Apple Inc.

Apple corecrypto Kernel Space Module for ARM (ccv10)
FIPS 140-2 Non-Proprietary Security Policy

Module Version 10.0

Prepared for:

Apple Inc.
One Apple Park Way
Cupertino, CA 95014

www.apple.com

Prepared by:

atsec information security Corp.
9130 Jollyville Road, Suite 260
Austin, TX 78759

www.atsec.com

Trademarks

Table of Contents

1 INTRODUCTION 5
1.1 PURPOSE 5
1.2 DOCUMENT ORGANIZATION / COPYRIGHT 5
1.3 EXTERNAL RESOURCES / REFERENCES 5
1.4 ACRONYMS 7

2 CRYPTOGRAPHIC MODULE SPECIFICATION 8
2.1 MODULE DESCRIPTION 8
2.1.1 Module Validation Level 8
2.1.2 Module components 8
2.1.3 Tested Platforms 8
2.2 MODES OF OPERATION 9
2.2.1 Approved Security Functions 10
2.2.2 Non-Approved Security Functions 12
2.3 CRYPTOGRAPHIC MODULE BOUNDARY 14
2.4 MODULE USAGE CONSIDERATIONS 14

3 CRYPTOGRAPHIC MODULE PORTS AND INTERFACES 15

4 ROLES, SERVICES AND AUTHENTICATION 16
4.1 ROLES 16
4.2 SERVICES 16
4.3 OPERATOR AUTHENTICATION 19

5 PHYSICAL SECURITY 20

6 OPERATIONAL ENVIRONMENT 21
6.1 APPLICABILITY 21
6.2 POLICY 21

7 CRYPTOGRAPHIC KEY MANAGEMENT 22
7.1 RANDOM NUMBER GENERATION 22
7.2 KEY / CSP GENERATION 22
7.3 KEY / CSP ESTABLISHMENT 22
7.4 KEY / CSP ENTRY AND OUTPUT 23
7.5 KEY / CSP STORAGE 23
7.6 KEY / CSP ZEROIZATION 23

8 ELECTROMAGNETIC INTERFERENCE/ELECTROMAGNETIC COMPATIBILITY (EMI/EMC) 24

9 SELF-TESTS 25
9.1 POWER-UP TESTS 25
9.1.1 Cryptographic Algorithm Tests 25
9.1.2 Software / firmware integrity tests 25
9.1.3 Critical Function Tests 25
9.2 CONDITIONAL TESTS 25
9.2.1 Continuous Random Number Generator Test 25
9.2.2 Pair-wise Consistency Test 26
9.2.3 SP800-90A Health Tests 26
9.2.4 Critical Function Test 26

10 DESIGN ASSURANCE 27
10.1 CONFIGURATION MANAGEMENT 27
10.2 DELIVERY AND OPERATION 27
10.3 DEVELOPMENT 27
10.4 GUIDANCE 27
10.4.1 Cryptographic Officer Guidance 27
10.4.2 User Guidance 27

11 MITIGATION OF OTHER ATTACKS 28

List of Tables

Table 1 Module Validation Level 8

Table 2 Tested Platforms 9

Table 3 Approved and Vendor Affirmed Security Functions 12

Table 3a Non-Approved but Allowed Security Functions 12

Table 4 Non-Approved or Non-Compliant Security Functions 13

Table 5 Roles 16

Table 6 Approved and Allowed Services in Approved Mode 18

Table 7 Non-Approved Services in Non-Approved Mode 19

Table 8 Keys and CSPs 22

Table 9 Cryptographic Algorithm Tests 25

List of Figures

No table of figures entries found.

1 Introduction

1.1 Purpose

This document is a non-proprietary Security Policy for the Apple corecrypto Kernel Space Module for ARM (ccv10). It describes the module and the FIPS 140-2 cryptographic services it provides. This document also defines the FIPS 140-2 security rules for operating the module.

This document was prepared in fulfillment of the FIPS 140-2 requirements for cryptographic modules and is intended for security officers, developers, system administrators, and end-users.

