

Sterling Commerce

Sterling FIPS Crypto-J Module
Version 2.1

FIPS 140-2
Non-Proprietary Security Policy

Level 1 Validation

Document Version 1.0



4600 Lakehurst Court
Dublin, Ohio 43016-2000
Phone: +1 (614) 793-7000
FAX: +1 (614) 793-4040
<http://www.sterlingcommerce.com>

© 2007 Sterling Commerce

This document may be freely reproduced and distributed whole and intact including this copyright notice.

Table of Contents

1	INTRODUCTION	6
1.1	OVERVIEW.....	6
1.2	PURPOSE.....	6
1.3	REFERENCES.....	6
1.4	REVISION HISTORY.....	5
2	CRYPTOGRAPHIC MODULE SPECIFICATION	7
2.1	PHYSICAL SPECIFICATIONS.....	7
2.2	COMPUTER HARDWARE, OS, AND JVM	7
2.3	SOFTWARE SPECIFICATIONS	9
3	CRYPTOGRAPHIC MODULE PORTS AND INTERFACES	10
4	ROLES, SERVICES, AND AUTHENTICATION	11
4.1	ROLES	11
4.2	SERVICES.....	11
4.3	OPERATOR AUTHENTICATION	11
5	FINITE STATE MODEL	14
6	PHYSICAL SECURITY	15
7	OPERATIONAL ENVIRONMENT	16
8	CRYPTOGRAPHIC KEY MANAGEMENT	17
8.1	KEY GENERATION	17
8.2	8.2 KEY ESTABLISHMENT.....	17
8.3	8.3 KEY ENTRY AND OUTPUT.....	17
8.4	8.4 KEY STORAGE	17
8.5	8.5 ZEROIZATION OF KEYS	18
9	SELF-TESTS	19
9.1	POWER-UP TESTS	19
9.1.1	<i>Tests upon Power-up</i>	<i>19</i>
9.1.2	<i>On-Demand Self-Tests</i>	<i>19</i>
9.2	CONDITIONAL TESTS.....	19
9.3	FAILURE OF SELF-TESTS.....	19
10	DESIGN ASSURANCE.....	20
10.1	CONFIGURATION MANAGEMENT	20
10.2	DELIVERY AND OPERATION.....	20
10.3	DEVELOPMENT	20
10.4	GUIDANCE DOCUMENTS.....	20
11	MITIGATION OF OTHER ATTACKS	21
11.1	TIMING ATTACK ON RSA	21
11.2	ATTACK ON BIASED PRIVATE KEY OF DSA	21
APPENDIX A CRYPTO OFFICER AND USER GUIDE.....		22
A.1	INSTALLATION.....	22
A.1.1	<i>Installing.....</i>	<i>22</i>
A.1.2	<i>Uninstalling</i>	<i>22</i>
A.2	COMMANDS.....	22
A.2.1	<i>Initialization.....</i>	<i>22</i>
A.2.2	<i>Deinitialization</i>	<i>22</i>
A.2.3	<i>Self-Tests.....</i>	<i>22</i>
A.2.4	<i>Show Status.....</i>	<i>22</i>
A.3	WHEN MODULE IS DISABLED	22

Table of Figures

FIGURE 1 – CRYPTOGRAPHIC MODULE HARDWARE BLOCK DIAGRAM	8
FIGURE 2 – CRYPTOGRAPHIC MODULE SOFTWARE BLOCK DIAGRAM.....	9

Table of Tables

TABLE 1 – LOGICAL AND PHYSICAL INTERFACES.....	10
TABLE 2 – ROLES, SERVICES AND ACCESS	11
TABLE 3 – SUPPORTED ALGORITHMS AND STANDARDS	12

Revision History

Version	Modification Date	Modified By	Description of Changes
0.1	2007-11-07	stidmore	Initial draft
0.2	2007-11-13	stidmore	Correction of typos
0.3	2007-11-19	stidmore	Added Certicom copyright info
0.4	2007-11-20	stidmore	Corrected platforms to test on
1.0	2007-11-29	stidmore	Getting document ready to publish

Certicom Corporation has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. and non-U.S. patents listed at <http://www.certicom.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries. Information subject to change.

