install or affix items such as tamper evident seals or security appliances (e.g. baffles, screens, etc.) to configure the module to operate in a FIPS Approved mode of operation. In addition, the operator may over the life-cycle of the module, modify some of the non-security relevant aspects of the module that would require the removal and replacement of tamper evident seals or security appliances.

Question/Problem

What specific information **shall** be included in the test report, certificate and Security Policy when a module at Level 2 has tamper evident seals or security appliances that the operator will apply or modify over the lifecycle of the module?

Resolution

The following specific information **shall** be included in the test report, certificate and Security Policy to meet the relevant assertions:

1. If the module is shipped unassembled, then **AS14.03** **shall** be addressed with appropriate detail.

2. In addition to other applicable caveats, the certificate caveat **shall** include as applicable the following:

   (The <tamper evident seals> and <security devices> installed as indicated in the Security Policy)

3. The Security Policy **shall** include the following:

   a. The reference photo/illustration required in **AS14.01** **shall** reflect the validated module configured or constructed as specified on the. Additional photos/illustrations may be provided to reflect other configurations that may include parts that are not included in the validation.

   b. If filler panels are needed to cover unpopulated slots or openings to meet the opacity requirements, they **shall** be included in the photo/illustration with tamper seals affixed as needed. The filler panels **shall** be included in the list of parts in **AS01.08**.

   c. There **shall** be unambiguous photos/illustrations on the precise placement of any tamper evident seal or security appliance needed to meet the physical security requirements.

   d. The total number of tamper evident seals or security appliances that are needed **shall** be indicated (e.g. 5 tamper evident seals and 2 opacity screens). The photos/illustrations which provide instruction on the precise placement **shall** have each item numbered in the photo/illustration and will equal the total number indicated (the actual tamper evident seals or security appliances are not required to be numbered).

   e. If the tamper evident seals or security appliances are parts that can be reordered from the module vendor, the Security Policy **shall** indicate the module vendor part number of the seal, security appliance or applicable security kit.

      Note: After reconfiguring, the operator of the module may be required to remove and introduce new tamper evident seals or security appliances.

   f. There **shall** be a statement in the Security Policy stating:

The <tamper evident seals> and <security devices> shall be installed for the module to operate in a FIPS Approved mode of operation.

g. The security policy **shall** identify the operator role responsible for:

- securing and having control at all times of any unused seals, and
- the direct control and observation of any changes to the module such as reconfigurations where the tamper evident seals or security appliances are removed or installed to ensure the security of the module is maintained during such changes and the module is returned to a FIPS Approved state.

h. If tamper evident seals or security appliances can be removed or installed, clear instructions **shall** be included regarding how the surface or device shall be prepared to apply a new tamper evident seal or security appliance.

Additional Comments

If a cryptographic module requires more than one tamper evident seal to be applied, the Physical Security Test report that is submitted to the CMVP for review shall address the testing of each tamper evident seal individually if the surface topography or surface material is different between different sets of seals.

FIPS 140-2 Annex A – *Approved Security Functions*

A.1 Validation Testing of SHS Algorithms and Higher Cryptographic Algorithm Using SHS Algorithms

Applicable Levels:	*All*
Original Publishing Date:	*08/19/2004*
Effective Date:	*08/19/2004*
Last Modified Date:	*08/19/2004*
Relevant Assertions:	*AS01.12*
Relevant Test Requirements:	*TE01.12.01*
Relevant Vendor Requirements:	*VE01.12.01*

Background

The Cryptographic Algorithm Validation Program (CAVP) validates every SHS algorithm implementation: SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512. Several higher cryptographic algorithms use those SHS hashing algorithms in their operation.

Question/Problem

What are validation testing requirements for the SHS algorithms and higher cryptographic algorithms implementing SHS algorithms for their use in FIPS Approved mode of operation?

Resolution

To be used in a FIPS Approved mode of operation:

- every SHS algorithm implementation must be tested and validated on the appropriate OS.
- for DSA, RSA, ECDSA and HMAC, every implemented combination must be tested and validated on the appropriate OS.

The algorithmic validation certificate annotates all the tested implementations that may be used in a FIPS Approved mode of operation.

Any algorithm implementation incorporated within a FIPS 140-2 cryptographic module that is not tested may not be used in a FIPS Approved mode of operation. If there is an untested subset of a FIPS Approved algorithm, it would be listed as non-Approved and non-compliant on the FIPS 140-2 validation certificate.

Additional Comments

A.2 Use of Non-NIST-Recommended Asymmetric Key Sizes and Elliptic Curves

Applicable Levels:	*All*
Original Publishing Date:	*09/12/2005*
Effective Date:	*09/12/2005*
Last Modified Date:	*02/26/2007*
Relevant Assertions:	*AS01.12*
Relevant Test Requirements:	*TE01.12.01*
Relevant Vendor Requirements:	*VE01.12.01*

Background

The Cryptographic Algorithm Validation Program (CAVP) validates implementations of DSA, RSA and ECDSA but only for the NIST-Recommended asymmetric key sizes and elliptic curves. The algorithm standards allow the use of other non-NIST-Recommended key sizes and curves. The Cryptographic Algorithm Validation System (CAVS) provided by the CAVP to the CST laboratories does not test for all the possible key sizes and curves that a module may implement.

Question/Problem

Does the CMVP allow the use of non-NIST-Recommended DSA and RSA key sizes and ECDSA curves in a FIPS Approved mode of operation? If so, what are the requirements for those to be used in FIPS mode?

Resolution

The CMVP allows the use of non-NIST-Recommended DSA and RSA key sizes and ECDSA curves in a FIPS Approved mode of operation providing:

- an algorithm implementation must have been tested and validated for at least one NIST-Recommended key size (DSA and RSA) and one NIST-Recommended curve (ECDSA) as applicable,
- the security policy must list all non-NIST-Recommended curves and associated key strengths that are implemented, and,
- the algorithm implementation MUST use an Approved message digest algorithms.

Additional Comments

All NIST-recommended curves, key and modulus sizes must be tested to be used in a FIPS Approved mode of operation.

For NIST-Recommended elliptic curves, the value of f is commonly considered to be the size of the private key (Table 2, NIST SP 800-57). From this value the strength can be determined.

Refer to IG 1.4 *Use of Cryptographic Algorithm Validation Certificates* for guidance on operational environment requirements.

A.3 Vendor Affirmation of Cryptographic Security Methods

Applicable Levels:	*All*
Original Publishing Date:	*01/25/2007*

Effective Date:	*01/25/2007*
Last Modified Date:	*08/04/2009*
Relevant Assertions:	*AS01.12*
Relevant Test Requirements:	*TE01.12.01*
Relevant Vendor Requirements:	*VE01.12.01*

Background

A cryptographic module **shall** implement at least one Approved security function used in an Approved mode of operation. Non-Approved security functions may also be included for use in non-Approved modes of operation or allowed for use in an Approved mode of operation. Documentation **shall** list all security functions, both Approved and non-Approved, that are employed by the cryptographic module and **shall** specify all modes of operation, both Approved and non-Approved. The vendor **shall** provide a validation certificate for all Approved cryptographic algorithms. The tester **shall** verify that the vendor has provided validated certificate(s) as described above.

Questions/Problems

For Approved security functions, Approved random number generators or Approved key establishment techniques specified in FIPS 140-2 Annexes A, C, and D, if CAVP testing is not available, can the Approved methods be used in FIPS mode, and if so, how shall it be tested and annotated on the module validation certificate and security policy?