FIPS 140-2 details the requirements of the Governments of the U.S. and Canada for cryptographic modules, aimed at the objective of protecting sensitive but unclassified information.

For more information on the FIPS 140-2 standard and validation program please refer to the NIST CMVP website [CMVP].

Throughout the document "Apple corecrypto Kernel Space Module for ARM (ccv10)", "cryptographic module", "corecrypto KEXT" or "the module" are used interchangeably to refer to the Apple corecrypto Kernel Space Module for ARM (ccv10). Throughout the document "OS" refers to "iOS", "iPadOS", "tvOS", "watchOS" and "TxFW" unless specifically noted. "ccv10" is used to refer to the module version 10.0.

1.2 Document Organization / Copyright

This non-proprietary Security Policy document may be reproduced and distributed only in its original entirety without any revision, ©2021 Apple Inc.

1.3 External Resources / References

The Apple website (http://www.apple.com) contains information on the full line of products from Apple Inc. For a detailed overview of the operating system iOS and the associated security properties refer to [OS] and [SEC]. For details on the OS releases with their corresponding validated modules and Crypto Officer Role Guides refer to the Apple OS Security Guide in the webpage - "Product security certifications, validations, and guidance for iOS" [Guide]. Additional References are listed below.

CMVP	Cryptographic Module Validation Program https://csrc.nist.gov/projects/cryptographic-module-validation-program
CAVP	Cryptographic Algorithm Validation Program https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program
FIPS 140-2	Federal Information Processing Standards Publication, "FIPS PUB 140-2 Security Requirements for Cryptographic Modules," Issued May-25-2001, Effective 15-Nov-2001, https://csrc.nist.gov/projects/cryptographic-module-validation-program/standards
FIPS 140-2 IG	NIST, "Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program," December 2019 https://csrc.nist.gov/csrc/media/projects/cryptographic-module-validation-program/documents/fips140-2/fips1402ig.pdf
FIPS 180-4	Federal Information Processing Standards Publication 180-4, March 2012, Secure Hash Standard (SHS)
FIPS 186-4	Federal Information Processing Standards Publication 186-4, July 2013, Digital Signature Standard (DSS)
FIPS 197	Federal Information Processing Standards Publication 197, November 26, 2001 Advanced Encryption Standard (AES)
FIPS 198	Federal Information Processing Standards Publication 198, July, 2008 The Keyed-Hash Message Authentication Code (HMAC)

OS	Technical Overview for all Apple Platforms
	https://developer.apple.com/
SEC	Security Overview
	https://developer.apple.com/security/
SP800-38 A	NIST Special Publication 800-38A, "Recommendation for Block Cipher Modes of Operation", December 2001
SP800-38 C	NIST Special Publication 800-38C, "Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality", May 2004
SP800-38 E	NIST Special Publication 800-38E, "Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices", January 2010
SP800-38 F	NIST Special Publication 800-38F, "Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping", December 2012
SP800-57P1	NIST Special Publication 800-57, "Recommendation for Key Management – Part 1: General", July 2016
SP800-67	NIST Special Publication 800-67, "Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher", (Revised) January 2012
SP800-90A	NIST Special Publication 800-90A, "Recommendation for Random Number Generation Using Deterministic Random Bit Generators
SP800-132	NIST Special Publication 800-132, "Recommendation for Password-Based Key Derivation", December 2010
Guide	User Guidance and CO Guidance:
	https://support.apple.com/HT211006
	https://support.apple.com/HT202739
	https://support.apple.com/HT208389
	https://support.apple.com/HT208390
	https://support.apple.com/HT208675

1.4 Acronyms

AES	Advanced Encryption Standard
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining mode of operation
CFB	Cipher Feedback mode of operation
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CTR	Counter mode of operation
DES	Data Encryption Standard
DRBG	Deterministic Random Bit Generator
ECB	Electronic Codebook mode of operation
ECC	Elliptic Curve Cryptography
ECDSA	DSA based on ECC
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FIPS	Federal Information Processing Standard
GCM	Galois/Counter Mode
HMAC	Hash-Based Message Authentication Code
KAT	Known Answer Test
KEXT	Kernel extension
KDF	Key Derivation Function
KPI	Kernel Programming Interface
MAC	Message Authentication Code
NIST	National Institute of Standards and Technology
OFB	Output Feedback (mode of operation)
PCT	Pair-wise Consistency Test
RNG	Random Number Generator
SHS	Secure Hash Standard
Triple-DES	Triple Data Encryption Standard

