Apple Inc.

Apple FIPS Cryptographic Module, v1.0
FIPS 140-2 Non-Proprietary Security Policy

Document Control Number
APPLEFIPS_SECPOL_002.7

Version 2.7
February, 2011

Prepared by:

Shawn Geddis
Apple Inc.
11921 Freedom Drive
Suite 600
Reston, VA 20190

Phone: (703) 264-5103
Fax: (703) 264-5157
www.apple.com

Table of Contents

Section 1 FIPS Security Level Overview

FIPS Section	Level
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles, Services, and Authentication	1
Finite State Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1
EMI/EMC	1
Self-Tests	1
Design Assurance	1
Mitigation of Other Attacks	N / A

Table 1 FIPS Security Level Overview

Section 2 Executive Summary

Section 2.1 Overview

This document is the non-proprietary security policy supporting the *Apple FIPS Cryptographic Module*, v1.0. This document may be reproduced only in its original entirety, without revision. This security policy describes the module and how it meets the security requirements of FIPS 140-2. It also provides a specification of the FIPS 140-2 security rules under which the module operates. This document was prepared as part of the FIPS 140-2 Level 1 validation of the module.

With the exception of this non-proprietary security policy as well as the Role Guide: Crypto Officer and Role Guide: User documentation, all other FIPS 140-2 Validation Submission Documentation is proprietary to Apple Inc. and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact Apple Inc.

Section 2.2 Introduction

The Level 1 *Apple FIPS Cryptographic Module*, v1.0 is used within Apple Mac OS X v10.6. It consists of the *Apple Cryptographic Service Provider (AppleCSP),* the module's PRNG, and the *FIPSPerformSelfTest* helper application. The module provides cryptographic services for Apple Mac OS X.

Section 3 Apple FIPS Cryptographic Module

Section 3.1 Overview

Mac OS X security services are built on two open-source standards: BSD (Berkeley Software Distribution) and CDSA (Common Data Security Architecture). BSD is a form of the UNIX operating system and provides fundamental services, such as the basis for the Mac OS X file system, including file access permissions. CDSA provides a much wider array of security services, including finer-grained access permissions, authentication of users' identities, encryption, and secure data storage. Although CDSA has its own standard application programming interface (API), it is complex and does not follow standard Macintosh programming conventions. Therefore, Mac OS X includes its own security APIs that call the CDSA API. The Mac OS X security architecture is layered, with BSD on the bottom, CDSA in the middle, Mac OS X security APIs above that, and applications[1] that call the security services at the top. Figure 1 below illustrates this architecture.