Resolution

As new methods are published and Approved, they will be added to the relevant FIPS 140-2 Annexes. The annexes may reference FIPS 140-2 Implementation Guidance for methods *allowed* in lieu of Approved methods.

1. If a new Approved methods (e.g. NIST FIPS, Special Publication, etc) are added to the Annexes which provides a new method that did not exist before (e.g. key establishment), until such time that CAVP testing is available for the new method, the CMVP would continue to:

   – allow methods as provided by guidance (untested and listed as non-Approved but *allowed* in FIPS mode); and
   – allow the vendor to implement the new Approved method (untested, listed as Approved and allowed in FIPS mode with the caveat *vendor affirmed*).

   Once testing is deployed by the CAVP to the testing laboratories:

   a. a transition period (e.g. n months) would be provided for new test reports received by the CMVP:

      ▪ during the transition period, a new Approved method would either be listed as Approved with a reference to a CAVP validation certificate, or as *vendor affirmed* if testing was not performed; and
      ▪ allow continued implementation of methods as provided by guidance (untested and listed as non-Approved but *allowed* in FIPS mode).

   b. when the transition period ends, for newly received test reports:

      ▪ only Approved methods that have been tested and received a CAVP validation certificate would be allowed. All other methods would be listed as non-Approved and not allowed in an Approved FIPS mode of operation.

   c. the vendor could optionally follow up with testing of un-tested vendor affirmed methods and if so, the reference to *vendor affirmed* would be removed and replaced by reference to the algorithm certificate. If there are no changes to the module, this change can be submitted under FIPS 140-2 IG G.8 Scenario 1[24]. If the module is changed, this change can be submitted under FIPS 140-2 IG G.8 Scenarios 1, 3 or 5 as applicable [26].

2. If a new Approved methods (e.g. NIST FIPS, Special Publication, etc) are added to Annexes which provides a new method commensurate with those that currently exist (e.g. an new symmetric key algorithm, RNG, DRBG, hash, digital signature, etc), until such time that CAVP testing is available for the new method, the CMVP would:

   – allow prior Approved methods (tested and listed as Approved); and
   – allow the vendor to implement the new Approved method (untested, listed as Approved and allowed in FIPS mode with the caveat *vendor affirmed*)

   Once testing is deployed by the CAVP to the testing laboratories:

   a. a transition period (e.g. n months) would be provided for new test reports received by the CMVP:

      - during the transition period, a new Approved method would either be listed as Approved with a reference to a CAVP validation certificate, or as *vendor affirmed* if testing was not performed.

   b. when the transition period ends, for newly received test reports:

      - only Approved methods that have been tested and received a CAVP validation certificate would be allowed. All other methods would be listed as non-Approved and not allowed in an Approved FIPS mode of operation.

   c. the vendor could optionally follow up with testing of prior un-tested vendor affirmed methods and if so, the reference to *vendor affirmed* removed and replaced by reference to the algorithm certificate. If there are no changes to the module, this change can be submitted under FIPS 140-2 IG G.8 Scenario 1[25]. If the module is changed, this change can be submitted under FIPS 140-2 IG G.8 Scenarios 1, 3 or 5 as applicable [25].

3. The Cryptographic Technology Group at NIST may determine that prior methods may be retroactively disallowed and moved to non-Approved and not allowed in a FIPS mode of operation (e.g. DES). A Federal Register notice would be published with a transition period to allow migration from the no longer Approved or allowed method.

4. For all Approved methods, all applicable FIPS 140-2 requirements **shall** be met (e.g., key management, self-tests, etc.)

Additional Comments

***Vendor Affirmed*:** a security method reference that is listed with this caveat has not been tested by the CAVP, and the CMVP or CAVP provide no assurance regarding its correct implementation or operation. Only the vendor of the module affirms that the method or algorithm was implemented correctly.

The users of cryptographic modules implementing vendor affirmed security functions must consider the risks associated with the use of un-tested and un-validated security functions.

[24] This is a special case where FIPS 140-2 IG G.8 Scenario 2 would not apply.
[25] If the change is security relevant either to the module or the method, then FIPS 140-2 IG Scenarios 3 or 5 would be applicable depending on the extent of the changes. If for example there was a non-security relevant change to the module not associated with the security method implementation, FIPS 140-2 Scenario 1 could be applicable.

Test Requirements

Until the FIPS 140-2 DTR and CRYPTIK tool are updated and released, please provide the following information under VE and TE 01.12.01.

Required Vendor Information

VE01.12.03: The vendor **shall** provide a list of all vendor affirmed security methods.

VE01.12.04: The vendor provided nonproprietary security policy **shall** include reference to all vendor affirmed security methods.

Required Test Procedures

TE01.12.03: The tester **shall** verify that the vendor has provided the list of vendor affirmed security methods as described above.

TE01.12.04: The tester **shall** verify that the vendor provided documentation specifies how the implemented vendor affirmed security methods conform to the relevant standards.

Required Use of "Vendor Affirmed" Caveat

All cryptographic methods that are Approved and *vendor affirmed* **shall** be specified on the certificate and in the security policy, and be annotated with, in addition to the other required caveats as applicable, the caveat (vendor affirmed: *FIPS or NIST Special Publication #*).

Caveat Annotation Examples

The only Approved DRNG implemented is vendor affirmed:
DRNG (SP 800-90, vendor affirmed)

Multiple Approved RNGs are implemented, both tested and vendor affirmed:
RNG (Cert. #nnn); DRNG (SP 800-90, vendor affirmed)

The only Approved Key Agreement Schemes implemented are vendor affirmed:
KAS (SP 800-56A, vendor affirmed)

Key Transport Schemes:
KTS (SP 800-56B, vendor affirmed)

A.4 CAVP Requirements for Vendor Affirmation of NIST SP 800-38D

Applicable Levels:	*All*
Original Publishing Date:	*12/18/2007*
Effective Date:	*12/18/2007*
Transition End Date:	***03/24/2009***
Last Modified Date:	*12/18/2007*
Relevant Assertions:	*AS01.12*
Relevant Test Requirements:	*TE01.12.01*
Relevant Vendor Requirements:	*VE01.12.01*

Background

NIST SP 800-38D was added to FIPS 140-2 Annex A on December 18, 2007. FIPS 140-2 Implementation Guidance, IG 1.10 was added January 25, 2007. Until CAVP testing for NIST SP 800-38D is available, IG 1.10 is applicable. NIST SP 800-38D includes information beyond the specifications of the Galois/Counter Mode itself; i.e., uniqueness requirements on IVs and keys.

Question/Problem

To claim *vendor affirmation* to NIST SP 800-38D, what sections of the standard need to be addressed?

Resolution

Validation testing for NIST SP 800-38D, *Recommendation for Block cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC* includes validation testing for the authenticated encryption function and the authenticated decryption function.. To claim *vendor affirmation* to SP 800-38D, information contained in the following sections that are supported by the implementation under test (IUT) **shall** be implemented:

Section 5	Elements of GCM
Section 6	Mathematical Components of GCM
Section 7	GCM Specifications

Additional Comments

1. The GCM functions in NIST SP 800-38D require the forward direction of an approved symmetric key block cipher with a block size of 128 bits. Currently, the only NIST-approved 128-bit block cipher is the Advanced Encryption Standard (AES) algorithm specified in Federal Information Processing Standard (FIPS) Pub. 197. The validation testing for the forward direction of this supporting algorithm, the AES Cipher (Encrypt) function, is found in its corresponding validation test suite and, therefore, **shall** be validated as a prerequisite to NIST SP 800-38D vendor affirmation.