1 Introduction

1.1 Overview

This is a non-proprietary Federal Information Processing Standard (FIPS) 140-2 Security Policy for Sterling **FIPS Crypto-J Module Version 2.1** (FIPS Crypto-J Module). FIPS Crypto-J Module is a cryptographic toolkit for Java language users, providing services of various cryptographic algorithms such as hash algorithms, encryption schemes, message authentication, and public key cryptography. This Security Policy specifies the rules under which FIPS Crypto-J Module must operate. These security rules are derived from the requirements of FIPS 140-2 [1], and related documents [6, 7, 8].

1.2 Purpose

This Security Policy is created for the following purposes:

- It is required for FIPS 140-2 validation.
- To outline FIPS Crypto-J Module's conformance to FIPS 140-2 Level 1 Security Requirements.
- To provide users with how to configure and operate the cryptographic module in order to comply with FIPS 140-2.

1.3 References

[1] NIST *Security Requirements For Cryptographic Modules*, December 3, 2002.

[2] NIST *Security Requirements For Cryptographic Modules, Annex A: Approved Security Functions for FIPS PUB 140-2*, March 19, 2007.

[3] NIST *Security Requirements For Cryptographic Modules, Annex B: Approved Protection Profiles for FIPS PUB 140-2*, June 14, 2007.

[4] NIST *Security Requirements For Cryptographic Modules, Annex C: Approved Random Number Generators for FIPS PUB 140-2*, March 19, 2007.

[5] NIST *Security Requirements For Cryptographic Modules, Annex D: Approved Key Establishment Techniques for FIPS PUB 140-2*, June 26, 2007.

[6] NIST *Derived Test Requirements for FIPS 140-2*, Draft, March 24, 2004.

[7] NIST *Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program*, July 26, 2007.

[8] NIST *Frequently Asked Questions for the Cryptographic Module Validation Program*, June 6, 2007.

2 Cryptographic Module Specification

FIPS Crypto-J Module is a multiple-chip standalone cryptographic module, consisting of the following components:

- A commercially available general-purpose computer hardware.
- A commercially available Operating System (OS) that runs on the computer hardware.
- A commercially available JavaVirtualMachine (JVM) that runs on the computer hardware and OS.
- FIPS Crypto-J Module software that runs on the computer hardware, OS, and JVM.

2.1 Physical Specifications

The general-computer hardware component consists of the following devices:

1. CPU (Microprocessor)
2. Memory
 - a. Working memory is located on the RAM containing the following spaces:
 - i. Input/output buffer
 - ii. Plaintext/ciphertext buffer
 - iii. Control bufferKey storage is not deployed in this module.
 - b. Program memory is also located on RAM.
3. Hard Disk (or disks)
4. Display Controller
5. Keyboard Interface
6. Mouse Interface
7. Network Interface
8. Serial Port
9. Parallel Port
10. Power Supply

The configuration of this component is illustrated in Figure 1.

2.2 Computer Hardware, OS, and JVM

FIPS Crypto-J Module is tested on the following representative combinations of computer hardware and OS:

1. Solaris 10, SPARC 32-bit SPARC
2. Solaris 9, SPARC 32-bit SPARC
3. Solaris 9, SPARC 64-bit SPARC
4. Red Hat Linux AS 3.0, 32-bit x86 (Binary compatible to AS 2.1)
5. Red Hat Linux AS 4.0, 64-bit x86
6. Windows XP, 32-bit x86 (Binary compatible to Windows 98/2000/2003)

7. Windows XP, 64-bit x86

The Java Runtime Environment (JRE) 1.3.1, 1.4.2, and 1.5.0 by Sun Microsystems are used to test FIPS Crypto-J Module on the platforms above. The module will run on the equivalent JREs on various hardware and OS, while maintaining its compliance to the FIPS 140-2 Level 1 requirements. Such hardware and OS include the following:

- 8. Any other Solaris Platforms,
- 9. Any other Linux Platforms,
- 10. Any other Windows Platforms,
- 11. AIX Platforms, and
- 12. HP-UX Platforms.