2 Cryptographic Module Specification

2.1 Module Description

The Apple corecrypto Kernel Space Module for ARM (ccv10) is a software cryptographic module running on a multi-chip standalone mobile device.

The cryptographic services provided by the module are:

- Data encryption / decryption
- Generation of hash values
- Message authentication
- Signature generation/verification
- Random number generation
- Key generation

2.1.1 Module Validation Level

The module is intended to meet requirements of FIPS 140-2 security level 1 overall. The following table shows the security level for each of the eleven requirement areas of the validation.

FIPS 140-2 Security Requirement Area	Security 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	1

Table 1 Module Validation Level

2.1.2 Module components

There are no components excluded from the validation testing of the Apple corecrypto Kernel Space Module for ARM (ccv10).

2.1.2.1 Software components

corecrypto has a KPI layer that provides consistent interfaces to the supported algorithms. These implementations include proprietary optimizations of algorithms that are fitted into the corecrypto framework. The corecrypto KEXT is linked dynamically into the OS kernel.

2.1.2.2 Hardware components

There are no hardware components within the cryptographic module boundary.

2.1.3 Tested Platforms

The module has been tested with and without PAA on the following hardware platforms. PAA=NEON is present in Apple A, S and T series processors.

Manufacturer	Operating System	Processor	Hardware Platform
Apple Inc.	iOS 13	Apple A9	iPhone 6S Plus
		Apple A10 Fusion	iPhone 7 Plus
		Apple A11 Bionic	iPhone 8 Plus
		Apple A12 Bionic	iPhone Xs Max
		Apple A13 Bionic	iPhone 11 Pro Max
	iPadOS 13	Apple A8	iPad mini 4
		Apple A8X	iPad Air 2
		Apple A9	iPad (5th generation)
		Apple A9X	iPad Pro (9.7 inch)
		Apple A10 Fusion	iPad (6th generations)
		Apple A10X Fusion	iPad Pro (12.9 inch, 2nd generation)
		Apple A12 Bionic	iPad mini (5th generation)
		Apple A12X Bionic	iPad Pro (12.9 inch, 3rd generation)
	tvOS 13	Apple A10X Fusion	Apple TV 4K
	watchOS 6	Apple S1P	Apple Watch Series 1
		Apple S3	Apple Watch Series 3
		Apple S4	Apple Watch Series 4
		Apple S5	Apple Watch Series 5
	TxFW 10.15	Apple T2	Apple T2[1]

Table 2 Tested Platforms

In addition to the configurations tested by the laboratory, vendor-affirmed testing was performed on the following platforms.

for iOS13:
- iPhone 6s and iPhone SE with an Apple A9
- iPhone 7 with an Apple A10 Fusion
- iPhone 8 and iPhone X with an Apple A11 Bionic
- iPhone Xr and iPhone Xs with an Apple A12 Bionic
- iPhone 11 and iPhone 11 Pro with an Apple A13 Bionic

for iPadOS 13
- iPad Pro (12.9) with an Apple A9X
- iPad (7th generation) with an Apple A10 Fusion
- iPad Pro (10.5-inch) with an Apple A10X Fusion
- iPad Air (3rd generation) with an Apple A12 Bionic
- iPad Pro (11-inch) with an Apple A12X Bionic

CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys when so ported if the specific operational environment is not listed on the validation certificate (IG G.5).

2.2 Modes of operation

The Apple corecrypto Kernel Space Module for ARM (ccv10) has an Approved and Non-Approved Mode of operation. The Approved Mode of operation is configured in the system by default. If the device starts up successfully then corecrypto KEXT has passed all self-tests and is operating in the Approved Mode. Any

[1] The user for Apple T2 are iMac Pro, Mac Pro, Mac mini, MacBook Air and MacBook Pro

calls to the Non-Approved security functions listed in Table 4 will cause the module to assume the Non-Approved Mode of operation.