2. The SP800-38D Self Tests required in cryptographic module implementations **shall** consist of a known answer that validates the correctness of the GCM elements, GCM mathematical components and GCM specifications of the two GCM functions, namely, the authenticated encryption function and the authenticated decryption function.

3. Section 8, *Uniqueness Requirement on IVs and Keys*, and Section 9, *Practical Considerations for Validating Implementations*, contain requirements for module validation, which is conducted by the CMVP. Therefore, Section 8 and Section 9 are outside of the scope of algorithm validation.

Derived Test Requirements

Upon the following successful review, the CST Lab **shall** affirm by annotating the algorithm entry per the IG G.13 annotation requirements

Required Vendor Information

The vendor **shall** provide evidence that their implementation implements the sections outlined above completely and accurately. This **shall** be accomplished by documentation and code review.

Required Test Procedures

The tester **shall** review the vendor’s evidence demonstrating that their implementation conforms to the specifications specified above. This **shall** be accomplished by documentation and code review. The tester **shall** verify the rationale provided by the vendor.

A.5 Key/IV Pair Uniqueness Requirements from NIST SP 800-38D

Applicable Levels:	*All*
Original Publishing Date:	*03/10/2009*
Effective Date:	*03/10/2009*
Last Modified Date:	*03/10/2009*
Relevant Assertions:	
Relevant Test Requirements:	
Relevant Vendor Requirements:	

Background

NIST Special Publication (SP) 800-38D was added to FIPS 140-2 Annex A on December 18, 2007. FIPS 140-2 Implementation Guidance (IG) 1.13, which was added on December 18, 2007, specifies the requirements to claim the vendor affirmation to SP 800-38D. IG 1.13 states that sections 8 and 9 of SP 800-38D are out of scope for CAVP. However, these sections of SP 800-38D are applicable to the CMVP cryptographic module testing and validation, and the probabilistic "uniqueness" of the (key, IV) pair is critical to the security of a cryptographic module that implements the AES Galois/Counter Mode (GCM). Specifically, SP 800-38D requires that **"the probability that the authenticated encryption function ever will be invoked with the same IV and the same key on two (or more) distinct sets of input data shall be no greater than** 2^{-32}**."**

One difficulty of testing the modules compliance with this requirement comes from the fact that each module is tested independently while SP 800-38D demands that the probability of the (Key, IV) pair collision between all modules at all times should be sufficiently low to ensure cryptographic strength.

Question/Problem

How shall a cryptographic module satisfy the requirements of Section 8 of SP 800-38D?

Resolution

There are several scenarios that may take place. First, the AES GCM key may be generated internally in a cryptographic module. Second, the key can be entered into the module.

The IV is generated internally, according to Section 9.1 of SP 800-38D. It may be either generated randomly or set deterministically, possibly by being incremented by 1 every time a new value is needed.

Here is the summary of the requirements that the cryptographic module **shall** satisfy.

1. The external generation of the IVs is not allowed.

2. If the IV used together with the GCM Key is generated internally *randomly* then

   - The generation **shall** use an Approved RNG, and
   - The RNG seed **shall** be generated internally from an internal entropy source.
   - The IV length **shall** be at least 96 bits (per SP 800-38D).

3. If the GCM Key is generated either internally or externally and the IV is generated internally *deterministically* then the requirement of SP 800-38D quoted in the Background section above will be modified. Instead of requiring that the probability of any (key, IV) collision anywhere in the

Universe at all times did not exceed 2^{-32}, it will only be required that for a given key distributed to one or more cryptographic modules, the (key, IV) collision probability would not exceed 2^{-32}. This is equivalent to the requirement that for any key distributed to one or more modules, the probability of a collision between the deterministically-generated IVs is no greater than 2^{-32}.

The text in the rest of this section will specify what the module has to do to meet this requirement.

A. Each deterministically established IV **shall** include an encoding of the module's name and the name **shall** be long enough to allow for at least 2^{32} different names. For example, if the module's name is such that it consists of at least 8 hexadecimal characters then this condition is satisfied, since 16^{8} is no smaller than (indeed, equal to) 2^{32}. Alternatively, if the name consists of at least 6 alphanumerical characters, each having at least 62 values, then this is also sufficient. Even though not all possible names are equally likely to be used, just the fact that the modules can possibly have at least 2^{32} different names will be sufficient to meet this requirement.

B. One of the following conditions must be satisfied:
B1: The module's memory **shall** be set in such way that it will reset to the last IV value used in case the module's power is lost and then restored. (This condition is enforced by the module and **shall** be tested by a testing lab.) **OR**
B2: There will be a human operator who will reset the IV to the last one used in case the module's power is lost and then restored. (This condition is not enforced but **shall** be stated in the module's Security Policy, under the "User Guide" heading.) **OR**
B3: In case the module's power is lost and then restored, the key used for the AES GCM encryption/decryption **shall** be re-distributed. (This condition is not enforced but **shall** be stated in the module's Security Policy, under the "User Guide" heading.)

Additional Comments

1. Having the name field sufficiently long to allow for 2^{32} different names does not in itself guarantee that the entropy of the name will be sufficiently large and that the name collision probability will not be greater than 2^{-32}. However, this is an acceptable solution.
2. The standard sets the minimum security requirements. The buyer is free to demand that the module allows for longer names. Users should be smart enough to name their modules in such a way that name collisions become extremely rare.
3. Including the module's name in the IV field does not amount to a passphrase-based key derivation. The IV is not a key. Their cryptographic properties are different.
4. This IG does not precisely calculate the (key, IV) collision probabilities in cases 2. and 3. in the "Resolution" section above. These probabilities will be very small if the module meets all of the stated requirements.

A.6 CAVP Requirements for Vendor Affirmation of FIPS 186-3 Digital Signature Standard

Applicable Levels:	*All*
Original Publishing Date:	*07/07/2009*
Effective Date:	*07/07/2009*
Transition End Date:	***10/02/2009** – See Below*
Transition End Date:	***06/30/2010** – See Below*

Transition End Date:	*08/27/2010 – See Below*
Last Modified Date:	*04/09/2010*
Relevant Assertions:	*AS01.12*
Relevant Test Requirements:	*TE01.12.01*
Relevant Vendor Requirements:	*VE01.12.01*

Transition

The Transition End Date for those elements of FIPS 186-3 DSA which CAVP testing is currently available [if not supporting the generation and validation of provably prime domain parameters p and q and canonical generation and validation of domain parameter g] is: **October 02, 2009**.

With the March 31, 2010 CAVP release of CAVS 9.0, testing for all elements of FIPS 186-3 DSA are available. For the new and final set of elements, the transition end date is: **June 30, 2010**

With the May 27, 2010 CAVP release of CAVS 10.0, testing for all elements of FIPS 186-3 ECDSA are available. The transition end date is: **August 27, 2010**

Currently the transition plan addressed in *draft* NIST SP 800-131 is not published for migration to FIPS 186-3. In lieu of a published transition plan, implementations to FIPS 186-2 and FIPS 186-3 are valid and allowed in an Approved FIPS mode of operation.

