The CCM Validation System (CCMVS)

Updated: January 9, 2012
March 30, 2006
November 29, 2004

Lawrence E. Bassham III

National Institute of Standards and Technology

Information Technology Laboratory

Computer Security Division

TABLE OF CONTENTS

Update Log

1/9/12

Section 6.6 - The Decryption-Verification Process Test, Subsection A.3 removed the word payload as one of the inputs to this test. Payload is not an input. It is an output when the test returns valid.

1 Introduction

This document, *The CCM Validation System (CCMVS)* specifies the procedures involved in validating implementations of the CCM Mode of Operation as specified in SP 800-38C, *Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality* [1]. The CCMVS is designed to perform automated testing on Implementations Under Test (IUTs). This document provides the basic design and configuration of the CCMVS.

This document defines the purpose, the design philosophy, and the high-level description of the validation process for CCM. The requirements and administrative procedures to be followed by those seeking formal validation of an implementation of CCM are presented. The requirements described include a specification of the data communicated between the IUT and the CCMVS, the details of the tests that the IUT must pass for formal validation, and general instruction for interfacing with the CCMVS. Additionally, an appendix is also provided containing samples of input and output files for the CCMVS.

A set of CCM test vectors is available on the http://csrc.nist.gov/cryptval/ website for testing purposes.

2 Scope

This document specifies the tests required to validate IUTs for conformance to CCM specified in [1]. When applied to an IUT, the CCMVS provides testing to determine the correctness of the implementation of CCM. The CCMVS is composed of five separate tests – four to test various aspects involved in Encryption-Generation process and one to test the Decryption-Verification process. In addition to performing the tests specified in CCMVS, the IUT must undergo testing of the underlying encryption algorithm, namely AES, implementation via the appropriate validation suite (AESVS).

3 Conformance

The successful completion of the tests contained within the CCMVS and the AESVS is required to be validated as conforming to the CCM standard. Testing for the cryptographic module in which the CCM is implemented is defined in FIPS PUB 140-2, *Security Requirements for Cryptographic Modules*.[2]

4 Definitions and Abbreviations

4.1 Definitions

DEFINITION	**MEANING**
Advanced Encryption	The algorithm specified in FIPS 197, *Advanced Encryption Standard*

Standard	*(AES)*
CMT laboratory	Cryptographic Module Testing laboratory that operates the CCMVS

4.2 Abbreviations

ABBREVIATION	**MEANING**
AES	Advanced Encryption Standard specified in FIPS 197
AESVS	Advanced Encryption Standard Validation System
FIPS	Federal Information Processing Standard
CCM	CCM Mode of Operation specified in SP 800-38C
IUT	Implementation Under Test

5 Design Philosophy of CCM Validation System

The CCMVS is designed to test conformance to the CCM specification rather than provide a measure of a product's security. The validation tests are designed to assist in the detection of accidental implementation errors, and are not designed to detect intentional attempts to misrepresent conformance. Thus, validation should not be interpreted as an evaluation or endorsement of overall product security.

The CCMVS has the following design philosophy:

1. The CCMVS is designed to allow the testing of an IUT at locations remote to the CCMVS. The CCMVS and the IUT communicate data via *REQUEST* and *RESPONSE* files. The CCMVS also generates *SAMPLE* files to provide the IUT with a sample of what the *RESPONSE* file should look like.

2. The testing performed within the CCMVS utilizes statistical sampling (i.e., only a small number of the possible cases are tested); hence, the successful validation of a device does not imply 100% conformance with the standard.

6 CCMVS Test

The CCMVS tests the implementation of CCM for its conformance to the CCM standard. The testing for CCM consists of five tests. These tests are:

- Variable Associated Data Test;

- Variable Payload Test;
- Variable Nonce Test;
- Variable Tag Test; and
- Decryption-Verification Process Test.

