Crypto Officer Role Guide for FIPS 140-2 Compliance

iOS 7

Contents

Overview

In highly regulated industries, IT System Administrators and Crypto Officers are frequently required to ensure deployed systems are correctly using FIPS 140-2 Validated Cryptographic Modules. The two Apple Cryptographic Modules in iOS 7 achieved **FIPS 140-2 Level 1 Conformance Validation** under the Cryptographic Module Validation Program (CMVP) – a joint American and Canadian security accreditation program for cryptographic modules.

These two modules are identified under the CMVP with the module names of: a) "**Apple iOS CoreCrypto Module v4.0**" and b) "**Apple iOS CoreCrypto Kernel Module v4.0**." The **CoreCrypto Module** is available to developers for Applications and Services running in User Space. The **CoreCrypto Kernel Module** is used only by the iOS Kernel.

Within this and other Apple documents, those modules are also referred to with the name of **"Apple FIPS Cryptographic Module v4.0."**

Apple iOS CoreCrypto Module v4.0	Validation Certificate **#2020**

http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2013.htm#2020

Apple iOS CoreCrypto Kernel Module v4.0	Validation Certificate **#2021**

http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2013.htm#2021

All Apple Validated Crypto Modules can be found under CMVP's FIPS 140-2 Vendor List here - http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401vend.htm

This Crypto Officer Role Guide provides IT System Administrators with the necessary technical information to ensure FIPS 140-2 compliance of iOS 7 systems. This guide walks the reader through the system's assertion of cryptographic module integrity and the steps necessary if module integrity requires remediation.

Compliant Applications and Services

Compliancy Requirements on Crypto Officers are not limited to the use of products containing a validated cryptographic module, but extend to their attestation that applications and services in use are FIPS 140-2 Compliant. Compliance is defined by both the use of a FIPS 140-2 validated module and the proper use of FIPS-Approved Algorithms. A cryptographic module may contain additional algorithms that are not FIPS-Approved and if used, would indicate a Non-FIPS Compliant condition. A FIPS 140-2 Level 1 Conformance Validation does not require the cryptographic module ensures applications and services only use FIPS-Approved algorithms.

Apple

A high-level, non-exhaustive list of Apple applications and services that are FIPS 140-2 Compliant in iOS 7 would include the following:

Services

Data Protection, Hardware Encryption, HTTPS, Keychain Services, S/MIME, TLS/SSL, VPN, and 802.1X.

Applications

App Store, iTunes Store, Calendar, Contacts, FaceTime, Messages, Mail, Safari, and Software Update.

Developer and Crypto Officer Resources

There are resources available to developers providing guidance on cryptographic services and API documentation for iOS 7. Developers should refer to these resources to ensure their products and services are FIPS 140-2 Compliant on iOS 7.

Apple iOS CoreCrypto Module, v4.0 FIPS 140-2 Non-Proprietary Security Policy

http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp2020.pdf

Apple iOS CoreCrypto Kernel Module, v4.0 FIPS 140-2 Non-Proprietary Security Policy

http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp2021.pdf

iOS Product Security: Validations and Guidance

http://support.apple.com/kb/HT5808

iOS Security Whitepaper

The iOS Security whitepaper's target audience is enterprise IT and provides both an overview and some low-level details about the security processes and cryptographic algorithms in use throughout various parts of the platform.

http://images.apple.com/iphone/business/docs/iOS_Security_Dec13.pdf

Security Overview

https://developer.apple.com/library/mac/documentation/Security/Conceptual/Security_Overview/Security_Overview.pdf

Cryptographic Services Guide

https://developer.apple.com/library/mac/documentation/Security/Conceptual/cryptoservices/cryptoservices.pdf

Certificate, Key, and Trust Services Programming Guide

https://developer.apple.com/library/mac/documentation/Security/Conceptual/CertKeyTrustProgGuide/CertKeyTrustProgGuide.pdf

Compliant Platforms

Compliant platforms are all supported Apple systems running iOS 7. During the validation process for FIPS 140-2 Conformance, the cryptographic modules are put through operational testing environments on supported platforms and noted on the issued certificate. The **CoreCrypto** and **CoreCrypto Kernel** modules were validated under the following operational testing environments:

Module:	**Apple iOS CoreCrypto Module v4.0**
Platforms:	A4 with iOS 7 (User Space) A5 with iOS 7 (User Space) A6 with iOS 7 (User Space)

