The CMAC Validation System (CMACVS)

March 30, 2006

Sharon S. Keller

National Institute of Standards and Technology

Information Technology Laboratory

Computer Security Division

TABLE OF CONTENTS

1 INTRODUCTION **1**

2 SCOPE **1**

3 CONFORMANCE **1**

4 DEFINITIONS AND ABBREVIATIONS **2**

4.1 DEFINITIONS 2
4.2 ABBREVIATIONS 2

5 DESIGN PHILOSOPHY OF CMAC VALIDATION SYSTEM **2**

6 CMACVS TEST **3**

6.1 CONFIGURATION INFORMATION 3
6.2 THE CMAC GENERATION TEST 4
6.3 THE CMAC VERIFICATION PROCESS TEST 5

APPENDIX A REFERENCES **7**

1 Introduction

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

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

A set of CMAC 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 the CMAC algorithm as specified in [1]. When applied to an IUT, the CMACVS provides testing to determine the correctness of the implementation of CMAC. The CMACVS is composed of two separate tests - the MAC Generation function and the MAC Verification function. These validations tests verify the functionality of the generation and verification functions of the CMAC algorithm as well as the functionality of the SubKey generation function.

The CMAC algorithm validation process requires additional prerequisite testing of the underlying encryption algorithm implementation via the appropriate validation suite; that is, the AES algorithm must be validated via the AESVS and/or the TDES algorithm must be validated via the SP800-20 and the Multi-block Message Text (MMT) tests.

3 Conformance

The successful completion of the tests contained within the CMACVS and the AESVS and/or SP800-20, *Modes of Operation Validation System for the Triple Data Encryption Algorithm (TMOVS): Requirements and Procedures.*[2] and the Multi-block Message Text (MMT) tests [3] is required to be validated as conforming to the CMAC standard. Testing for the cryptographic module in which the CMAC is implemented is defined in FIPS PUB 140-2, *Security Requirements for Cryptographic Modules.*[4]

4 Definitions and Abbreviations

4.1 Definitions

DEFINITION	MEANING
Advanced Encryption Standard	The algorithm specified in FIPS 197, *Advanced Encryption Standard (AES)*
Triple Data Encryption Standard	The algorithm specified in FIPS 46-3, *Data Encryption Standard (DES)*
CMT laboratory	Cryptographic Module Testing laboratory that operates the CMACVS

4.2 Abbreviations

ABBREVIATION	MEANING
AES	Advanced Encryption Standard specified in FIPS 197
AESVS	Advanced Encryption Standard Validation System
FIPS	Federal Information Processing Standard
CMAC	CMAC Mode of Operation specified in SP 800-38B
IUT	Implementation Under Test
TDES	Triple Data Encryption Standard specified in FIPS 46-3

5 Design Philosophy of CMAC Validation System

The CMACVS is designed to test conformance to the CMAC 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 CMACVS has the following design philosophy:

1. The CMACVS is designed to allow the testing of an IUT at locations remote to the CMACVS. The CMACVS and the IUT communicate data via *REQUEST* and

RESPONSE files. The CMACVS 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 CMACVS 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 CMACVS Test

The CMACVS tests the implementation of CMAC for its conformance to the CMAC standard. The testing for CMAC consists of two tests. These tests are:

- MAC Generation Test; and
- MAC Verification Process Test.

6.1 Configuration Information

To initiate the validation process of the CMACVS, a vendor submits an application to an accredited laboratory requesting the validation of its implementation of CMAC. 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 CMACVS 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 CMAC tests, including:

   a) Underlying algorithm(s) supported – AES and/or TDES;

   b) For each underlying algorithm – key size combination:

      - Specify whether or not the implementation supports zero length messages;

- Specify two message lengths divisible by the block size. If the underlying algorithm is AES, the block size is 16 bytes. If the underlying algorithm is TDES, the block size is 8 bytes.

- Specify two message lengths not divisible by the block size.

- Specify the largest message size supported by the implementation or check the 2^16 box, which ever is larger.

- Specify a minimum, middle, and maximum CMAC length supported by the implementation. If the underlying algorithm is AES, the maximum CMAC length is 16 bytes. If the underlying algorithm is TDES, the maximum CMAC length is 8 bytes. The minimum length is 1 byte.

6.2 The CMAC Generation Test

For each combination of key length, message length, MAC length, the CMAC Generation Test provides a set of 8 key-message combinations to the IUT. The IUT generates a MAC as specified by CMAC using the data provided. The CMACVS verifies the correctness of the MAC produced by the IUT. Note that this test also tests SubKey generation.

The CMACVS:

A. Creates a *REQUEST* file (Filename: CMACGen{Supported Algorithm Name}{KeySize}.req) containing:

   1. The Product Name;
   2. The algorithm being tested; and
   3. The length of the key(s), message and MAC, the key value(s) and the message value to be used as input to the CMAC algorithm.

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

B. Creates a *FAX* file (Filename: CMACGen{Supported Algorithm Name}{KeySize}.fax) containing:

   1. The information from the *REQUEST* file; and
   2. The MAC generated by the CMAC algorithm.

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

The IUT:

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

B. Creates a *RESPONSE* file (Filename: CMACGen{Supported Algorithm Name}{KeySize}.rsp) containing:

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

2. The MAC generated by the CMAC algorithm.

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

The CMACVS:

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 CMAC Verification Process Test

For each combination of key length, message length, MAC length, the CMAC Verification Process Test provides a set of 20 key-message-MAC combinations to the IUT. The IUT uses the data provided to determine if the MAC passes or fails the verification process. Note that this test also tests SubKey generation.

The CMACVS:

A. Creates a *REQUEST* file (Filename: CMACVer{Supported Algorithm Name}{KeySize}.req) containing:

1. The Product Name;
2. The algorithm being tested; and
3. The length of the key(s), message and MAC, the key value(s) and the message value to be used as input to the CMAC verification process of the CMAC algorithm.

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

B. Alter some of the MACs produced to ensure the CMAC verification process fails

C. Creates a *FAX* file (Filename: CMACVer{Supported Algorithm Name} {KeySize}.fax) containing:

1. The information from the *REQUEST* file; and
2. An indication of whether or not the MAC passes the CMAC verification process.

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

The IUT:

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

B. Creates a *RESPONSE* file (Filename: CMACVer{Supported Algorithm Name}{KeySize}.rsp) containing:

1. The information from the *REQUEST* file; and
2. Whether or not the CMAC verification process passed or failed.

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

The CMACVS:

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 CMAC Mode for Authentication*, Special Publication 800-38B, National Institute of Standards and Technology, May 2004.

[2] *Modes of Operation Validation System for the Triple Data Encryption Algorithm (TMOVS): Requirements and Procedures,* Special Publication 800-20, National Institute of Standards and Technology, April 2000.

[3] *The Multi-block Message Test (MMT) for DES and TDES,* National Institute of Standards and Technology.

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