The module transitions back into FIPS mode immediately when invoking one of the approved ciphers as all keys and Critical Security Parameters (CSP) handled by the module are ephemeral and there are no keys and CSPs shared between any functions. A re-invocation of the self-tests or integrity tests is not required. Even when using this FIPS 140-2 non-approved mode, the module configuration ensures that the self-tests are always performed during initialization of the module. This includes the hardware-assisted AES (AES-NI) and SHA implementations.

The module contains multiple implementations of the same cipher as listed Table 3. If multiple implementations of the same cipher are present, the module selects automatically which cipher is used based on internal heuristics.

2.2.1 Approved Security Functions

Approved security functions are listed in Table 3. The Algorithm Certificate Numbers listed in Table 3 are obtained from NIST for successful validation testing of the cryptographic algorithms implementations of the module that runs on the hardware platforms listed in Table 2. For the current standards, test requirements, and special abbreviations used in the following tables, please refer to [CAVP]. Not all the algorithms/modes verified through CAVP are available for use by the module.

Cryptographic Function	Standards and Algorithm	Modes and Options	Algorithm Certificate Number
Random Number Generation	[SP800-90A] DRBG	HMAC_DRBG Modes: HMAC-SHA-384 HMAC-SHA-512 Without Prediction Resistance	A16 (c_ltc)
		CTR_DRBG Modes: AES-128 AES-256 Derivation Function Enabled Without Predication Resistance	A15 (c_asm) A13 (vng_asm)
		HMAC_DRBG Modes: HMAC-SHA-1 HMAC-SHA-384 HMAC-SHA-224 HMAC-SHA-512 HMAC-SHA-256 Without Predication Resistance	A18 (vng_ltc)
Symmetric Encryption and Decryption	[FIPS 197] AES [SP800-38 A] [SP800-38 E] [SP800-38 F]	Key Length: 128, 192, 256 Modes CBC ECB CFB8 OFB CFB128 CTR	A15 (c_asm)
		Key Length: 128, 192, 256 Modes: CCM GCM CTR ECB	A13 (vng_asm)

Cryptographic Function	Standards and Algorithm	Modes and Options	Algorithm Certificate Number
		Key Length: 128, 192, 256 Modes: CBC CFB128 ECB OFB XTS (Key length: 128, 256 bits key)	A14 (asm_arm)
	[SP800-67] Triple-DES	Keying Option: 1; All Keys Independent Modes: ECB	A16 (c_ltc)
Key Wrapping	SP800-38D	Key Length: 128, 192, 256 Modes: AES-CCM AES-GCM	A13 (vng_asm)
	SP800-38F	Key Length: 128, 192, 256 Modes AES-KW	A15 (c_asm)
Digital Signature	[FIPS186-4] RSA	Signature Generation (PKCS#1 v1.5 and PSS) Modulus: 2048, 3072, 4096 Signature Verification (PKCS#1 v1.5 and PSS) Modulus: 1024, 2048, 3072, 4096	A18 (vng_ltc)
	[FIPS 186-4] ECDSA ANSI X9.62	Key Pair Generation (PKG): P-224, P-256, P-384, P-521 Public Key Validation (PKV) P-224, P-256, P-384, P-521 Signature Generation: P-224, P-256, P-384, P-521 Signature Verification: P-224, P-256, P-384, P-521	A18 (vng_ltc)
Message Digest	[FIPS 180-4] SHS	Modes SHA-384 SHA-512	A16 (c_ltc)
		Modes SHA-1 SHA-224 SHA-256 SHA-384 SHA-512	A18(vng_ltc)
		Modes SHA-256	A17[2] (vng_neon)
Keyed Hash	[FIPS 198] HMAC	Modes HMAC-SHA-384 HMAC-SHA-512	A16 (c_ltc)
		Modes HMAC-SHA-256	A17[2] (vng_neon)

[2] The S1P and S3 from the armv7 processor family do not implement vng_neon and do not have the A17 ACVT certificate.