Background

Federal Information Processing Standard (FIPS) 186-3, ***Digital Signature Standard (DSS)*** was added to FIPS 140-2 Annex A on June 18, 2009. FIPS 186-3 specifies a suite of algorithms that can be used to generate a digital signature. These include the DSA, ECDSA, and RSA algorithms. CAVP testing is currently available for DSA as specified in FIPS 186-3, with the exception of generation and validation of provably prime domain parameters *p* and *q* and canonical generation and validation of domain parameter *g*. CAVP testing is not available for ECDSA and RSA. Until CAVP testing for FIPS 186-3 is available for the above elements of DSA and for ECDSA and RSA algorithms, IG A.6 is applicable.

Question/Problem

To claim *vendor affirmation* to the above listed domain parameter generation and validation methods of DSA, ECDSA, and RSA as specified in FIPS 186-3, what sections of the publication needs to be addressed?

Resolution

Validation testing for FIPS 186-3, ***Digital Signature Standard (DSS)*** is separated into the three digital signature algorithms. Validation testing is available for FIPS 186-3 DSA, with the exception of the domain parameter generation and validation method listed above. These methods, along with FIPS 186-3 ECDSA and RSA, will require *vendor affirmation* until validation testing is available in the CAVS tool.

Vendor Affirmation for FIPS 186-3 DSA Domain Parameter Generation and Validation for provable primes p and q and verifiable canonical generation of the generator g

To claim vendor affirmation for FIPS 186-3 DSA generation of provably primes *p* and *q*:

1. The vendor must affirm that the method of FIPS 186-3 A.1.2.1.2 is used to generate provable primes *p* and *q*.
2. 2. The vendor **shall** use the CAVP to validate the underlying SHA implementation used by this DSA implementation and report the validation number.

To claim vendor affirmation for FIPS 186-3 DSA verifiable canonical generation of the generator *g*:

1. The vendor must affirm that the method of FIPS 186-3 A.2.3 is used for verifiable canonical generation of the generator *g*.
2. 2. The vendor **shall** use the CAVP to validate the underlying SHA implementation used by this DSA implementation and report the validation number.

To claim vendor affirmation for FIPS 186-3 DSA validation of provable primes *p* and *q*:

1. The vendor must affirm that the method of FIPS 186-3 A.1.2.2 is used for validation of provable primes *p* and *q*.
2. The vendor **shall** use the CAVP to validate the underlying SHA implementation used by this DSA implementation and report the validation number.

To claim vendor affirmation for FIPS 186-2 DSA validation when the canonical generation of the generator *g* was used:

1. The vendor must affirm that the method of FIPS 186-3 A.2.4 is used for validation of *g* where the verifiable canonical generation of *g* was used.
2. The vendor **shall** use the CAVP to validate the underlying SHA implementation used by this DSA implementation and report the validation number.

Vendor Affirmation for FIPS 186-3 ECDSA

To claim vendor affirmation for FIPS 186-3 ECDSA, the following **shall** be affirmed:

1. For all ECDSA implementations, the assurances listed in FIPS 186-3, Section 3 and 3.1 **shall** be defined. If Signature Validation is implemented, Section 3.3 Assurances are also required.

2. If Key Pair Generation is implemented:
   a. The vendor **shall** affirm that at least one of the methods in FIPS 186-3 Appendix B.4 is used to generate d and Q, the private and public keys.
   b. The implementation must support at least one of the NIST curves in FIPS 186-3 Appendix D.1.
   c. The vendor **shall** use the CAVP to validate the underlying RNG or DRBG implementation used by this ECDSA implementation and report the validation number.

3. If Public Key Validation (PKV) is implemented:
   a. The vendor must run the FIPS 186-2 ECDSA PKV tests and report the validation number.

4. If Signature Generation is implemented:
   a. The vendor **shall** affirm compliance with FIPS 186-3 Section 6.4.
   b. The vendor **shall** affirm compliance with FIPS 186-3 Appendix B.5 for generation of the Per-message secret number.
   c. The vendor **shall** use the CAVP to validate the underlying SHA implementation used by this ECDSA implementation and report the validation number.

5. If Signature Validation is implemented:
   a. The vendor **shall** affirm compliance with FIPS 186-3 Section 6.4.
   b. The vendor **shall** use the CAVP to validate the underlying SHA implementation used by this ECDSA implementation and report the validation number.

Vendor Affirmation for FIPS 186-3 RSA

To claim vendor affirmation for FIPS 186-3 RSA, the following **shall** be affirmed:

1. For all RSA implementations, the assurances listed in Section 3 **shall** be defined.
2. If Key Pair Generation is implemented:
   a. The vendor **shall** affirm that at least one of the methods in FIPS 186-3 Appendix B.3 is used to generate the key pairs.
   b. The vendor **shall** affirm that at least one of the modulus lengths 1024, 2048 or 3072 bits is supported by the implementation. Note, the length of the modulus is dependent on the generation method selected. See FIPS 186-3 Appendix B.3.1.
   c. The vendor **shall** affirm that the public exponent e **shall** be selected with the following constraints:

i. The public verification exponent e **shall** be selected prior to generating the primes p and q, and the private signature exponent d.

ii. The exponent e **shall** be an odd positive integer such that $2^{16} < e < 2^{256}$.

d. The vendor **shall** use the CAVP to validate the underlying SHA implementation used by this RSA Key Pair Generation implementation and report the validation number.

e. The vendor **shall** affirm that the length in bits of the hash function output block **shall** meet or exceed the security strength associated with the bit length of the modulus *n* (see SP800-57).

f. If the RSA parameters are randomly generated (i.e., the primes *p* and *q*, and optionally, the public key exponent *e*), the vendor **shall** use the CAVP to validate the underlying RNG or DRBG implementation used by this RSA implementation and report the validation number.

3. If ANS X9.31 RSA Signature Generation or Signature Verification is implemented:
   a. The vendor must run the ANS X9.31 RSA validation tests and report the validation number. (Note that the specification in FIPS 186-3 Section 5.4 concerning the extraction of the hash value *H(M)'* from the data structure *IR'* is tested in the ANS X9.31 RSA validation testing supplied by the CAVP.)
   b. The vendor **shall** affirm that at least one of the modulus lengths 1024, 2048 or 3072 bits is supported by the implementation.
   c. The vendor **shall** use the CAVP to validate the underlying RNG or DRBG implementation used by this RSA implementation and report the validation number.

4. If PKCS #1 Version 1.5 and/or PKCS #1 Version PSS is implemented:
   a. The vendor **shall** confirm that implementations that generate RSA key pairs use the criteria and methods in FIPS 186-3 Appendix B.3 to generate those key pairs.
   b. The vendor **shall** use the CAVP to validate the underlying approved SHA implementation used by this implementation and report the validation number.
   c. The vendor **shall** confirm that only two prime factors *p* and *q* **shall** be used to form the modulus *n*.
   d. The vendor **shall** use the CAVP to validate the underlying RNG or DRBG implementation used by this RSA implementation and report the validation number.
   e. If PKCS #1 Version 1.5 is implemented, the vendor must run the PKCS1.5 validation tests for Signature Generation and/or Signature Verification and report the validation number.
   f. If PKCS#1 Version PSS is implemented, the vendor must run the PKCSPSS validation tests for Signature Generation and/or Signature Verification and report the validation number.
   g. If PKCS#1 Version PSS is implemented, the vendor **shall** confirm that the implementation's salt length (*sLen*) satisfies 0 <=*sLen*<=*hlen*, where *hlen* is the length of the hash function output block.