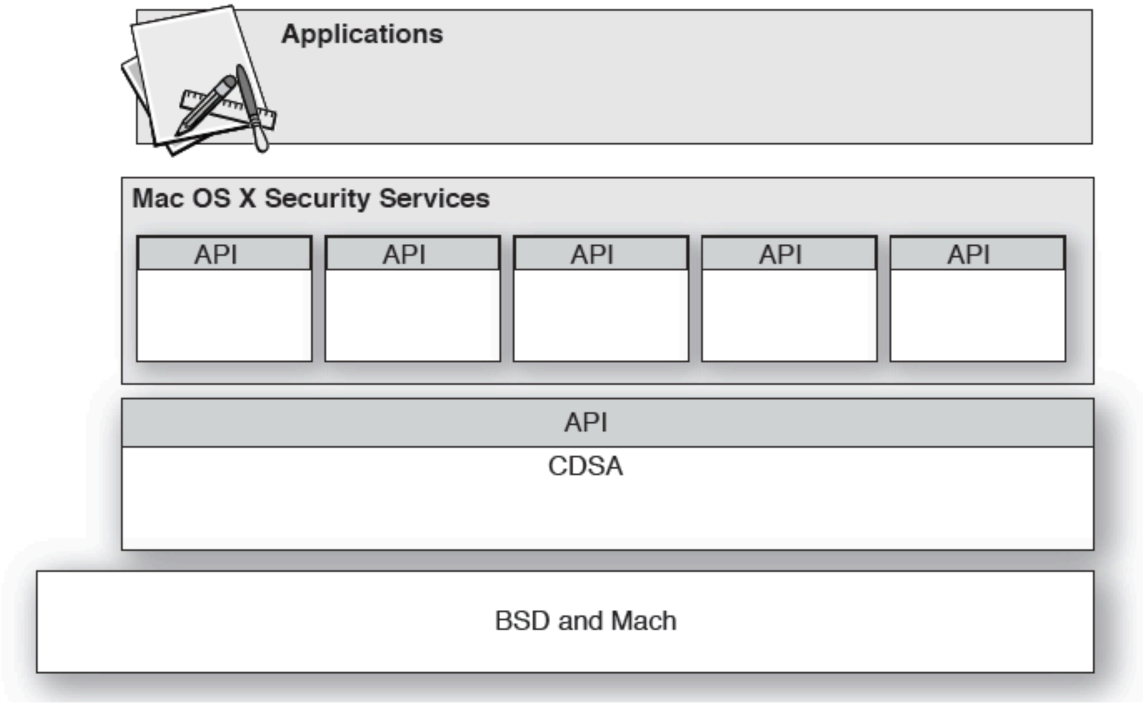

Figure 1 Mac OS X Security Architecture Overview

[1] Apple applications include Keychain Access, FileVault, Finder, and Safari.

CDSA is an Open Source security architecture adopted as a technical standard by the Open Group. Apple has developed its own Open Source implementation of CDSA. The core of CDSA is CSSM (Common Security Services Manager), a set of Open Source code modules that implement a public API called the CSSM API. CSSM provides APIs for cryptographic services (such as creation of cryptographic keys, encryption and decryption of data), certificate services (such as creation of digital certificates, reading and evaluation of digital certificates), secure storage of data, and other security services.

CSSM also defines an interface for plug-ins that implements security services for a particular operating system and hardware environment. The implementation on a given platform can optionally supply a middleware layer that provides an operating-system-specific API for applications. Whether such a layer is present or not, applications can call the CSSM API directly. Mac OS X implements nearly all the standard features of CSSM, plus a set of middleware security services to provide a Mac OS X-standard interface for application programmers. In addition, to enhance the security of the most sensitive operations, the Mac OS X implementation runs a Security Server daemon as a separate process. The Security Server daemon launches another process, the Security Agent, which serves as the user interface for Security Server.

The CDSA standard defines a four-layer architecture, with the top layer being the applications that use the CDSA security features. Figure 2 below illustrates the Mac OS X implementation of CDSA and shows the first three layers: the CDSA plug-ins, CSSM, and the Mac OS X APIs, which constitute the middleware layer. The Authorization Services, the Security Server daemon, and the Security Agent shown in the figure are technically outside of CDSA, but they are shown here for completeness because they constitute an integral part of the Mac OS X security architecture.