Cryptographic Function	**Standards and Algorithm**	**Modes and Options**	**Algorithm Certificate Number**
		Modes HMAC-SHA-1 HMAC-SHA-384 HMAC-SHA-224 HMAC-SHA-512 HMAC-SHA-256	A18(vng_ltc)
CKG	[SP800-133]	ECDSA Key Pair Generation: curves P-224, P-256, P-384, P-521	Vendor Affirmed

Table 3 Approved and Vendor Affirmed Security Functions

Cryptographic Function	**Standards and Algorithm**	**Modes and Options**	**Algorithm Certificate Number**
RSA Key Wrapping	Non-[SP 800-56B], [SP800-131A]	KTS RSA-OAEP PKCS#1 v1.5 Modulus size: 2048, 3072 and 4096 bits	Non-Approved, but Allowed
MD5 (Used as part of the TLS key establishment scheme only)	Message Digest	Digest size 128 bit	Non-Approved, but Allowed:
NDRNG	Random Number Generation	N/A	Non-Approved, but Allowed: provided by the underlying operational environment

Table 3a Non-Approved but Allowed Security Functions

2.2.2 Non-Approved Security Functions

Cryptographic Function	**Usage / Description**	**Note**
DES	Encryption / Decryption: Key Size 56 bits	Non-Approved
MD2	Message Digest Digest size 128 bit	Non-Approved
MD4	Message Digest Digest size 128 bit	Non-Approved
RIPEMD	Message Digest Digest size 160	Non-Approved
Ed25519	Key Agreement Signature Generation Signature Verification	Non-Approved
ANSI X9.63 KDF	ANSI X9.63 Hash Based KDF	Non-Approved
RFC6637 KDF	KDF based on RFC6637	Non-Approved
ECDSA	Key Pair Generation for compact point representation of points	Non-Approved

Cryptographic Function	**Usage / Description**	**Note**
Integrated Encryption Scheme on elliptic curves	Encryption / Decryption	Non-Approved
ECDSA	PKG: Curve P-192 PKV: Curve P-192 Signature Generation: Curve P-192 Signature Verification: Curve P-192	Non-Approved
CAST5	Encryption / Decryption Key Sizes 40 to 128 bits in 8-bit increments	Non-Approved
Blowfish	Encryption / Decryption	Non-Approved
RC2	Encryption / Decryption	Non-Approved
RC4	Encryption / Decryption	Non-Approved
OMAC (One-Key CBC MAC)	MAC generation	Non-Approved
RSA Key Wrapping	PKCS#1 v1.5 and KST RSA-OAEP Key Size < 2048	Non-Approved
RSA	PKCS#1 v1.5 and PSS Signature Generation: Key Size < 2048 Key Size: 1024-4096 bits in multiple of 32 bits not listed in table 3 Signature Verification Key Size (modulus) < 1024 Key Size: 1024-4096 bits in multiple of 32 bits not listed in table 3	Non-Approved
	ANSI X9.31 Signature Generation Key Size < 2048 Key Size: 1024-4096 bits in multiple of 32 bits not listed in table 3 Signature Verification Key Size < 1024 Key Size: 1024-4096 bits in multiple of 32 bits not listed in table 3	
RSA Signature Verification	PKCS#1 v1.5 Key sizes (modulus): 1536 bits	Non-Compliant
SP800-56C	KDF	Non-Compliant
Triple-DES Symmetric Encryption and Decryption	Optimized Assembler (asm_arm) Implementation: Encryption / Decryption Mode: CTR	Non-Compliant
	Encryption / Decryption: Two-Key implementation	
AES-CMAC	AES-128/192/256 MAC generation /verification	Non-Compliant

Table 4 Non-Approved or Non-Compliant Security Functions

Note: A Non-Approved function in Table 4 is that the function implements a non-Approved algorithm, while a Non-Compliant function is that the function implements an Approved algorithm, but the implementation is either not validated by the CAVP or/and the self-tests are not implemented (IG 9.4).