Annotation

Refer to IG G.13 for annotation examples.

FIPS 140-2 Section 4.9 Self-Tests

In addition to the above requirements, all algorithmic implementations **shall** meet all the applicable self-test requirements in FIPS 140-2 Section 4.9.

Derived Test Requirements

Upon the following successful review, the CST Lab **shall** affirm by annotating the algorithm entry per the IG G.13 annotation requirements.

Required Vendor Information

The vendor **shall** provide evidence that their implementation implements the sections outlined above completely and accurately. This **shall** be accomplished by documentation and code review.

Required Test Procedures

The tester **shall** review the vendor's evidence demonstrating that their implementation conforms to the specifications specified above. This **shall** be accomplished by documentation and code review. The tester **shall** verify the rationale provided by the vendor.

A.7 CAVP Requirements for Vendor Affirmation of NIST SP800-38E

Applicable Levels:	*All*
Original Publishing Date:	*01/27/2010*
Effective Date:	*01/27/2010*
Transition End Date:	***06/30/2010***
Last Modified Date:	*04/09/2010*
Relevant Assertions:	*AS01.12*
Relevant Test Requirements:	*TE01.12.01*
Relevant Vendor Requirements:	*VE01.12.01*

Background

NIST SP 800-38E, *Recommendation for Block cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Block-Oriented Storage Devices*, was added to FIPS 140-2 Annex A on January 27, 2010. Until CAVP testing for NIST SP 800-38E is available, this IG is applicable. NIST SP 800-38E approves the XTS-AES mode as specified in the Institute of Electrical and Electronics Engineers, Inc (IEEE) Std. 1619-2007, subject to one additional requirement on the lengths of the data units. That is, the data unit for any instance of an implementation of XTS-AES SHALL NOT exceed 2^{20} blocks.

Question/Problem

To claim vendor affirmation to NIST SP 800-38E; what sections of the IEEE standard and the NIST Special Publication need to be addressed?

Resolution

To claim vendor affirmation to NIST SP800-38E, the information contained in the following sections that are supported by the Implementation Under Test (IUT) **shall** be implemented:

SP800-38E	Section 4	Conformance
IEEE Std. 1619-2007	Section 5	XTS-AES transform

The following information **shall** be specified:

1. The underlying AES implementation **shall** be validated by the CAVP:

   a. For XTS-AES Encrypt: the validation referenced **shall** include an AES mode of operation that uses the forward cipher function.

   b. For XTS-AES Decrypt: the validation referenced **shall** include an AES mode of operation that uses the forward and inverse cipher function (i.e., AES-ECB or AES-CBC).

2. The XTS-AES key sizes supported: XTS-AES-128 (256 bits) AND/OR XTS-AES-256 (512 bits).

3. The block sizes supported: complete blocks only OR complete and partial blocks

4. Procedures supported: XTS-AES encryption AND/OR XTS-AES decryption

5. Provide assurance that the length of the data unit for any instance of an implementation of XTS-AES **shall** not exceed 2^{20} blocks.

6. Provide assurance that the XTS-AES key **shall** not be associated with more than one key scope.

Additional Comments

Bullets 5 and 6 above satisfy the **shall** statements included in SP800-38E and IEEE Std 1619-2007 that are not testable by the CAVP.

Upon the following successful review, the CST Lab **shall** affirm by annotating the FIPS Approved algorithm entry as follows:

AES (XTS-AES: AES Cert. #nnn, vendor affirmed)

When CAVP CAVS testing is available, the annotation will simply change to:

AES (Cert. #nnn)

Derived Test Requirements

Required Vendor Information

The vendor **shall** provide evidence that their implementation implements the sections outlined above completely and accurately. This **shall** be accomplished by documentation and code review.

Required Test Procedures

The tester **shall** review the vendor's evidence demonstrating that their implementation conforms to the specifications specified above. This **shall** be accomplished by documentation and code review. The tester **shall** verify the rationale provided by the vendor.

FIPS 140-2 Annex B – *Approved Protection Profiles*

FIPS 140-2 Annex C – *Approved Random Number Generators*

C.1 CAVP Requirements for Vendor Affirmation of NIST SP 800-90

Applicable Levels:	*All*
Original Publishing Date:	*06/21/2007*
Effective Date:	*06/21/2007*
Transition End Date:	***02/15/2008***
Last Modified Date:	*06/21/2007*
Relevant Assertions:	*AS01.12*
Relevant Test Requirements:	*TE01.12.01*
Relevant Vendor Requirements:	*VE01.12.01*

Background

NIST Special Publication 800-90 was added to FIPS 140-2 Annex C on January 24, 2007. FIPS 140-2 Implementation Guidance, IG A.3, was added January 25, 2007. Until CAVP testing for NIST SP 800-90 is available, IG A.3 is applicable. NIST SP 800-90 includes information beyond the specifications of the deterministic random bit generation (DRBG) algorithms themselves, e.g., stricter entropy requirements, and assurance.

Question/Problem

To claim *vendor affirmation* to NIST SP 800-90, what sections of the publication need to be addressed?

Resolution

To claim *vendor affirmation*, the vendor **shall** affirm compliance with the following three sections of NIST SP 800-90, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*:

Section 9	DRBG Mechanism Functions
Section 10	DRBG Algorithm Specifications
Section 11	Assurance

The vendor is not required to meet the requirements in Section 8, including the entropy requirements in Section 8.6. Entropy requirements are addressed in FIPS 140-2 DTR AS.07.13.

Additional Comments

The requirements of NIST SP 800-90 depend on several NIST Approved security functions, for example, SHA, AES, and three-key Triple-DES. The validation testing for these supporting security functions is found in their corresponding validation test suites and, therefore, they **shall** be validated as a prerequisite to NIST SP 800-90 vendor affirmation.

To claim vendor affirmation to NIST SP 800-90, the following supporting security functions, if used, **shall** be tested and validated:

- Supported hash algorithms (SHA224, SHA256, SHA384, and/or SHA512)
- Supported Message Authentication Code (MAC) algorithm (HMAC)

- Advanced Encryption Standard (AES)
- Three key Triple-DES

Derived Test Requirements

Upon the following successful review, the CST Lab **shall** affirm by annotating the algorithm entry per the IG G.13 annotation requirements

Required Vendor Information

The vendor **shall** provide evidence that their implementation implements the sections outlined above completely and accurately. This **shall** be accomplished by documentation and code review.

Required Test Procedures

The tester **shall** review the vendor's evidence demonstrating that their implementation conforms to the specifications specified above. This **shall** be accomplished by documentation and code review. The tester **shall** verify the rationale provided by the vendor.

C.2 Use of other Core Symmetric Algorithms in ANSI X9.31 RNG

Applicable Levels:	*All*
Original Publishing Date:	*01/21/2005*
Effective Date:	*01/21/2005*
Last Modified Date:	*01/21/2005*
Relevant Assertions:	*AS07.10*
Relevant Test Requirements:	*TE07.10.01*
Relevant Vendor Requirements:	*VE07.10.01*

Background

ANSI X9.31 Appendix A.2.4 specifies 2-key Triple-DES as the core symmetric algorithm in its deterministic random number generator.

Question/Problem

Is it acceptable to use other FIPS Approved symmetric algorithms as the ANSI X9.31 Appendix A.2.4 RNG core algorithm?