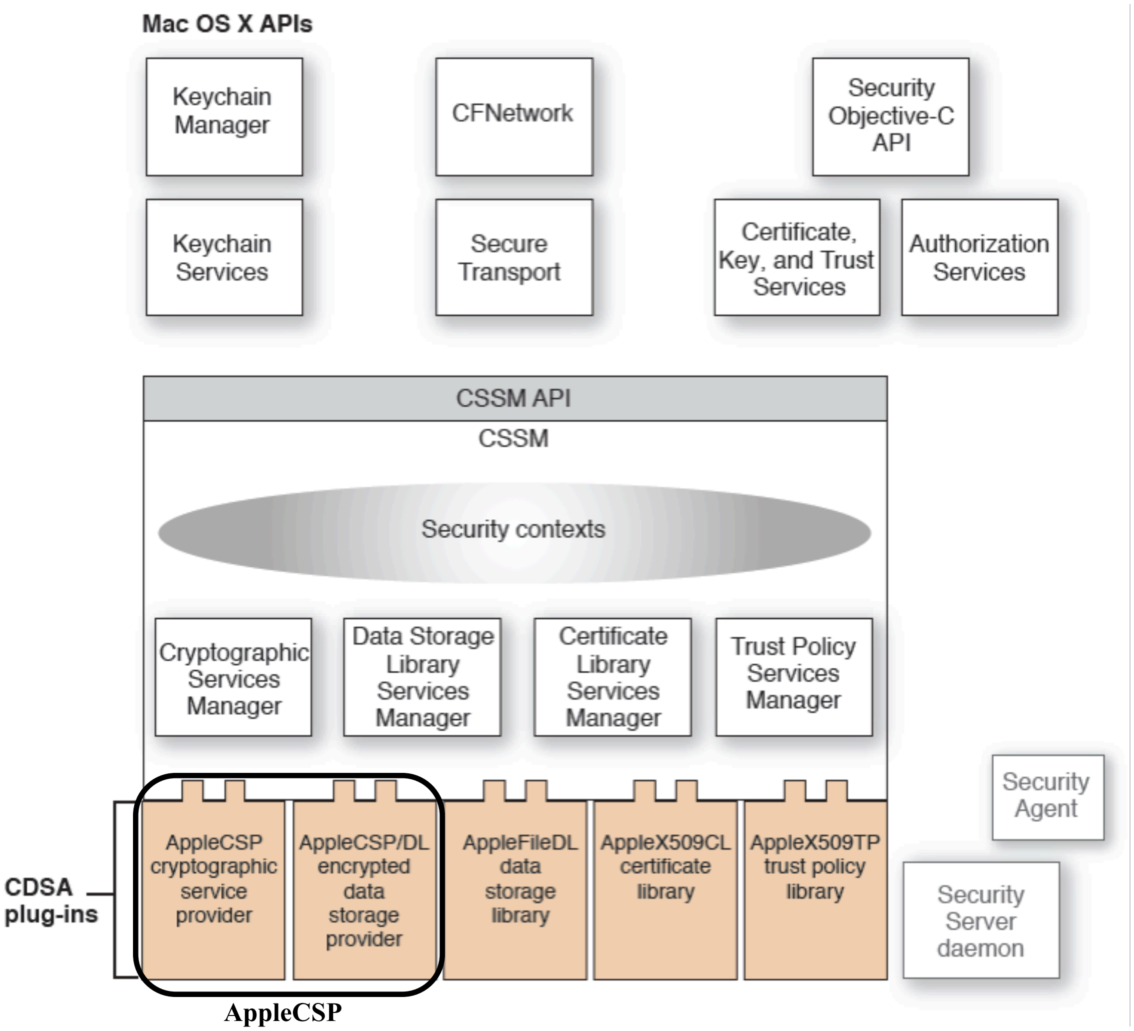
Mac OS X APIs
Keychain Manager
CFNetwork
Security Objective-C API
Keychain Services
Secure Transport
Certificate, Key, and Trust Services
Authorization Services
CSSM API
CSSM
Security contexts
Cryptographic Services Manager
Data Storage Library Services Manager
Certificate Library Services Manager
Trust Policy Services Manager
Security Agent
CDSA plug-ins
AppleCSP cryptographic service provider
AppleCSP/DL encrypted data storage provider
AppleFileDL data storage library
AppleX509CL certificate library
AppleX509TP trust policy library
Security Server daemon
AppleCSP

Figure 2 Mac OS X Implementation of CDSA

Security contexts in Figure 2 are data structures used by CSSM to assist applications in managing the many parameters used in security operations. The CSSM managers implement the standard CSSM API. The CDSA plug-ins shown in Figure 2 are those provided as part of Mac OS X. The CDSA specification allows any number of plug-ins. As long as a plug-in follows the rules for interfacing with the CSSM managers, it can implement any portion of the CDSA feature set, including a combination of features associated with two or more of the CSSM managers. The CDSA specification even allows for the expansion of CDSA by the addition of elective module managers and associated plug-ins. Plug-ins can call each other as well as being called by the CSSM managers and, in fact, it is common for them to do so. All secure communications and authentication protocols are based on keys and encryption provided by the AppleCSP.