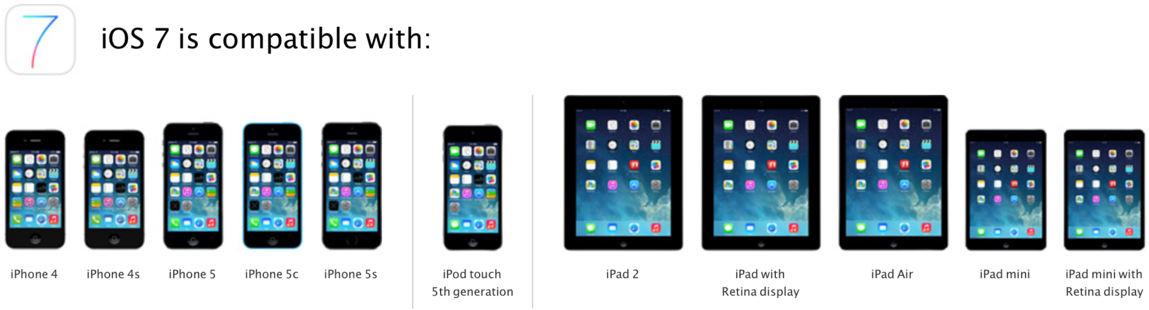
Module:	**Apple iOS CoreCrypto Kernel Module v4.0**
Platforms:	A4 with iOS 7 (Kernel Space) A5 with iOS 7 (Kernel Space) A6 with iOS 7 (Kernel Space)

Self-assertion for A7-based devices

The FIPS 140-2 Conformance Validation process for these two cryptographic modules began prior to the release of iOS devices based on the A7 processor. At this time, Apple is self-asserting the FIPS 140-2 compliance when running on A7-based iOS devices under this validation and will include those devices in the operational testing environments for the next round of FIPS 140-2 Conformance Validation for the CoreCrypto and CoreCrypto Kernel modules.

Compliant hardware

For FIPS 140-2 Compliance, the platforms noted above articulate Apple systems which were used for operational testing of the cryptographic modules. The CoreCrypto and CoreCrypto Kernel modules on Apple systems with either the A4, A5, A6 or A7 processors running iOS 7 also take advantage of the additional processor embedded cryptographic engine. Compliant hardware are all Apple systems meeting the technical specifications to run iOS 7. The platforms that are compatible with iOS 7 as of November 2013 can be found here http://www.apple.com/ios/whats-new/ which notes the following:



The FIPS Power-On-Self-Test (POST) process flow

1. Apple iOS system is physically Powered on

2. Operating System (iOS 7) begins bootstrap process

3. Operating System ensures integrity of the **CoreCrypto Kernel Module**

    3.1. Validation of the `corecrypto.kext`
        3.1.1. The kernel determines operating environment (i.e arm7)
        3.1.2. The kernel reads a validated HMAC_SHA256 from the `corecrypto.kext`
        3.1.3. The `corecrypto.kext` is launched and given the correct validated HMAC from 3.1.2
        3.1.4. The `corecrypto.kext` will generate an HMAC_SHA256 of the corecrypto.kext code and compare the result against the validated HMAC_SHA256 from 3.1.2
        3.1.5. If the calculated HMAC_SHA256 does not match the validated HMAC_SHA256, the system will panic and halt
    3.2. The cipher Power-On-Self-Test (POST) validates the algorithms and modes
        3.2.1. The `corecrypto.kext` performs POST on algorithms and modes
        3.2.2. If any part of the POST fails, the system will panic and halt

4. Operating System ensures Integrity of **CoreCrypto Module**

    4.1. Validation of the `corecrypto.dylib`
        4.1.1. Upon user space environment setup by the kernel, **launchCtl** will launch the test application `/usr/libexec/cc_fips_test`
        4.1.2. An HMAC_SHA256 of the user space corecrypto.dylib will be generated and compared to the HMAC_SHA256 value stored at `/var/db/FIPS/fips_data`
        4.1.3. If the calculated HMAC_SHA256 does not match the stored HMAC_SHA256, the system will panic and halt
    4.2. The cipher Power-On-Self-Test (POST) validates the algorithms and modes
        4.2.1. The `cc_fips_test` performs POST on algorithms and modes
        4.2.2. If any part of the POST fails, the system will panic and halt

5. Halt upon failure of any tests

    5.1. If any phase or step of testing components fails, the system will log the failure and panic and halt the device immediately.
    5.2. The logging messages are sent to the `console` and can be viewed using tools such as Xcode's "Organizer".

How to verify integrity of the modules

A boot-up of the iOS 7 device forces the FIPS POST which verifies the integrity of both the CoreCrypto Kernel and CoreCrypto modules. If the device boots-up successfully, both modules have passed integrity verification. If the device halts or shuts down during boot-up, an integrity issue has been found during the POST process.