Resolution

In addition to 2-key Triple-DES, it is acceptable to use the following FIPS Approved symmetric algorithms as the ANSI X9.31 RNG core algorithm:

- AES
- 3-key Triple-DES
- SKIPJACK

CAVS testing is available for the 2-key Triple-DES, 3-key Triple-DES and AES. Until such time as CAVS testing is available for RNG testing using SKIPJACK, for module testing purposes, the core cryptographic algorithm SKIPJACK **shall** be validated and the RNG implementation will be marked as “vendor affirmed”.

Additional Comments

FIPS 140-2 Annex C has been updated to include reference to the NIST RNG specification for implementing 3-key Triple-DES and AES with ANSI X9.31 Appendix A.2.4.

FIPS 140-2 Annex D – *Approved Key Establishment Techniques*

D.1 CAVP Requirements for Vendor Affirmation of NIST SP 800-56A

Applicable Levels:	*All*
Original Publishing Date:	*06/21/2007*
Effective Date:	*06/21/2007*
Transition End Date:	***03/24/2009***
Last Modified Date:	*10/20/2009*
Relevant Assertions:	*AS01.12*
Relevant Test Requirements:	*TE01.12.01*
Relevant Vendor Requirements:	*VE01.12.01*

Background

NIST Special Publication 800-56A was added to FIPS 140-2 Annex D on January 24, 2007. FIPS 140-2 Implementation Guidance, IG A.3, was added January 25, 2007. Until CAVP testing for NIST SP 800-56A is available, IG A.3 is applicable. NIST SP 800-56A includes information beyond the specifications of the key agreement algorithm itself; i.e. Instructions to the implementer to aid in the implementation of the algorithm.

Question/Problem

To claim *vendor affirmation* to NIST SP 800-56A, what sections of the publication need to be addressed?

Resolution

Validation testing for NIST SP 800-56A, *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography* includes validation testing for the key agreement schemes and key confirmation. To claim *vendor affirmation* to SP 800-56A, information contained in the following sections that are supported by the implementation under test (IUT) **shall** be implemented:

Section 5.6.2.4	FFC Full Public Key Validation Routine (if implement FFC)
Section 5.6.2.5	ECC Full Public Key Validation Routine (if implement ECC)
Section 5.7	DLC Primitives
Section 5.8	Key Derivation Functions for Key Agreement Schemes
Section 6	Key Agreement

If key confirmation is supported by the implementation, the applicable information contained in the following section must be implemented:

Section 8	Key Confirmation

Additional Comments

1. The components in SP 800-56A **shall** only be used within the SP 800-56A protocol. This includes the full public key validation routines, the DLC primitives, the key derivation functions, the key agreement functions, and the key confirmation functions.

2. The requirements specified in NIST SP 800-56A depend on several NIST Approved security functions, for example, SHA, DSA, ECDSA, etc. While validation testing for NIST SP 800-56A concentrates on the key agreement and key confirmation components, other supporting security functions are not thoroughly tested by the testing in NIST SP 800-56A. The validation testing for these supporting security functions are found in the validation test suite for this specific function. Therefore, these supporting security functions **shall** be validated as a prerequisite to NIST SP 800-56A vendor affirmation.

   To claim vendor affirmation to NIST SP 800-56A, the underlying security functions used by this IUT **shall** be tested and validated prior to claiming vendor affirmation. These include:

   - Supported hash algorithms (SHA1, SHA224, SHA256, SHA384, and/or SHA512)
   - Supported Message Authentication Code (MAC) algorithms (CMAC, CCM, and/or HMAC)
   - Supported Random Number Generators (RNG)
   - If Finite Field Cryptography (FFC) is supported,
     - If the IUT generates domain parameters the DSA PQG generation and/or verification tests.
     - If the IUT generates key pairs, the DSA key pair generation tests.
   - If Elliptic Curve Cryptography (ECC) is supported,
     - If the IUT generates key pairs, the ECDSA key pair generation test and/or the Public Key Validation (PKV) test.

3. The SP 800-56 self tests required in cryptographic module implementations must consist of a known answer test that validates the correctness of the implemented DLC primitives and key derivation functions for each key agreement scheme implemented.

Annotation

Refer to IG G.13 for annotation examples.

Derived Test Requirements

Upon the following successful review, the CST Lab **shall** affirm by annotating the algorithm entry per the IG G.13 annotation requirements.

Required Vendor Information

The vendor **shall** provide evidence that their implementation implements the sections outlined above completely and accurately. This **shall** be accomplished by documentation and code review.

Required Test Procedures

The tester **shall** review the vendor's evidence demonstrating that their implementation conforms to the specifications specified above. This **shall** be accomplished by documentation and code review. The tester **shall** verify the rationale provided by the vendor.

D.2 Acceptable Key Establishment Protocols

Applicable Levels:	*All*
Original Publishing Date:	*02/10/2004*
Effective Date:	*02/10/2004*

Last Modified Date:	*10/20/2009*
Relevant Assertions:	*AS07.21*
Relevant Test Requirements:	*TE07.21.01*
Relevant Vendor Requirements:	*VE07.21.01-02*

Transition Status

Currently there is not a transition plan published for migration to NIST SP 800-56A, NIST SP 800-56B or symmetric key wrapping.

In lieu of a transition plan for **key agreement schemes**, there are currently five scenarios that are valid and allowed in an Approved FIPS mode of operation. The first four apply when a key is established (i.e. key agreement) and the fifth when only the DLC primitive is implemented (e.g. in a software toolkit):

1. CAVP KAS Certificate
2. Vendor Affirmation per IG D.1 – Transition for submitting CST Laboratory test reports ended March 24, 2009
3. non-Approved but allowed per this IG (DLC primitive as defined in SP 800-56A with a KDF specified in this IG)
4. non-Approved but allowed legacy implementation
5. non-Approved DLC primitive only from SP 800-56A.

In lieu of a transition plan for **key transport**, there are currently four scenarios that are valid and allowed in an Approved FIPS mode of operation and describe within this IG.

FIPS 140-2 certificate annotation examples for the above can be found in IG G.13.

Background

Cryptographic modules may use various symmetric and asymmetric key establishment schemes within protocols to establish and maintain secure communication links between modules. FIPS 140-2 Annex D provides a list of the Approved key establishment techniques for establishing keying material that are applicable to FIPS 140-2.

Question/Problem

FIPS 140-2 Annex D states that SP 800-56A provides approved asymmetric key establishment schemes to establish keying material. Annex D also states that additional symmetric and asymmetric key establishment schemes are allowed in a FIPS Approved mode of operation. What are these additional schemes?

Resolution

Key establishment is the process by which secret keying material is securely established between two or more entities. Keying material is data that is necessary to establish and maintain a cryptographic keying relationship[26]. Secret keying material includes keys, secret initialization vectors and other secret information. Symmetric and asymmetric key establishment may be accomplished using either key agreement or key transport schemes.