Section 3.2 Cryptographic Module Specification

The logical cryptographic boundary of *Apple FIPS Cryptographic Module*, v1.0 ("Module library") is the shared object library itself. The logical cryptographic boundary consists of the *Apple Cryptographic Service Provider (AppleCSP),* the module's PRNG, and the *FIPSPerformSelfTest* helper application. The *AppleCSP* is a basic plug-in module that works together with the helper application. The PRNG is used in generating the module's keys. The *FIPSPerformSelfTest* file performs the FIPS required power on self-tests for the *AppleCSP*. The physical cryptographic boundary of the Module library is the enclosure of the computer system on which the module is running.

Figure 3 below shows the cryptographic boundary of the module. The logical boundary is indicated by the red dotted line while the physical boundary is indicated by the black dotted line. The **Power On Self Test** block within the diagram represents the *FIPSPerformSelfTest* file, the **PRNG** block represents the module's PRNG, and the **CSP Module** block within the diagram represents the *AppleCSP*.

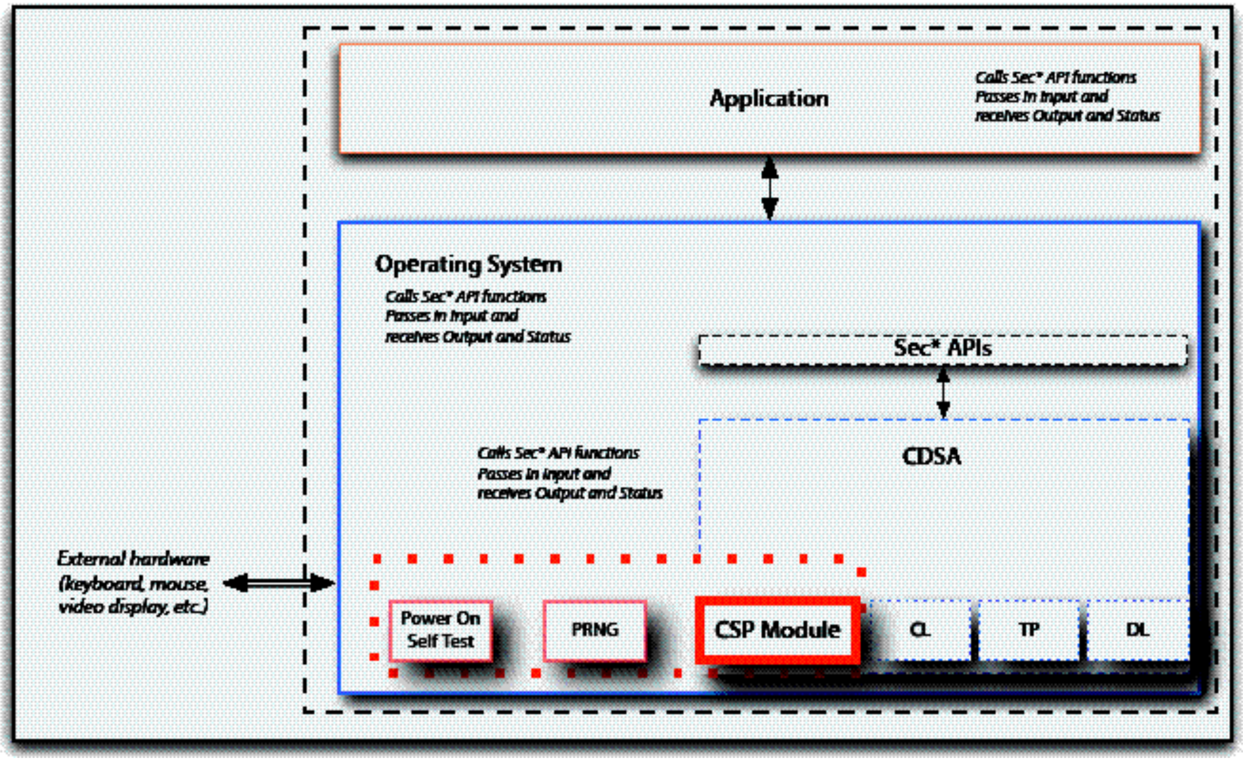

Figure 3 Cryptographic Module Boundary

Section 3.3 *Modes of Operation*

The module has two modes of operation: Approved mode and Non-approved mode. The module runs in the Approved mode by default, and when the module uses an internally generated RSA key pair for signature generation and verification and for RSA key wrapping, or any non-allowed algorithms listed in Table 6, the module is considered running in the Non-approved mode.