2.3 Cryptographic Module Boundary

The physical boundary of the module is the physical boundary of the iPhone, iPad, Apple TV, Apple Watch or T2 running iOS, iPadOS, tvOS, watchOS or TxFW respectively. Consequently, the embodiment of the module is a multi-chip standalone cryptographic module.

The logical module boundary is depicted in the logical block diagram given in Figure 1.

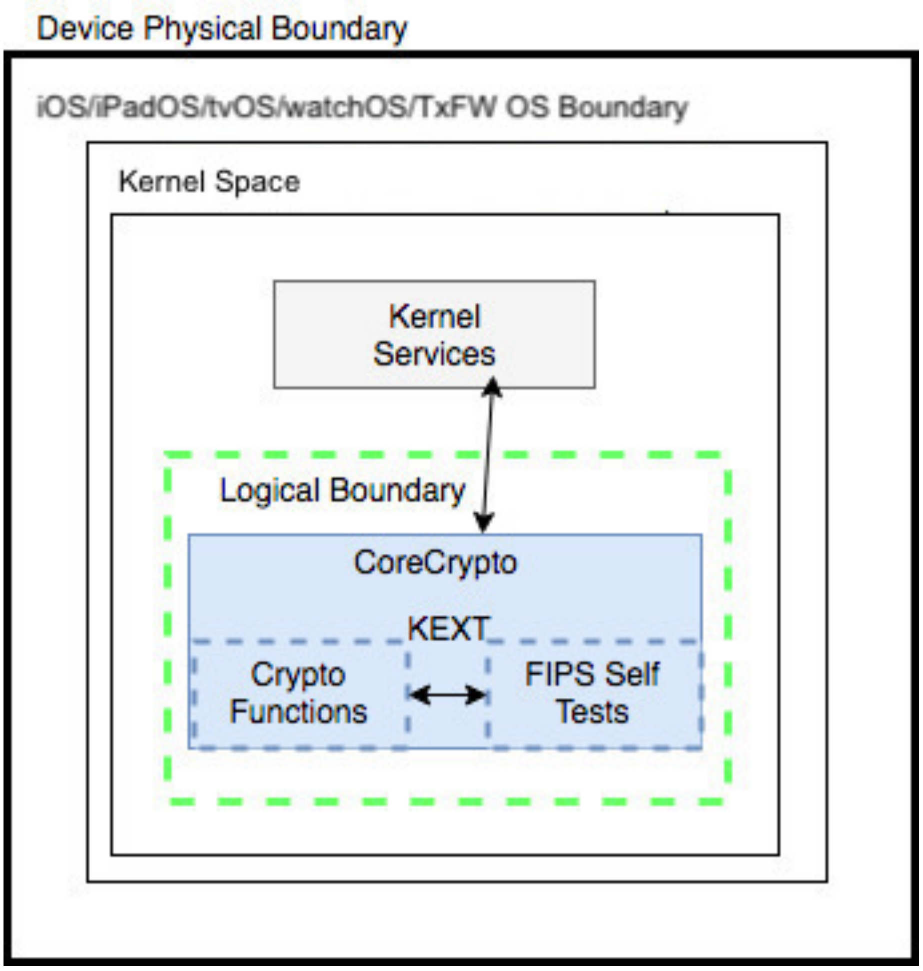

Figure 1: Logical Block Diagram

2.4 Module Usage Considerations

A user of the module must consider the following requirements and restrictions when using the module:

- AES-GCM IV is constructed in accordance with SP800-38D section 8.2.1 in compliance with IG A.5 scenario 1. The GCM IV generation follows RFC 5288 and shall only be used for the TLS protocol version 1.2. Users should consult [SP 800-38D], especially section 8, for all of the details and requirements of using AES-GCM mode In case the module's power is lost and then restored, the key used for the AES GCM encryption/decryption shall be re-distributed.
- AES-XTS mode is only approved for hardware storage applications. The length of the AES-XTS data unit does not exceed 2^{20} blocks.
- In order to meet the IG A.13 requirement, the same Triple-DES key shall not be used to encrypt more than 2^{16} 64-bit blocks of data.
- When using AES, the caller must obtain a reference to the cipher implementation via the functions of ccaes_[cbc|ecb]_[encrypt|decrypt]_mode.
- When using SHA, the caller must obtain a reference to the cipher implementation via the functions ccsha[1|224|256|384|512]_di.