Key agreement is a method of key establishment where the resulting keying material is a function of information contributed by two or more participants, so that no party can predetermine the value of the secret keying material independently from the contribution of any other party. Key agreement is performed using key agreement schemes. At this time, NIST has specified key agreement schemes in SP 800-56A using Discrete Logarithm Cryptography (DLC). Key agreement schemes for Integer Factorization Cryptography (e.g., RSA)

[26] The state existing between two entities in which they share at least one cryptographic key.

will be specified in a subsequent document. Each scheme in SP 800-56A consists of several elements:

- A primitive (i.e., an algorithm) that is used to generate a shared secret from the public and/or private keys of the initiator and responder in a key agreement transaction. The shared secret is an intermediate value that is used as input to a key derivation function.
- A key derivation function (KDF) that uses the shared secret and other information to derive keying material[27].
- An optional message authentication code (MAC) that is used for key confirmation or implementation validation. Key confirmation is a procedure that provides assurance to one party (the key confirmation recipient) that another party (the key confirmation provider) actually possesses the correct secret keying material and/or shared secret.
- The rules for using the scheme securely. The rules specified in SP 800-56A include criteria for generating the domain parameters and asymmetric key pairs used during key agreement, methods for obtaining the required assurances, and specifications for performing key confirmation.

Several of the currently used implementations of DLC key agreement schemes do not comply with all requirements of SP 800-56A. In many cases, the KDF used to generate the keying material from the shared secret is different than a KDF specified in SP 800-56A.

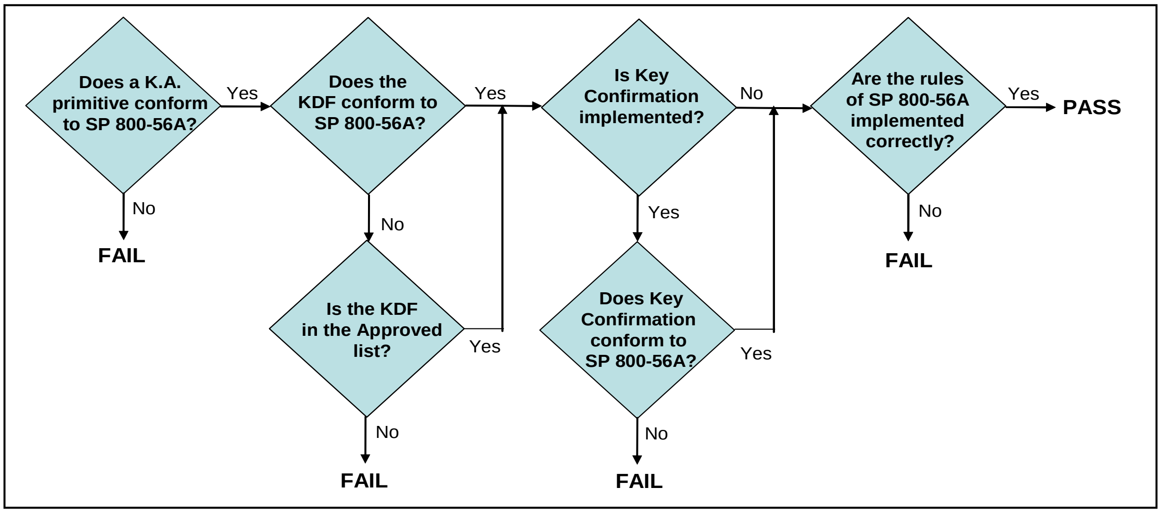

Does a K.A. primitive conform to SP 800-56A?
Yes
No
FAIL
Does the KDF conform to SP 800-56A?
Yes
No
Is the KDF in the Approved list?
Yes
No
FAIL
Is Key Confirmation implemented?
No
Yes
Does Key Confirmation conform to SP 800-56A?
Yes
No
FAIL
Are the rules of SP 800-56A implemented correctly?
Yes
PASS
No
FAIL

Figure 7.1-1: DLC Key Agreement Validation

Figure 7.1-1 depicts the DLC key agreement validation process. All implementations of DLC key agreement schemes to be submitted for FIPS 140-2 validation **shall** include:

1. One or more of the key agreement primitives specified in SP 800-56A. Domain parameters and key sizes **shall** conform to SP 800-56A.

2. KDFs **shall** conform to:

   - One of the KDFs in SP 800-56,
   - The KDF specified in IKEv2 (IETF RFC 4306), which is allowed only for the purpose of establishing keying material for security associations managed by IKEv2. The PRF used in IKEv2 **shall** employ the HMAC as specified in FIPS 198 (based on an Approved hash function).

[27] The keying material may be used directly (e.g., as a key), or the keying material may be used to derive (other) keys. The use of the keying material is outside the scope of SP 800-56A.

- Until December 31, 2010, **shall** conform to one of the following:

  a. One of the KDFs specified in American National Standard (ANS) X9.42-2001, *Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography*. An example of a protocol that uses ANS X9.42 is specified in RFC 2631, *Diffie-Hellman Key Agreement Method.* For the KDFs specified in ANS X9.42:

     1) The *OtherInfo* field of the key derivation function **should** be defined and used as specified in SP 800-56.

     2) The *counter* in the ASN.1 key derivation function **should** be represented as a 32-bit, big-endian bit string.

  b. The KDFs specified in American National Standard (ANS) X9.63-2001, *Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography*. An example of a protocol that uses ANS X9.63 is specified in RFC 3278, *Use of Elliptic Curve Cryptography* (*ECC*) *Algorithms in Cryptographic Message Syntax* (*CMS*). For the KDFs specified in ANS X9.63, the *OtherInfo* field of the key derivation function **should** be defined and used as specified in SP 800-56.

  c. The KDF specified in IKEv1 (IETF RFC 2409) is allowed only for the purpose of establishing keying material for security associations managed by IKEv1. The PRF used in IKEv1 **shall** employ the HMAC specified in FIPS 198 (based on an Approved hash function).

  d. The KDF specified in SSH (IETF RFC 4253) is allowed only for the purpose of establishing SSH sessions, and

  e. The KDF in TLS or DTLS is allowed only for the purpose of establishing keying material (in particular, the *master secret*) for a TLS or DTLS session with the following restrictions, even though the use of the SHA-1 and MD5 hash functions are not consistent with in Table 1 or Table 2 of SP 800-56A:

     1) The use of MD5 is allowed in the TLS or DTLS protocol only; MD5 **shall** not be used as a general hash function.

     2) The maximum number of blocks of secret keying material that can be produced by repeated use of the pseudorandom function during a single call to the TLS or DTLS key derivation function **shall** be 2^{32}-1.

3. If key confirmation is claimed for a key agreement scheme, one or more of the key confirmation methods in SP 800-56A **shall** be used.

4. An implementation **shall** conform to the key agreement rules specified in SP 800-56A, with the possible exception of the format of the KDF (see above).

Key transport is a method of key establishment whereby one party (the sender) selects a value for the secret keying material and then securely distributes that value to another party (the receiver). Key transport may be accomplished by:

- Wrapping the key using a secret symmetric key and a symmetric encryption algorithm. Key transport by wrapping a key obtained from DLC-based key agreement is addressed in SP 800-56A.
- Wrapping the key using the GDOI Group Key Management Protocol described in the IETF RFC 3547.

- In the absence of applicable standards, the CMVP will allow the wrapping of keys to be transported using the AES or the Triple DES symmetric key algorithms. The wrapping **should** be performed in compliance with AES Key Wrap Specification (Draft), published by the National Institute of Standards and Technology on 16 November 2001. If the Triple DES is used, then it **should** be used in exactly the same way that is defined for AES in the aforementioned draft standard. Both the 2-key and the 3-key Triple DES can be used for key wrapping.