The installation of the *Apple FIPS Cryptographic Module* by the Crypto Officer involves four steps and more information about these steps can be found in the "Role Guide: Crypto Officer" document:

1. Obtaining the FIPS Administration Tools installer
2. Installing the FIPS Administration Tools
3. Verifying the FIPS Administration Tools were successfully installed
4. Verify the integrity of the FIPS Administration Tools

The User can also verify the *Apple FIPS Cryptographic Module* status by running the *FIPSPerformSelfTest status* command in the Terminal application. More information on the User verification of the *Apple FIPS Cryptographic Module* can be found in the "Role Guide: User" documentation.

Section 3.4 Cryptographic Module Ports and Interfaces

The cryptographic module is a software module. This module was tested on the 15-inch MacBook Pro laptop computer platform. The platform for the module provides a number of physical ports and logical interfaces. The platform's physical ports correspond to the ports of the laptop computer that executes the module. They include a 15.4 inch display, power button, power adaptor port, rechargeable battery pack, two USB 2.0 ports, audio line in/optical digital audio input, headphone/optical digital audio output, two AirPort Extreme/Bluetooth wireless antennas, ExpressCard/34 slot, FireWire 400 port, FireWire 800 port, Gigabit Ethernet, DVI port, SuperDrive optical drive, keyboard, trackpad, speaker, microphone, iSight video camera and LEDs. The module implements the required FIPS 140-2 logical interfaces through application programming interface (API) calls as shown in the following table.

FIPS 140-2 Logical Interfaces	Module Physical Ports	Module Logical Interfaces
Data Input	USB, audio line in/optical digital audio input, wireless antennas, ExpressCard/34, FireWire, Ethernet, SuperDrive, microphone, iSight video camera	Data passed to the API calls to be used by the Module
Data Output	Display, USB, headphone/optical digital audio output, wireless antennas, ExpressCard/34, FireWire, Ethernet, DVI, SuperDrive, speaker	Data returned from API calls, generated by the Module
Control Input	USB, wireless antennas, ExpressCard/34, FireWire, Ethernet, SuperDrive, trackpad, keyboard	Exported API calls
Status Output	Display, USB, wireless antennas, ExpressCard/34, FireWire, Ethernet, SuperDrive, DVI, LEDs	Returned status information and return codes provided by API function calls after execution
Power	Power button, power adaptor port, battery pack	N/A

Table 2 Mapping of Ports and Interfaces

Section 3.5 Roles, Services, and Authentication

Section 3.5.1 Roles

The Apple cryptographic module supports two authorized roles: *User* and *Crypto Officer*.

The *User* can request access to the module in order to use its cryptographic services.

The *Crypto Officer* can request access to install or remove the module as well as perform power on self tests and check the status of the module.

Section 3.5.2 Services

Role	Service	Critical Security Parameter (CSP) Access
User		
	Show FIPS Enabled Status	Read
	Show FIPSPerformSelfTest Version	Read
	AES secret key data encryption/ decryption	Write, Execute
	Triple-DES secret key data encryption/decryption	Write, Execute
	RSA/DSA/ECDSA Signature generation and verification	Write, Execute
	Diffie-Hellman public/private key agreement	Write, Execute
	Elliptic Curve Diffie-Hellman public/private key agreement	Write, Execute
	Pseudo Random Number Generation (PRNG)	Write, Execute
	SHS Hashing	Write, Execute
	HMAC SHA-1 Keyed Hashing	Write, Execute
Crypto Officer		
	Installation	Execute
	Show FIPS Enabled Status	Read
	Show FIPSPerformSelfTest Version	Read
	Perform Full FIPS Self Test	Execute
	AES secret key data encryption/ decryption	Write, Execute