Thus, this validation is applicable to these platforms as well

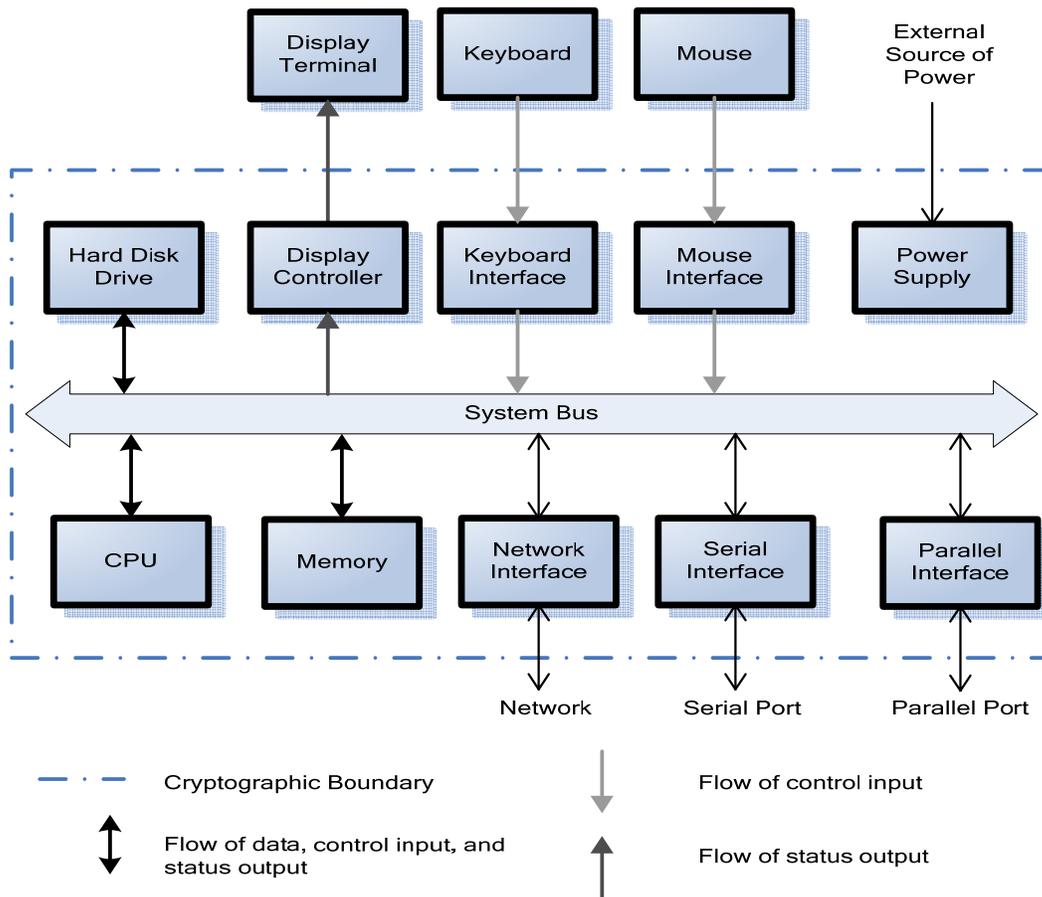


Figure 1 – Cryptographic Module Hardware Block Diagram

2.3 Software Specifications

FIPS Crypto-J Module software is manufactured by Certicom Corp., providing services to the Java computer language users in the form of a Java archive (JAR). The same binary is used for all identified computer hardware and OS because the JVM underneath FIPS Crypto-J Module will absorb the differences of the computer hardware and OS.

The interface into FIPS Crypto-J Module is via Application Programmer’s Interface (API) method calls. These method calls provide the interface to the cryptographic services, for which the parameters and return codes provide the control input and status output (see Figure 2).