6.1 Configuration Information

To initiate the validation process of the CCMVS, a vendor submits an application to an accredited laboratory requesting the validation of its implementation of CCM. The vendor's implementation is referred to as the Implementation Under Test (IUT). The request for validation includes background information describing the IUT along with information needed by the CCMVS to perform the specific tests. More specifically, the request for validation includes:

1. Vendor Name;
2. Product Name;
3. Product Version;
4. Implementation in software, firmware, or hardware;
5. Processor and Operating System with which the IUT was tested if the IUT is implemented in software or firmware;
6. Brief description of the IUT or the product/product family in which the IUT is implemented by the vendor (2-3 sentences); and
7. Configuration information for the CCM tests, including:

    a) Which AES key sizes are supported: 128, 192, and/or 256;

    b) Specify a minimum (greater than or equal to zero bytes) and maximum (less than or equal to 32 bytes) associated data length. Additionally, can the implementation handle an associated data length of 2^16 bytes;

    c) Specify a minimum (greater than or equal to zero bytes) and maximum (less than or equal to 32 bytes) payload length;

    d) Specify the nonce lengths supported: 7, 8, 9, 10, 11, 12, and/or 13; and

    e) Specify the tag lengths supported: 4, 6, 8, 10, 12, 14, and/or 16.

6.2 The Variable Associated Data Test

For each associated data length supported, the Variable Associated Data Test provides 10 sets of associated data and payload to the IUT. The IUT generates a ciphertext as specified by CCM using the data provided. The CCMVS verifies the correctness of the ciphertext produced by the IUT.

The CCMVS:

A. Creates a *REQUEST* file (Filename: VADT{KeySize}.req) containing:

   1. The Product Name;

   2. The algorithm being tested; and

   3. The keys, nonce, associated data, and payload values to be used as input to the CCM algorithm.

   Note: The CMT laboratory sends the *REQUEST* file to the IUT.

B. Creates a *FAX* file (Filename: VADT{KeySize}.fax) containing:

   1. The information from the *REQUEST* file; and

   2. The CT generated by the CCM algorithm.

   Note: The CMT laboratory retains the *FAX* file.

The IUT:

A. Generates the requested CTs using the data specified in the *REQUEST* file.

B. Creates a *RESPONSE* file (Filename: VADT{KeySize}.rsp) containing:

   1. The information from the *REQUEST* file; and

   2. The CT generated by the CCM algorithm.

   Note: The IUT sends the *RESPONSE* file to the CMT laboratory for processing by the CCMVS.

The CCMVS:

A. Compares the contents of the *RESPONSE* file with the contents of the *FAX* file.

B. If all values match, records PASS for this test; otherwise, records FAIL.

6.3 The Variable Payload Test

For each payload length supported the Variable Payload Test provides 10 sets of associated data and payload to the IUT. The IUT generates a ciphertext as specified by CCM using the data provided. The CCMVS verifies the correctness of the ciphertext produced by the IUT.

The CCMVS:

A. Creates a *REQUEST* file (Filename: VPT{KeySize}.req) containing:

1. The Product Name;
2. The algorithm being tested; and
3. The keys, nonce, associated data, and payload values to be used as input to the CCM algorithm.

Note: The CMT laboratory sends the *REQUEST* file to the IUT.

B. Creates a *FAX* file (Filename: VPT{KeySize}.fax) containing:

1. The information from the *REQUEST* file; and
2. The CT generated by the CCM algorithm.

Note: The CMT laboratory retains the *FAX* file.

The IUT:

A. Generates the requested CTs using the data specified in the *REQUEST* file.

B. Creates a *RESPONSE* file (Filename: VPT{KeySize}.rsp) containing:

1. The information from the *REQUEST* file; and
2. The CT generated by the CCM algorithm.

Note: The IUT sends the *RESPONSE* file to the CMT laboratory for processing by the CCMVS.

The CCMVS:

A. Compares the contents of the *RESPONSE* file with the contents of the *FAX* file.

B. If all values match, records PASS for this test; otherwise, records FAIL.

6.4 The Variable Nonce Test

For each nonce length supported the Variable Nonce Test provides 10 sets of associated data and payload to the IUT. The IUT generates a ciphertext as specified by CCM using the data provided. The CCMVS verifies the correctness of the ciphertext produced by the IUT.

The CCMVS:

A. Creates a *REQUEST* file (Filename: VNT{KeySize}.req) containing:

1. The Product Name;
2. The algorithm being tested; and
3. The keys, nonce, associated data, and payload values to be used as input to the CCM algorithm.