  The symmetric key algorithm used for key wrapping **shall** be tested, even if the algorithm is not otherwise used by the cryptographic module, and the algorithm's certificate number **shall** be shown on the module's certificate. The use of this algorithm for key wrapping **shall** be documented on the non-Approved line of the cryptographic module's certificate. If the security strength of the key wrapping key and algorithm combination can be lower than that of the (potential) security strength of the wrapped key, then the resulting security strength of the wrapped key is the security strength of the key wrapping key and algorithm, and **should** be shown on the module's certificate in accordance with IG G.13.

- Encrypting the keying material using a public key and an asymmetric algorithm. The methods in NIST SP 800-56B **should** be used.

The allowable key sizes for key transport are specified in SP 800-57, Recommendation for Key Management, Part 1.

Any key transport scheme using an RSA-based key transport methodology that uses the allowable key sizes specified in SP 800-57 is acceptable until NIST provides further guidance.

Key transport schemes in the following protocols using asymmetric algorithms will be allowed for validation in FIPS mode to establish keying material until such time as Approved key transport schemes are determined:

1. The key transport scheme in SSL v3.1 is acceptable for use in the FIPS mode.

2. The key transport schemes in TLS, DTLS, EAP-TLS, EAP-FAST and PEAP-TLS may be used in the FIPS mode. While the protocols use the same cryptographic algorithms as the versions of SSL prior to version 3.1, the manner in which the algorithms are used makes them acceptable to be used in FIPS mode.

The following key establishment methods are *unacceptable*:

- SSL: all versions of SSL, except SSL v3.1[28], are not to be used in the FIPS mode. The manner in which the method uses approved and non-approved cryptographic algorithms for its operation prohibits its usage.[29]

[28] SSL v3.1 is allowed, as it is equivalent to TLS v1.0.

[29] The problem with SSL 3.0 is the key derivation process that applies to all SSL 3.0 cipher suites: half of the master key that is set up during the SSL key exchange depends entirely on the MD5 hash function. MD5 is not a FIPS approved algorithm, and its collision resistance property has recently been broken by Antoine Joux.

TLS also uses MD5 in the key derivation process, but in a different manner, so that all of the master key depends on both MD5 and SHA-1, and nothing in TLS actually depends on MD5 for its security.

Therefore, TLS implementations can be validated under FIPS 140-2, while SSL 3.0 implementations cannot. TLS is version 3.1 of SSL, and most current servers and clients are capable of doing both SSL 3.0 and TLS.

William Burr, NIST Cryptographic Technology Group

- Password-Based Key Establishment Methods: all password-based key establishment methods such as PKCS#5 are not to be used in the FIPS mode.

The CMVP *may* allow other techniques and/or methods for use in a FIPS mode but they **shall** meet all the following requirements:

- are industry accepted;
- are commercially available;
- are widely used by government and industry; and
- are known in the public domain.

The **final determination** of an allowed method for use in an Approved FIPS mode is made by the NIST Cryptographic Technology Group. Please contact William Burr for review and determination of proposed methods. If allowed, the CST Laboratories test report submission **shall** include the affirmation correspondence as evidence for validation.

Additional Comments

This IG does not address key establishment for use in authentication techniques.

The key establishment method(s) used by the cryptographic module must be listed under AS.07.21.

D.3 Assurance of the Validity of a Public Key for Key Establishment

Applicable Levels:	*All*
Original Publishing Date:	*10/21/2009*
Effective Date:	*10/21/2009*
Last Modified Date:	*10/21/2009*
Relevant Assertions:	*AS07.17*
Relevant Test Requirements:	*TE07.17.01-02*
Relevant Vendor Requirements:	*VE07.17.01*

Background

The correct functioning of public key algorithms depends, in part, on the arithmetic validity of the public key.

Both the owner and the recipient of a public key need to obtain assurance of public key validity before using the key for operational purposes after key establishment. Public key algorithms for key establishment are specified in NIST Special Publication (SP) 800-56A and 800-56B. Methods for obtaining assurance of public key validity are provided in Section 5.6.2 of SP 800-56A, and in Section 6.4 of SP 800-56B.

The key establishment schemes in SP 800-56A are specified using either static (long term, multi-use) keys or ephemeral (short term, single use) keys or both. The keys used in the SP 800-56B schemes are generally long term (i.e., static) keys.

Since a static key is normally used for a relatively long period of time, and a number of methods are provided for obtaining assurance of public key validity either by the owner or recipient directly, or by using a trusted third party, the process of obtaining the assurance is not too onerous. However, methods for obtaining this assurance for ephemeral keys are more limited, since a trusted third party is normally not available for obtaining the required assurance. The owner of an ephemeral public key generates that key, and obtains assurance of ephemeral public key validity by virtue of generating the key as specified in SP 800-56A (see

Section 5.6.2.1; Note that this section applies to the owner assurances of both Static and Ephemeral public key validity). However, the recipient of an ephemeral public key must obtain the assurance by performing an explicit public key validation process.

Question/Problem

Public key validation requires a certain amount of time to perform, which can significantly affect communication performance. Can this process be omitted if at least some of the security goals (i.e., authentication of the public key owner and the integrity of the ephemeral key) are fulfilled by other means?

Resolution

The owner or a recipient of a static public key **shall** obtain assurance of the validity of that public key using one or more of the methods specified in SP 800-56A or SP 800-56B, as appropriate. The owner of an ephemeral public key **shall** obtain assurance of the validity of that key as specified in SP 800-56A. Explicit public key validation of an ephemeral public key is required as specified in SP 800-56A by a recipient, except in the following situation; in this case, explicit public key validation of the ephemeral public key by the recipient is optional:

1. The ephemeral public key was generated for use in an FFC dhEphem key agreement scheme or an ECC Ephemeral Unified Model key agreement scheme, and
2. The key agreement scheme is being conducted using a protocol that authenticates the source and the integrity of each received ephemeral public key by means of an approved security technique (e.g., a digital signature or an HMAC).

Protocols that satisfy #2 above and, therefore, may omit the explicit ephemeral public key validation process include:

- Internet Key Exchange (IKE) protocol,
- Internet Key Exchange protocol, version 2 (IKEv2),
- Transport Layer Security (TLS) protocol, versions 1.0, and
- Datagram Transport Layer Security (DTLS) protocol, version 1.0.

In this case, when explicit public key validation is not performed on the ephemeral public key by an implementation in the manner specified in SP 800-56A (and therefore is not tested by the CAVS), the cryptographic algorithm's validation will indicate that the capability to provide assurance of ephemeral public key validity is not required for algorithm validation, based on this IG. However, the cryptographic algorithm validation and the cryptographic module validation may still claim that the algorithm and module are otherwise compliant with SP 800-56A.

Additional Comments

CAVP

Example of the Description/Notes field of a SP800-56A algorithm validation entry where the explicit public key validation of an ephemeral public key is not required for algorithm validation based on this IG (and therefore is not tested by the CAVS):

ECC: (ASSURANCES <5.5.2 #3>

ASSURANCE 5.6.2.3: requirement is not required for algorithm validation, based on FIPS 140-2 IG 7.10)
SCHEMES [EphemeralUnified (KARole(s): Responder)
(EC: P-256   SHA256)]
SHS Val#650 DRBG Val#1

CMVP

If a cryptographic module includes a key agreement scheme whereby the recipient of an ephemeral public key omits the explicit public key validation, the modules Security Policy **shall** indicate the appropriate protocol listed above that allows the omission of the validation in order to claim conformance to this IG.

Expired Implementation Guidance

End of Document