Rebooting the iOS 7 device will always force integrity verification of both modules.

How to mitigate integrity issues of the modules

If a crypto module integrity issue has been identified during the FIPS POST, the only recourse the Crypto Office has for mitigation is to re-install iOS 7 on the device.

If the Crypto Officer needs assistance in restoring the iOS 7 Software, Apple Knowledge Base Articles should prove to be quite helpful.

A few helpful support articles available from the Apple Support Knowledge Base:

iTunes: Restoring iOS software

http://support.apple.com/kb/HT1414

iTunes 11 for Mac: Update and restore software on iPod, iPhone, or iPad

http://support.apple.com/kb/PH12124

iTunes 11 for Windows: Update and restore software on iPod, iPhone, or iPad

http://support.apple.com/kb/PH12324

If needing to perform an Apple Support-wide search for all articles pertaining to "Restoring iOS Software", use the following URL:

http://support.apple.com/kb/index?page=search&product=&q=Restoring%20iOS%20Software&src=support_site.kbase.search.searchresults

If choosing to perform an Apple Support-wide search for all articles pertaining to "FIPS iOS", use the following URL:

http://support.apple.com/kb/index?page=search&product=&q=FIPS%20iOS&src=support_site.kbase.search.searchresults

FIPS 140-2 Validated Algorithms

The CoreCrypto and CoreCrypto Kernel Modules are cryptographic libraries offering various cryptographic mechanisms to Apple frameworks. Algorithms from the two Apple Cryptographic Modules in iOS 7 achieved **Cryptographic Algorithm Validation** under the Cryptographic Algorithm Validation Program (CAVP)

Modes of Operation

The CoreCrypto and CoreCrypto Kernel Modules have an Approved and Non-Approved modes of operation. The Approved mode of operation is configured in the system by default and cannot be changed. If the device boots up successfully then CoreCrypto framework and CoreCrypto KEXT have passed all self-tests and are operating in the Approved mode.

The Approved security functions are listed in **Table 3: Approved Security Functions** of the Non-Proprietary Security Policy documents posted along with the module validation certificate under CMVP. The Security Policy document links can be found above in the *Developer Resources* section. Column four (Val. No.) lists the validation numbers obtained from NIST for successful validation testing of the implementation of the cryptographic algorithms on the platforms as shown in Table 2 under CAVP.

Any calls to the non-Approved security functions listed in **Table 4: Non-Approved Security Functions** of the Non-Proprietary Security Policy documents will cause the module to assume the non-Approved mode of operation. Operators of the modules are strongly advised to avoid calling the functions in Table 4. If the module is operating in the non-Approved mode, operators are strongly cautioned to not use any CSP's previously utilized in the Approved mode of operation.

Note in the Security Policy documents under Key / CSP Establishment that the module provides DH- and ECDH-based key establishment services in the Approved mode. The module provides key establishment services in the Approved mode through the PBKDFv2 algorithm. The PBKDFv2 function is provided as a service and returns the key derived from the provided password to the caller. The caller shall observe all requirements and should consider all recommendations specified in SP800-132 with respect to the strength of the generated key, including the quality of the password, the quality of the salt as well as the number of iterations. The implementation of the PBKDFv2 function requires the user to provide this information.

Refer to http://csrc.nist.gov/groups/STM/cavp/index.html for the current standards, test requirements, and special abbreviations used.

The Approved Security Functions Table has been recreated below for quick and easy access, but is not the exhaustive list of all algorithms supported by the cryptographic modules. Crypto Officers are highly encouraged to obtain and read the Security Policy document for complete technical explanations on the CoreCrypto and CoreCrypto Kernel modules.

Suite B Cryptographic Algorithms

The CoreCrypto Module (User Space) does provide for the use of Suite B Cryptographic Algorithms as are called out on the NSA Suite B Cryptography web page. Those algorithms include AES, ECDH, ECDSA and SHA-256/-384. For further information from NSA about Suite B Algorithms, refer to http://www.nsa.gov/ia/programs/suiteb_cryptography/.