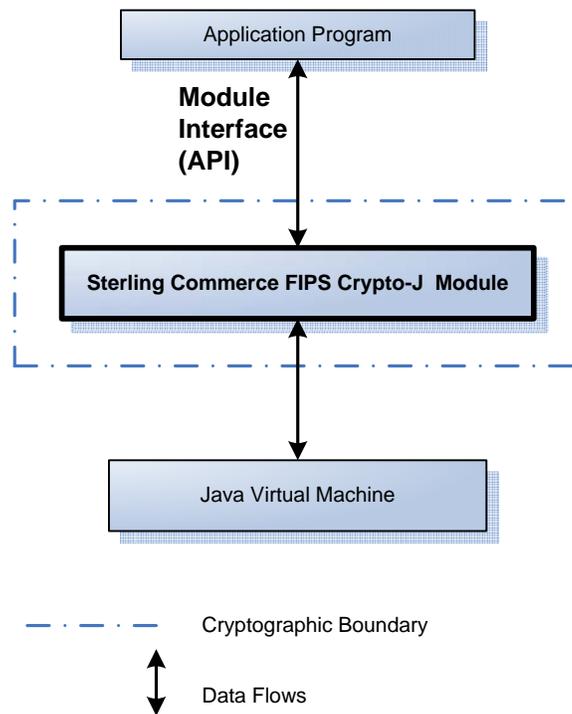


Figure 2 – Cryptographic Module Software Block Diagram

3 Cryptographic Module Ports and Interfaces

The physical and logical interfaces are summarized in Table 1.

Table 1 – Logical and Physical Interfaces

I/O	Logical Interface	Physical Interface
Data Input	API	Ethernet Port
Data Output	API	Ethernet Port
Control Input	API	Keyboard and Mouse
Status Output	Return Code	Display
Power Input	Initialization Function <code>FIPSManger.activateFIPSMode()</code>	The Power Supply is the power interface.

4 Roles, Services, and Authentication

4.1 Roles

FIPS Crypto-J Module supports Crypto Officer and User Roles, meeting FIPS 140-2 Level 1 requirements. These roles are enforced by this Security Policy. The Crypto Officer has the responsibility for installing FIPS Crypto-J Module (see Table 2).

In order to operate the module securely, it is the Crypto Officer and User's responsibility to confine calls to those methods that have been FIPS 140-2 Approved. Thus, in the approved mode of operation, all Roles shall confine themselves to calling FIPS Approved algorithms, as marked in Table 3.

4.2 Services

FIPS Crypto-J Module supports many cryptographic algorithms. The set of cryptographic algorithms supported by FIPS Crypto-J Module are given in Table 3.

The TDES, AES, SHS (SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512), HMAC-SHS (HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA256, HMAC-SHA-384, and HMAC-SHA-512), RNG, DSA, RSA PKCS #1, and ECDSA algorithms have been validated to comply with FIPS. FIPS Crypto-J Module also supports FIPS Approved key establishment techniques (key agreement and key wrapping), DH, ECDH, ECMQV, and RSA PKCS #1. In order to operate the module in compliance with FIPS, only these FIPS Approved algorithms should be used.

DES, DESX, ARC2, ARC4, MD5, MD2, and HMAC-MD5 are supported as non FIPS Approved algorithms. In order to operate the module in compliance with FIPS, these algorithms should not be used.

4.3 Operator Authentication

FIPS Crypto-J Module does not deploy an authentication mechanism. The roles of Crypto Officer and User are implicitly selected by the operator.

Table 2 – Roles, Services and Access

Service	Crypto Officer	User	Keys and CSPs	Access
Installation, etc.				
Installation	X			
Uninstallation	X			
Self-Tests	X	X	ECDSA public key	Use
Show Status	X	X		
Symmetric Ciphers				
Key Generation	X	X	AES, TDES keys	Create, Read
Encrypt	X	X	AES, TDES keys	Use
Decrypt	X	X	AES, TDES keys	Use
Hash Algorithms and Message Authentication				
Hashing	X	X		

