The Elliptic Curve Cryptography Cofactor Diffie-Hellman (ECC CDH) Primitive

Validation System (ECC_CDHVS)

April 12, 2011

Sharon S. Keller

National Institute of Standards and Technology

Information Technology Laboratory

Computer Security Division

TABLE OF CONTENTS

1 Introduction

This document, *The Elliptic Curve Cryptography Cofactor Diffie-Hellman (ECC CDH) Primitive Validation System (ECC_CDHVS)*, specifies the procedures involved in validating the Elliptic Curve Cryptography Cofactor Diffie-Hellman (ECC CDH) Primitive which is a component of SP 800-56A*, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography* [1] described in Section 5.7.1.2. The ECC CDH primitive is a Discrete Logarithm Cryptography (DLC) primitive that is used to compute the shared secret value *Z*. The ECC_CDHVS is designed to perform automated testing on Implementations Under Test (IUTs).

This document defines the purpose, the design philosophy, and the high-level description of the validation process for the ECC CDH primitive. The requirements and administrative procedures to be followed by those seeking formal validation of an implementation of the ECC CDH primitive are presented. The requirements described include a specification of the data communicated between the IUT and the ECC_CDHVS, the details of the tests that the IUT must pass for formal validation, and general instruction for interfacing with the ECC_CDHVS.

A set of ECC_CDH primitive 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 implementations of the ECC CDH primitive for conformance to the specifications in Section 5.7.1.2 of SP 800-56A. When applied to an Implementation Under Test (IUT), the ECC_CDHVS provides testing to determine the correctness of the implementation.

3 Conformance

The successful completion of the tests contained within the ECC_CDHVS is required to claim conformance to SP800-56A Section 5.7.1.2. Testing for the cryptographic module in which the ECC CDH Primitive is implemented is defined in FIPS PUB 140-2, *Security Requirements for Cryptographic Modules*.[2]

4 Definitions and Abbreviations

4.1 Definitions

DEFINITION	**MEANING**
CST laboratory	Cryptographic and Security Testing (CST) laboratories that operates the ECC_CDHVS

4.2 Abbreviations

ABBREVIATION	MEANING
CDH	Cofactor Diffie-Hellman
DLC	Discrete Logarithm Cryptography
ECC	Elliptic Curve Cryptography
ECC CDH	Elliptic Curve Cryptography Cofactor Diffie-Hellman
FIPS	Federal Information Processing Standard
HMACVS	HMAC Validation System
IUT	Implementation Under Test
KAS	Key Agreement Scheme
Z	A shared secret that is used to derive secret keying material using a key derivation function; a DLC primitive – either Diffie-Hellman or MQV.

5 Design Philosophy of ECC CDH Primitive Validation System

The ECC_CDHVS is designed to test conformance to specifications in Section 5.7.1.2 of SP 800-56A 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 ECC_CDHVS has the following design philosophy:

1. The ECC_CDHVS is designed to allow the testing of an IUT at locations remote to the ECC_CDHVS. The ECC_CDHVS and the IUT communicate data via *REQUEST* and *RESPONSE* files. The ECC_CDHVS also generates *SAMPLE* files to provide the IUT with an example of the format required by the *RESPONSE* file.

2. The testing performed within the ECC_CDHVS 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 Recommendation.

6 ECC CDH Primitive Validation System (ECC_CDHVS) Test

When applied to an IUT, the ECC_CDHVS provides testing to determine the correctness of the implementation of the ECC CDH Primitive specifications. As detailed in the Recommendation, the validation test suite verifies that an IUT has implemented the ECC CDH primitive component of the key agreement scheme according to the specifications in the Recommendation therefore conforming to Section 5.7.1.2 of SP800-56A.

6.1 Configuration Information

To initiate the validation process of the ECC_CDHVS, a vendor submits an application to an accredited laboratory requesting the validation of its implementation of Section 5.7.1.2 ECC CDH Primitive component. The vendor's implementation is referred to as the IUT. The request for validation includes background information describing the IUT, along with information needed by the ECC_CDHVS to perform the specific tests. More specifically, the request for validation includes:

1. Cryptographic algorithm implementation information

    a. Vendor Name;

    b. Implementation Name;

    c. Implementation Version;

    d. Indication if implementation is software, firmware, or hardware;

    e. Processor and Operating System with which the IUT was tested if the IUT is implemented in software or firmware;

    f. Brief description of the IUT or the product/product family in which the IUT is implemented by the vendor (2-3 sentences); and

2. Configuration information for the ECC_CDHVS tests consists of

    a. The elliptic curves supported by the IUT

6.2 The KAS ECC CDH Primitive Test

A request, sample and fax file are created for this test. The name of these files is KAS_ECC_CDH_PrimitiveTest followed by the appropriate extension (.req, .sam, fax). Each file will contain a separate section for each curve tested by the IUT. Each section will be denoted with a header indicating the curve being tested, e.g., [P-256]. Within each curve section, the CAVS provides 25 sets of CAVS's public key pairs (QCAVSx and QCAVSy) to be used in the ECC CDH primitive computations. In addition to this information, the sample file also contains labels for information that is to be supplied by the IUT. This includes the IUT's public key pair (QIUTx = ?; QIUTy = ?) and the Z value generated by the IUT (ZIUT = ?).

The IUT uses the NIST-approved curves to obtain their own public/private key pair. The IUT uses the CAVS public keys (labeled QCAVSx and QCAVSy) and its own private key to calculate the shared secret value *Z* (labeled ZIUT). The *Z* value is computed using the specifications in Section 5.7.1.2, ECC CDH Primitive.

The IUT's public key pair (QIUTx; QIUTy), and the Z value (ZIUT) are stored in the *RESPONSE* file in the format specified in the *SAMPLE* file. There shall be a *RESPONSE* file for every *SAMPLE* file.

The ECC_CDHVS verifies the correctness of the IUT's shared secret value ZIUT by using the IUT's public key pair (QIUTx; QIUTy) and the CAVS private key to calculate the shared secret value, which, for purposes of this document, will be denoted as ZCAVS. The CAVS compares the value of the IUT's *Z* to the CAVS's *Z* to see if they are the same. If they are, then it can be determined that the ECC CDH Primitive has been implemented correctly according to the Recommendation. If the values do not match, the IUT has an error in it. During the validation of the IUT, if an error occurs, the values of *Z* generated by the CAVS are stored in the log file. The laboratory uses the information in the log file to assist the vendor in debugging their IUT.

Appendix A References

[1] *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography*, Special Publication 800-56A, National Institute of Standards and Technology, March 2006.

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