	Triple-DES secret key data encryption/decryption	Write, Execute
	RSA/DSA/ECDSA Signature generation and verification	Write, Execute
	Diffie-Hellman public/private key agreement	Write, Execute
	Elliptic Curve Diffie-Hellman public/private key agreement	Write, Execute
	Pseudo Random Number Generation (PRNG)	Write, Execute
	SHS Hashing	Write, Execute
	HMAC SHA-1 Keyed Hashing	Write, Execute

Table 3 Roles and Services

Section 3.5.3 Authentication

Within the constraints of FIPS 140-2 Level 1, the module does not implement an authentication mechanism for operator authentication. The module relies upon the operating system, which lies outside the logical boundary, for operator authentication.

Section 3.6 Physical Security

Physical Security is not required for the software module. The FIPS software was tested on a 15-inch MacBook Pro laptop computer with an Intel microprocessor running at a clock speed of 2.33 GHz. The computer is made from production grade components and includes a lightweight aluminum alloy production grade enclosure.

Section 3.7 Operational Environment

The software module runs on the Apple Mac OS X 10.6 in single operator mode of operation. When the Mac operating system loads the module into memory, the *FIPSPerformSelfTest* runs code signing (RSA Signature) validations on all components of the module with the exception of HMAC-SHA1 validation on the PRNG, which will ensure a full cryptographic verification of the module. Loading will only continue if the module passes these checks. A number of other self-tests are also run at this time. The complete list of self-tests are listed in section 3.10.

Section 3.8 Cryptographic Key Management

The module provides the capability to use cryptographic keys with several algorithms. The implemented FIPS-approved algorithms include AES, Triple-DES, RSA/DSA/ECDSA, SHA-1/224/256/384/512, HMAC SHA-1, and FIPS 186-2 PRNG.

Section 3.8.1 Key Generation

This module implements the FIPS Approved FIPS 186-2 PRNG to generate keys.

Section 3.8.2 Key Establishment

The module uses Diffie-Hellman and Elliptic Curve Diffie-Hellman key agreement for key establishment. Methodologies providing a minimum of 80 bits of encryption strength are allowed in the FIPS mode of operation. Encryption strength is determined in accordance with FIPS 140-2 Implementation Guidance 7.5 and NIST Special Publication 800-57 (Part 1).

Section 3.8.3 Key Entry and Output

All keys are imported from, or output to, the invoking program running on the same computer. All keys entered into the module are electronically entered in plain text form. Keys are output from the module in plain text form.

Section 3.8.4 Key Storage

Keys stored in memory are stored in plaintext.

Section 3.8.5 Key Zeroization

All keys can be zeroized by overwriting them, deleting them, or by rebooting the computer.

Section 3.8.6 List of Keys and CSP

CSPs	CSPs type	Generation	Storage	Use
AES keys	Symmetric secret keys	Internal via FIPS 186-2 PRNG	Plaintext[2]	Data encryption/ decryption
Triple-DES keys	Symmetric secret keys	Internal via FIPS 186-2 PRNG	Plaintext[2]	Data encryption/ decryption
RSA/DSA/ ECDSA Key Pairs	Asymmetric private and public key pairs	Internal via FIPS 186-2 PRNG	Plaintext[2]	Signing and Verification
Diffie-Hellman and Eliptic Curve Diffie-Hellman key pairs	Diffie-Hellman and Eliptic Curve Diffie-Hellman private and public key pairs	Internal via FIPS 186-2 PRNG	Plaintext[2]	Key agreement
RSA Key Pair[1]	Key wrapping key	Internal via FIPS 186-2 PRNG	Plaintext[2]	Key wrapping
HMAC key	Triple-DES key	Internal via FIPS 186-2 PRNG	Plaintext[2]	Message authentication
FIPS 186-2 PRNG seed keys	Secret key values	Internal – by gathering entropy	Plaintext[2]	Pseudo-random number generator for keys

Note [1]: Internally generated RSA keys must never be used in a FIPS Approved mode of operation for signature generation and verification and for RSA key wrapping.