Service	Crypto Officer	User	Keys and CSPs	Access
Msg. Authentication	X	X	HMAC keys	Use
Random Number Generation				
Seeding	X	X	Seed	Use
Request	X	X		
Digital Signature				
Key pair generation	X	X	RSA, DSA, ECDSA key pairs	Create, Read
Sign	X	X	RSA, DSA, ECDSA private keys	Use
Verify	X	X	RSA, DSA, ECDSA public keys	Use
Key Agreement				
Key pair generation	X	X	ECDH, ECMQV, DH key pairs	Create, Read
Shared secret generation	X	X	ECDH, ECMQV, DH key pairs	Use
Key Wrapping				
Key pair generation	X	X	RSA key pairs	Create, Read
Wrap	X	X	RSA public keys	Use
Unwrap	X	X	RSA private keys	Use

Table 3 – Supported Algorithms and Standards

	Algorithm	FIPS Approved or Allowed	Cert Number
Block Ciphers	DES (ECB, CBC, CFB64, OFB64)		
	TDES (ECB, CBC, CFB64, OFB64) [FIPS 46-3]	Yes	#485
	DESX (ECB, CBC, CFB64, OFB64)		
	AES (ECB, CBC, CFB128, OFB128, CTR, CCM) [FIPS 197]	Yes	#469
	ARC2 (ECB, CBC, CFB64, OFB64) [RFC 2268]		
Stream Cipher	ARC4		
Hash Functions	SHA-1 [FIPS 180-2]	Yes	#537
	SHA-224 [FIPS 180-2]	Yes	#537
	SHA-256 [FIPS 180-2]	Yes	#537
	SHA-384 [FIPS 180-2]	Yes	#537
	SHA-512 [FIPS 180-2]	Yes	#537
	MD5 [RFC 1321]		
	MD2 [RFC 1115]		
Message Authentication	HMAC-SHA-1 [FIPS 198]	Yes	#227
	HMAC-SHA-224 [FIPS 198]	Yes	#227

	Algorithm	FIPS Approved or Allowed	Cert Number
	HMAC-SHA-256 [FIPS 198]	Yes	#227
	HMAC-SHA-384 [FIPS 198]	Yes	#227
	HMAC-SHA-512 [FIPS 198]	Yes	#227
	HMAC-MD5 [RFC 2104]		
RNG	ANSI X9.62 RNG [ANSI X9.62]	Yes	#254
Digital Signature	DSS [FIPS 186-2]	Yes	#193
	ECDSA [FIPS 186-2, ANSI X9.62]	Yes	#41
	RSA PKCS1-v1_5 [PKCS #1 v2.1]	Yes	#191
Key Agreement	DH [ANSI X9.42]	Yes	
	ECDH [ANSI X9.63]	Yes	
	ECMQV [ANSI X9.63]	Yes	
Key Wrapping	RSA PKCS1-v1_5 [PKCS #1 v2.1]	Yes	
	RSA OAEP [PKCS #1 v2.1]	Yes	

5 Finite State Model

The Finite State model contains the following states:

- Installed/Uninitialized
- Initialized
- Self-Test
- Idle
- Crypto Officer/User
- Error

The following are the important features of the state transition:

1. When the module is installed by the Crypto Officer, the module is in the Installed/Un-initialized state.
2. When the initialization command is applied to the module, i.e., the module is loaded on the memory, turning to the Initialization state. Then, it transits to the Self-Test state automatically, running the Power-up Tests. While in the Self-Test state, all data output via the data output interface is prohibited. On success the module enters Idle; on failure the module enters Error and the module is disabled. From the Error state the Crypto Officer may need to re-install to attempt correction.
3. From the Idle state (which is only entered if Self-Tests have succeeded), the module can transit to the Crypto Officer/User state when an API method is called.
4. When the API method has completed successfully, the state transits back to Idle.
5. If the Conditional Test (Continuous RNG Test or Pair-wise Consistency Test) fails, the state transits to Error and the module is disabled.
6. When On-demand Self-Test is executed, the module enters the Self-Test state. While in the Self-Test state, all data output via the data output interface is prohibited. On success the module enters Idle; on failure the module enters Error and the module is disabled.
7. When the de-initialization command is executed, the module goes back to the Installed/Un-initialized state.

6 Physical Security

Physical security is not applicable to this software module at Level 1 Security.