Module Name:			CoreCrypto Module v4.0		(User Space)
Alg.	Platform Certificate			Standards	Description
	A4	A5	A6		
AES	2508	2509	2547	FIPS 197 SP 800-38 A SP 800-38 D	**User space and generic, non-optimized software.** **ECB** (128 , 192 , 256) **CBC** (128 , 192 , 256) **CFB8** (128 , 192 , 256) **CFB128** (128 , 192 , 256) **OFB** (128 , 192 , 256) **CTR** (128 , 192 , 256) - ***int only*** **GCM** **KS:** **AES_128** Tag Length(s): 128 120 112 104 96 64 32 **AES_192** Tag Length(s): 128 120 112 104 96 64 32 **AES_256** Tag Length(s): 128 120 112 104 96 64 32 **IV Generated:** Internally (using Section 8.2.1) **PT Lengths Tested:** (1024) **AAD Lengths tested:** (1024) **IV Lengths Tested:** (8 , 1024) **96BitIV_Supported** **GMAC_Supported** **DRBG:** **A4:** Val# 356 **A5:** Val# 357 **A6:** Val# 380
	2505	2506	2507		**User space and the AES hardware offered by the processor.** **CBC** (128 , 192 , 256)
	2502	2503	2504		**User space and the Gladman AES CBC implementation.** **CBC** (128 , 192 , 256)
	2499	2500	2501		**User space and assembler optimized AES.** **GCM** **KS:** **AES_128** Tag Length(s): 128 120 112 104 96 64 32 **AES_192** Tag Length(s): 128 120 112 104 96 64 32 **AES_256** Tag Length(s): 128 120 112 104 96 64 32 **IV Generated:** Internally (using Section 8.2.2) **PT Lengths Tested:** (1024) **AAD Lengths tested:** (1024) **IV Lengths Tested:** (8 , 1024) **96BitIV_Supported** **GMAC_Supported** **DRBG:** **A4:** Val #353 **A5:** Val #354 **A6:** Val #355

DRBG	356	357	380	SP 800-90A	**User space and generic, non-optimized software.** **CTR_DRBG**: Prediction Resistance Tested: Enabled; BlockCipher_Use_df: (**AES-128)** **AES** **A4** **A5** **A6** Val# 2508 Val# 2509 Val# 2547
	353	354	355		**User space and assembler optimized AES**. **CTR_DRBG**: Prediction Resistance Tested: Enabled; BlockCipher_Use_df: (**AES-128**) **AES** **A4** **A5** **A6** Val# 2499 Val# 2500 Val# 2501
ECDSA	428	429	437	FIPS 186-3 ANSI X9.62	**User space and generic, non-optimized software.** **FIPS186-2:** **PKG: CURVES** (P-256 P-384) **PKV: CURVES** (P-256 P-384) **SIG(gen): CURVES** (P-256 P-384) **SIG(ver): CURVES** (P-256 P-384) **A4** **A5** **A6** **SHS:** Val# 2119 Val# 2120 Val# 2148 **DRBG:** Val# 356 Val# 357 Val# 380
HMAC	1541	1542	1568	FIPS 198	**User space and generic, non-optimized software.** (**Key Sizes:Block Sizes tested: KS<BS KS=BS KS>BS)** **A4** **A5** **A6** **HMAC-SHA1** Val# 2119 Val# 2120 Val# 2148 **HMAC-SHA224** Val# 2119 Val# 2120 Val# 2148 **HMAC-SHA256** Val# 2119 Val# 2120 Val# 2148 **HMAC-SHA384** Val# 2119 Val# 2120 Val# 2148 **HMAC-SHA512** Val# 2119 Val# 2120 Val# 2148
	1589	1591	1593		**User space and optimized software.** (**Key Sizes:Block Sizes tested: KS<BS KS=BS KS>BS)** **A4** **A5** **A6** **HMAC-SHA1** Val# 2168 Val# 2170 Val# 2172 **HMAC-SHA224** Val# 2168 Val# 2170 Val# 2172 **HMAC-SHA256** Val# 2168 Val# 2170 Val# 2172
PBKDF2	N/A	N/A	N/A	SP 800-132	Password based key derivation according to PKCS#5 using HMAC with SHA-1 or SHA-2 as pseudorandom function.

RSA	1289	1290	1302	FIPS 186-3 ANSI X9.31 FIPS 186-2 PKCS#1v1.5	**User space and generic, non-optimized software.** **FIPS186-2:** **ALG[ANSIX9.31]:** Key(gen)(MOD: 1024 , 1536 , 2048 , 3072 , 4096 PubKey Values: 3 , 17 , 65537 DRBG: **A4:** Val# 356 **A5**: Val# 357 **A6**: Val# 380 **ALG[RSASSA-PKCS1_V1_5]:** SIG(gen), SIG(ver): 1024,1536,2048,3072,4096 **SHS: A4 A5 A6** SHA-1 Val# 2119 Val# 2120 Val# 2148 SHA-224 Val# 2119 Val# 2120 Val# 2148 SHA-256 Val# 2119 Val# 2120 Val# 2148 SHA-384 Val# 2119 Val# 2120 Val# 2148 SHA-512 Val# 2119 Val# 2120 Val# 2148
SHS	2119	2120	2148	FIPS 180-3	**User space and generic, non-optimized software.** **SHA-1** (BYTE-only) **SHA-224** (BYTE-only) **SHA-256** (BYTE-only) **SHA-384** (BYTE-only) **SHA-512** (BYTE-only)
	2168 (2121)	2170 (2122)	2172 (2123)		**User space and optimized software.** **SHA-1** (BYTE-only) **SHA-224** (BYTE-only) **SHA-256** (BYTE-only)
TDES	1530	1531	1542	ANSIX9.52-1998 FIPS 46-3 SP 800-67 SP 800-38A Appendix E	**User space and generic, non-optimized software.** **TECB** (KO 1,2) **TCBC** (KO 1,2) **TCFB8** (KO 1,2) **TCFB64** (KO 1,2) **TOFB** (KO 1,2) **CTR** (int only)