Note [2]: Keys stored in memory are stored in plaintext.

Table 4 List of Keys and CSP

Section 3.9 EMI/EMC

The module is designed to meet security level 1 requirements for EMI/EMC. The module was tested and found compliant with requirements for a Class B digital device.

Section 3.10 *Self-Tests*

The module performs a set of self-tests to ensure proper operation in compliance with FIPS 140-2. These self-tests are run during power-on (power-on self-tests) or when certain conditions are met (conditional self-tests). Self tests are performed for the approved security functions and algorithms as required.

<u>Power-On Self-Tests</u>
Software Integrity Test (RSA and HMAC-SHA1)
RNG KAT
AES KAT
Triple-DES KAT
RSA SHA-1 KAT
RSA SHA-224 KAT
RSA SHA-256 KAT
RSA SHA-384 KAT
RSA SHA-512 KAT
DSA Pairwise Consistency Test (DSA Key GEN/DSA SIG GEN/DSA SIG VER)
ECDSA Pairwise Consistency Test (ECDSA KEYGEN/ECDSA SIG GEN/ECDSA SIG VER)
SHA-1 KAT
SHA-224 KAT
SHA-256 KAT
SHA-384 KAT
SHA-512 KAT
HMAC SHA-1 KAT

<u>Conditional Self-Tests</u>
CRNG Tests

Section 3.11 *Design Assurance*

Apple manages and records source code and associated documentation files. Apple implements a system for document and source code management compliant with FIPS 140-2 Level 1 security.

The Apple module hardware data, which includes descriptions, parts data, part types, bills of materials, manufacturers, changes, history, and hardware documentation are managed and recorded. Additionally, configuration management is provided for the module's FIPS documentation. Document management utilities provide access control, versioning, and logging.

Section 3.12 *Mitigation of Other Attacks*

The module does not use other security mechanisms to mitigate against specific attacks.

Section 4 Secure Operation

Section 4.1 Security Functions

The module meets Level 1 requirements for FIPS 140-2.

The Apple cryptographic module supports the following approved and non-approved security functions.

Service	Algorithm	Standard	Mode/Key Size/Description	Certificate Number
Asymmetric Key				
	RSA	PKCS#1 v1.5	PKCS#1 v1.5: SigGen; SigVer; 1024, 1536, 2048, 3072, 4096; SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	681
	ECDSA	ANSI X9.62	KeyGen; SigGen; SigVer: Curves (P-192 P-256 P-384 P-521)	176
	DSA	FIPS 186-2	FIPS186-2: KeyGen Mod(1024); SigGen Mod(1024); SigVer Mod (1024)	453
Symmetric Key				
	AES	FIPS 197	ECB(e/d; 128,192,256); CBC(e/d; 128,192,256)	1400
	Triple-DES	FIPS 46-3, SP 800-67	TECB(e/d; KO 1,2); TCBC(e/d; KO 1,2)	955
PRNGs				
	FIPS186-2 PRNG	FIPS 186-2	FIPS 186-2: x-Original; SHA-1	767
Hashes				
	SHA-1	FIPS 180-2	Byte orienting hashing	1271
	SHA-224	FIPS 180-2	Byte orienting hashing	1271
	SHA-256	FIPS 180-2	Byte orienting hashing	1271
	SHA-384	FIPS 180-2	Byte orienting hashing	1271
	SHA-512	FIPS 180-2	Byte orienting hashing	1271
Keyed-Hashes				
	HMAC SHA-1	FIPS 198		823