7 Operational Environment

This module is to be run in single user mode on the target operating system. For an operating system such as UNIX, Linux or Windows, it must be configured to run in single user mode.

8 Cryptographic Key Management

FIPS Crypto-J Module provides the underlying method to support FIPS 140-2 Level 1 key management. The user will select FIPS Approved algorithms and will handle keys with appropriate care to build up a system that complies with FIPS 140-2. It is the Crypto Officer and User's responsibility to select FIPS 140-2 validated algorithms (see Table 3).

8.1 Key Generation

SB FIPS Module provides FIPS 140-2 compliant key generation. The underlying random number generation uses a FIPS Approved method, the ANSI X9.62 RNG [4].

8.2 Key Establishment

FIPS Crypto-J Module provides the following FIPS allowed key establishment techniques [5]:

1. Diffie-Hellman (DH)
2. EC Diffie-Hellman (ECDH)
3. ECMQV
4. RSA PKCS1-v1 5
5. RSA OAEP

The RSA key wrapping techniques above are based on the PKCS #1 v2.1 standard, and are used to transport keys.

The ECDH and ECMQV key agreement technique implementations support elliptic curve sizes from 160 bits to above 512 bits that provide between 80 and 256 bits of security strength. The DH key agreement technique implementation supports modulus sizes from 512 bits to 15360 bits that provide between 56 and 256 bits of security strength, where 1024 bits and above must be used to provide minimum of 80 bits of security. The RSA implementation supports modulus sizes from 512 bits to 15360 bits that provide between 56 and 256 bits of security strength, where 1024 bits and above must be used to provide minimum of 80 bits of security.

It is the users responsibility to ensure that the appropriate key establishment techniques are applied to the appropriate keys.

8.3 Key Entry and Output

It is not allowed to import or export keys, including a seed key, and other security sensitive information outside of physical boundary in plaintext format. For key entry and export, users must ensure to deploy appropriate encryption method using FIPS Approved encryption algorithms in FIPS Crypto-J Module or any other FIPS validated cryptographic module.

8.4 Key Storage

FIPS Crypto-J Module is a low-level cryptographic toolkit, and as such does not provide key storage.

8.5 Zeroization of Keys

FIPS Crypto-J Module provides a zeroizable interface which implements a zeroization method. Zeroization of all keys and CSPs are performed in the finalizing method of the objects; JVM executes the finalizing methods every time it operates garbage collection. Also, a zeroization class is provided, where it will invoke the zeroize method when given an object that implements the zeroizable interface.

9 Self-Tests

9.1 Power-up Tests

9.1.1 Tests upon Power-up

Self-Tests are initiated automatically by the module at start-up. The following tests are applied:

1. **Known Answer Tests (KATs):**

KATs are performed on TDES, AES, SHS (via HMAC-SHS), HMAC-SHS, RNG, and RSA PKCS #1 v1.5 Signature Algorithm. For DSA and ECDSA, Pair-wise Consistency Test is used.

2. **Software Integrity Test:**

The software integrity test deploys ECDSA signature validation to verify the integrity of the module.

9.1.2 On-Demand Self-Tests

On-demand self tests may be invoked by the Cryptographic Officer or User by invoking a method, which is described in the Crypto Officer And User Guide in Appendix A.

9.2 Conditional Tests

The Continuous RNG Test is executed on all RNG generated data, examining the first 160 bits of each requested random generation for repetition. This ensures that the RNG is not stuck at any constant value.

Also, upon each generation of a RSA, DSA, or ECDSA key pair, the generated key pair is tested of their correctness by generating a signature and verifying the signature on a given message as a Pair-wise Consistency Test.

9.3 Failure of Self-Tests

Failure of the Self-Tests places the cryptographic module in the Error state, wherein no cryptographic operations can be performed. If any Self-Test fails, the cryptographic module will output error code.

10 Design Assurance

10.1 Configuration Management

A configuration management system for the cryptographic module is employed and has been described in a document to the certifying lab. It uses the Concurrent Versioning System (CVS) to track the configurations.

10.2 Delivery and Operation

Please refer to Section A.1 of Crypto Officer And User Guide in Appendix A to review the steps necessary for the secure installation and initialization of the cryptographic module.