Module Name:	CoreCrypto Kernel Module v4.0				(Kernel Space)
Alg.	Platform Certificate			Standards	Description
	A4	A5	A6		
AES	2496	2497	2498	FIPS 197 SP 800-38A	**Kernel space and generic, non-optimized software.** **ECB** (128 , 192 , 256) **CBC** (128 , 192 , 256)
	2493	2494	2495		**Kernel space and assembler optimized AES.** **CBC** (128 , 192 , 256)
DRBG	350	351	352	SP 800-90A	**Kernel space and generic, non-optimized software.** **CTR_DRBG**: Prediction Resistance Tested: Enabled; BlockCipher_Use_df: (**AES-128**) **A4** **A5** **A6** AES Val# 2496 Val# 2497 Val# 2498
ECDSA	425	426	427	FIPS 186-3 ANSI X9.62	**Kernel space and generic, non-optimized software.** **FIPS186-2:** **PKG:** **CURVES** (P-256 P-384) **PKV:** **CURVES** (P-256 P-384) **SIG(gen):** **CURVES** (P-256 P-384) **SIG(ver):** **CURVES** (P-256 P-384) **A4** **A5** **A6** **SHS:** Val# 2113 Val# 2114 Val# 2115 **DRBG:** Val# 350 Val# 351 Val# 352
HMAC	1535	1536	1537	FIPS 198	**Kernel space and generic, non-optimized software.** **(Key Sizes:Block Sizes tested: KS<BS KS=BS KS>BS)** **A4** **A5** **A6** **HMAC-SHA1** Val# 2113 Val# 2114 Val# 2115 **HMAC-SHA224** Val# 2113 Val# 2114 Val# 2115 **HMAC-SHA256** Val# 2113 Val# 2114 Val# 2115 **HMAC-SHA384** Val# 2113 Val# 2114 Val# 2115 **HMAC-SHA512** Val# 2113 Val# 2114 Val# 2115
	1588 (1538)	1590 (1539)	1592 (1540)		**Kernel space and optimized software.** **(Key Sizes:Block Sizes tested: KS<BS KS=BS KS>BS)** **A4** **A5** **A6** **HMAC-SHA1** Val# 2167 Val# 2169 Val# 2171 **HMAC-SHA224** Val# 2167 Val# 2169 Val# 2171 **HMAC-SHA256** Val# 2167 Val# 2169 Val# 2171
PBKDF2	N/A	N/A	N/A	SP 800-132	

RSA	1284	1285	1286	PKCS#1v1.5	**User space and generic, non-optimized software.** **ALG[RSASSA-PKCS1_V1_5]:** SIG(ver): 1024 , 1536 , 2048 , 3072 , 4096 **SHS:** **A4** **A5** **A6** SHA-1 Val# 2113 Val# 2114 Val# 2115 SHA-224 Val# 2113 Val# 2114 Val# 2115 SHA-256 Val# 2113 Val# 2114 Val# 2115 SHA-384 Val# 2113 Val# 2114 Val# 2115 SHA-512 Val# 2113 Val# 2114 Val# 2115
SHS	2113	2114	2115	FIPS 180-3	**Kernel space and generic, non-optimized software.** **SHA-1** (BYTE-only) **SHA-224** (BYTE-only) **SHA-256** (BYTE-only) **SHA-384** (BYTE-only) **SHA-512** (BYTE-only)
	2167 (2116)	2169 (2117)	2171 (2118)		**Kernel space and optimized software.** **SHA-1** (BYTE-only) **SHA-224** (BYTE-only) **SHA-256** (BYTE-only)
TDES	1527	1528	1529	ANSIX9.52-1998 FIPS 46-3 SP 800-67 SP 800-38A Appendix E	**Kernel space and generic, non-optimized software.** **TECB** (KO 1,2) **TCBC** (KO 1,2)