Note: The CMT laboratory sends the *REQUEST* file to the IUT.

B. Creates a *FAX* file (Filename: VNT{KeySize}.fax) containing:

1. The information from the *REQUEST* file; and

2. The CT generated by the CCM algorithm.

Note: The CMT laboratory retains the *FAX* file.

The IUT:

A. Generates the requested CTs using the data specified in the *REQUEST* file.

B. Creates a *RESPONSE* file (Filename: VNT{KeySize}.rsp) containing:

1. The information from the *REQUEST* file; and
2. The CT generated by the CCM algorithm.

Note: The IUT sends the *RESPONSE* file to the CMT laboratory for processing by the CCMVS.

The CCMVS:

A. Compares the contents of the *RESPONSE* file with the contents of the *FAX* file.

B. If all values match, records PASS for this test; otherwise, records FAIL.

6.5 The Variable Tag Test

For each tag length supported the Variable Tag Test provides 10 sets of associated data and payload to the IUT. The IUT generates a ciphertext as specified by CCM using the data provided. The CCMVS verifies the correctness of the ciphertext produced by the IUT.

The CCMVS:

A. Creates a *REQUEST* file (Filename: VTT{KeySize}.req) containing:

1. The Product Name;
2. The algorithm being tested; and
3. The keys, nonce, associated data, and payload values to be used as input to the CCM algorithm.

Note: The CMT laboratory sends the *REQUEST* file to the IUT.

B. Creates a *FAX* file (Filename: VTT{KeySize}.fax) containing:

1. The information from the *REQUEST* file; and
2. The CT generated by the CCM algorithm.

Note: The CMT laboratory retains the *FAX* file.

The IUT:

A. Generates the requested CTs using the data specified in the *REQUEST* file.

B. Creates a *RESPONSE* file (Filename: VTT{KeySize}.rsp) containing:

1. The information from the *REQUEST* file; and

2. The CT generated by the CCM algorithm.

Note: The IUT sends the *RESPONSE* file to the CMT laboratory for processing by the CCMVS.

The CCMVS:

A. Compares the contents of the *RESPONSE* file with the contents of the *FAX* file.

B. If all values match, records PASS for this test; otherwise, records FAIL.

6.6 The Decryption-Verification Process Test

For each combination of associated data length, payload length, nonce length, and tag length provided as input, 15 sets of input plus ciphertext are supplied to the IUT. The IUT uses the data provided to determine if the ciphertext passes or fails the verification process.

The CCMVS:

A. Creates a *REQUEST* file (Filename: DVPT{KeySize}.req) containing:

1. The Product Name;
2. The algorithm being tested; and
3. The keys, nonce, associated data, and ciphertext values to be used as input to the decryption-verification process of the CCM algorithm.

Note: The CMT laboratory sends the *REQUEST* file to the IUT.

B. Alter some of the ciphertext produced to ensure the decryption-verification process fails.

C. Creates a *FAX* file (Filename: DVPT{KeySize}.fax) containing:

1. The information from the *REQUEST* file; and
2. An indication of whether or not the ciphertext passes the decryption-verification process.
3. If the verification passed, list the payload generated by the CCM algorithm.

Note: The CMT laboratory retains the *FAX* file.

The IUT:

A. Performs the decryption-verification process to determine whether the data sets verify correctly or not.

B. Creates a *RESPONSE* file (Filename: DVPT{KeySize}.rsp) containing:

1. The information from the *REQUEST* file; and
2. Whether or not the decryption-verification process passed or failed.
3. If the verification passed, list the payload generated by the CCM algorithm.

Note: The IUT sends the *RESPONSE* file to the CMT laboratory for processing by the CCMVS.

The CCMVS:

A. Compares the contents of the *RESPONSE* file with the contents of the *FAX* file.

B. If all values match, records PASS for this test; otherwise, records FAIL.

Appendix A References

[1] *Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality*, Special Publication 800-38C, National Institute of Standards and Technology, May 2004.

[2] *Security Requirements for Cryptographic Modules*, FIPS Publication 140-2, National Institute of Standards and Technology, May 2001.