10.3 Development

Detailed design information and procedures have been described in documentation submitted to the certifying laboratory.

This toolkit is designed and developed using high level language Java for Java users.

Development for the cryptographic module is carried out in a multi-platform environment. The CVS revision control system is used to control revisions. Development of new versions and major features are performed on a branch of the software, and these branches merged back into the trunk after testing and review, but the branch is maintained to perform post-release maintenance.

Releases are tested first in engineering (via an automated daily procedure, the daily build). They are then committed to the product candidate repository. These releases are then sent to the QA department which must run its own tests before they move them into the product branch. Only QA can move candidates into products.

10.4 Guidance Documents

Crypto Officer Guide and User Guide are provided in Appendix A. This appendix outlines the operations for Crypto Officer and User to ensure the security of the module.

11 Mitigation of Other Attacks

FIPS Crypto-J Module implements mitigation of the following attacks:

1. Timing attack on RSA
2. Attack on biased private key of DSA.

11.1 Timing Attack on RSA

When employing Montgomery computations, timing effects allow an attacker to tell when the base of exponentiation is near the secret modulus. This leaks information concerning the secret modulus.

In order to mitigate this attack, the following is executed: The bases of exponentiation are randomized by a novel technique that requires no inversion to remove (unlike other blinding methods e.g. BSAFE Crypto-C User Manual v4.2).

Note that Remote Timing Attacks are Practical:

<http://crypto.stanford.edu/dabo/papers/ssl-timing.pdf>

11.2 Attack on Biased Private Key of DSA

The standards for choosing ephemeral values in El-Gamal type signatures introduce a slight bias. Means to exploit these biases were presented to ANSI by D. Bleichenbacher.

In order to mitigate this attack, the following is executed: The bias in the RNG is reduced to levels which are far below the Bleichenbacher attack threshold.

Change Notice 1 of FIPS 186-2 is published to mitigate this attack:

<http://csrc.nist.gov/CryptoToolkit/tkdigsigs.html>

Appendix A Crypto Officer And User Guide

A.1 Installation

In order to carry out a secure installation of FIPS Crypto-J Module, the Crypto Officer must follow the procedure described in this section.

A.1.1 Installing

The Crypto Officer is responsible for the installation of FIPS Crypto-J Module. Only the Crypto Officer is allowed to install the product. The Crypto Officer must have administrative privileges on the computer.

Place `EccpressoFIPS.jar`, `EccpressoFIPSJca.jar`, and `EccpressoJDK15ECC.jar` files in `CLASSPATH` or as an installed extension. Note that `EccpressoFIPS.jar` is the cryptographic module.

A.1.2 Uninstalling

Remove the jar files from the computer hardware.

A.2 Commands

A.2.1 Initialization

```
FIPSManager.activateFIPSMode()
```

This method runs a series of Self-Tests on the module. These tests examine the integrity of the shared object, and the correct operation of the cryptographic algorithms. If these tests are successful, the module will be enabled.

A.2.2 Deinitialization

```
FIPSManager.deactivateFIPSMode()
```

This method de-initializes the module.

A.2.3 Self-Tests

```
FIPSManager.runSelfTests()
```

This method runs a series of Self-Tests, and returns true if the tests are successful. These tests examine the integrity of the shared object, and the correct operation of the cryptographic algorithms. If these tests fail, the module will be disabled. Section A.3 of this document describes how to recover from the disabled state.

A.2.4 Show Status

Status can be found by calling `FIPSManager.isFIPSMode()` and `FIPSManager.requestCryptoOperation()`. If both methods return true, the module is in the Idle state.

A.3 When Module is Disabled

When FIPS Crypto-J Module becomes disabled, attempt to bring the module back to the Installed state by calling the de-initialization method, and then to initialize the module using the initialization method. If the initialization is

successful, the module is recovered. If this attempt fails, uninstall the module and re-install it. If the module is initialized successfully by this re-installation, the recovery is successful. If this recovery attempt fails, it indicates a fatal error. Please contact Sterling Commerce Support immediately.