Table 5 Approved FIPS 140-2 Security Functions

Service	Algorithm	Standard	Mode of Operation
Ciphers			
	DES		ECB, CBC
	Blowfish		ECB, CBC
	CAST		ECB, CBC
	ASC		
	RC2		ECB, CBC
	RC4		ECB, CBC
	RC5		ECB, CBC
Asymmetric Key			
	RSA Encrypt/Decrypt		RSA (key wrapping; key establishment methodology provides between 80 and 128 bits of encryption strength; non-compliant less than 80 bits of encryption strength)
	RSA Key Generation	PKCS#1	RSA (key generation)
	Diffie-Hellman	ANSI X9.42	Diffie-Hellman (key agreement; key establishment methodology provides 80 or 112 bits of encryption strength; non-compliant less than 80-bits of encryption strength)
	Elliptic Curve Diffie-Hellman	ANSI X9.63	EC Diffie-Hellman (key agreement; key establishment methodology provides between 80 and 256 bits of encryption strength).
	FEE		
Hashes			
	MD2		
	MD5		
Keyed-Hashes			
	HMAC MD5		

Table 6 Non-Approved FIPS 140-2 Security Functions

Section 4.2 Crypto Officer Guidance

The Crypto Officer must operate the module in a manner consistent with the guidance provided within the "Role Guide: Crypto Officer" document. The secure operation procedures include the initial setup, configuring the module in a FIPS compliant manner, and keeping the module in a FIPS-approved mode of operation.

Section 4.3 User Guidance

The User must operate the module in a manner consistent with the guidance provided within the "Role Guide: User" document to make sure that only approved security functions are allowed in the FIPS approved mode of operation. Only the services listed in Table 3 should be used if a FIPS approved mode of operation is to be maintained. All security functions listed in Table 5 can be used in the FIPS approved mode of operation. Although outside the boundary of the module, the User should be careful not to provide cryptographic keys or other critical security parameters (CSPs) to other unauthorized parties.

In addition to the security functions listed in Table 5, both Diffie-Hellman and Elliptic Curve Diffie-Hellman for key agreement listed in Table 6 are also allowed in the FIPS approved mode of operation. No other non-approved security function should be used. Key establishment methodologies provide a minimum of 80 bits of encryption strength. Encryption strength is determined in accordance with FIPS 140-2 Implementation Guidance 7.5 and NIST Special Publication 800-57 (Part 1).

The User can verify the *Apple FIPS Cryptographic Module* status by running the *FIPSPerformSelfTest status* command in the Terminal application. The User can verify the *Apple FIPS Cryptographic Module* version by running the *FIPSPerformSelfTest version* command in the Terminal application. More information can be found in the "Role Guide: User" documentation.

Section 5 Glossary and References

Section 5.1 *Glossary*

API	Application Programming Interface
BSD	Berkeley Software Distribution
CBC	Cipher Block Chaining
CDSA	Common Data Security Architecture
CMVP	Cryptographic Module Validation Program
CRC	Cyclical Redundancy Check
CSP	Critical Security Parameter
CSSM	Common Security Services Manager
EDC	Error Detection Code
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FCC	Federal Communication Commission
FIPS	Federal Information Processing Standard
KAT	Known Answer Test
LED	Light Emitting Diode
MAC	Message Authentication Code
NIST	National Institute of Standards and Technology
PRNG	Pseudo Random Number Generator
RAM	Random Access Memory
SHA	Secure Hash Algorithm

Section 5.2 References

This document deals only with operations and capabilities of the module in the technical terms of a FIPS 140-2 cryptographic module security policy. More information is available about the module and the Apple Mac OS X on the Apple Web site at (http://www.apple.com/macosx/security/).

To get the latest updates on Apple's security services and for pointers to other Apple security resources, go to the ADC technology page for security at http://developer.apple.com/security/.

CDSA, implemented as part of the Mac OS X security architecture, is an Open Source standard by the Open Group (http://www.opengroup.org/security/cdsa.htm). For an introduction to CDSA, see *CDSA Explained*, second edition, from the Open Group. The CDSA/CSSM technical standard is *Common Security: CDSA and CSSM*, version 2 (with corrigenda), also from the Open Group.

Information on the full line of products from Apple can be found at (http://www.apple.com/mac).

Information on FIPS 140-2 validations and the Cryptographic Module Validation Program can be found at (http://csrc.nist.gov/groups/STM/cmvp/). The website also contains contact information for answers to technical or sales-related questions regarding the Cryptographic Module Validation Program.