3 Cryptographic Module Ports and Interfaces

The underlying logical interfaces of the module are the C Language Kernel Programming Interfaces (KPIs). In detail, these interfaces are the following:

- Data input and data output are provided in the variables passed in the KPI and callable service invocations, generally through caller-supplied buffers. Hereafter, KPIs and callable services will be referred to as "KPI."
- Control inputs which control the mode of the module are provided through dedicated parameters, namely the kernel module plist whose information is supplied to the module by the kernel module loader.
- Status output is provided in return codes and through messages. Documentation for each KPI lists possible return codes. A complete list of all return codes returned by the C language KPIs within the module is provided in the header files and the KPI documentation. Messages are documented also in the KPI documentation.

The module is a kernel extension optimized for library use within the OS kernel and does not contain any terminating assertions or exceptions. Once the module is loaded into the OS kernel its cryptographic functions are made available to OS kernel services only. Any internal error detected by the module is reflected back to the caller with an appropriate return code. The calling OS Kernel service must examine the return code and act accordingly. There is one notable exceptions: ECDSA does not return a key if the pair-wise consistency test fails.

The function executing FIPS 140-2 module self-tests does not return an error code but causes the system to crash if any self-test fails – see Section 9.

The module communicates error status synchronously through the use of documented return codes (indicating the module's status). It is the responsibility of the caller to handle exceptional conditions in a FIPS 140-2 appropriate manner.

Caller-induced or internal errors do not reveal any sensitive material to callers.

Cryptographic bypass capability is not supported by the module.

4 Roles, Services and Authentication

This section defines the roles, services and authentication mechanisms and methods with respect to the applicable FIPS 140-2 requirements.

4.1 Roles

The module supports a single instance of the two authorized roles: the Crypto Officer and the User. No support is provided for multiple concurrent operators or a maintenance operator.

Role	**General Responsibilities and Services (details see below)**
User	Utilization of services (section 4.2) of the module tested platforms (section 2.1)
Crypto Officer (CO)	Utilization of services (section 4.2) of the module tested platforms (section 2.1)

Table 5 Roles

4.2 Services

The module provides services to authorized operators of either the User or Crypto Officer Role according to the applicable FIPS 140-2 security requirements.

Table 6 contains the cryptographic functions employed by the module in the Approved Mode. For each available service it lists, the associated role, the Critical Security Parameters (CSPs) and cryptographic keys involved, and the type(s) of access to the CSPs and cryptographic keys.

CSPs contain security-related information (secret and private cryptographic keys, for example) whose disclosure or modification can compromise the main security objective of the module, namely the protection of sensitive information.

The access types are denoted as follows:

- R: the item is read/ execute or referenced by the service
- W: the item is written or updated by the service
- Z: the persistent item is zeroized by the service

Service	**Roles**		**CSPs and crypto keys**	**Access Type**
	USER	CO		
Triple-DES encryption and decryption Encryption *Input:* plaintext, IV, key *Output:* ciphertext Decryption *Input:* ciphertext, IV, key *Output:* plaintext	X	X	Triple-DES key	R
AES encryption and decryption Encryption *Input:* plaintext, IV, key *Output:* ciphertext Decryption *Input:* ciphertext, IV, key *Output:* plaintext	X	X	AES key	R

Service	Roles		CSPs and crypto keys	Access Type
	USER	CO		
AES Key Wrapping Encryption *Input:* plaintext, key *Output:* ciphertext Decryption *Input:* ciphertext, key *Output:* plaintext	X	X	AES key	R
RSA Key Wrapping Using PKCS#1 v1.5 and KST RSA-OAEP, (non-approved but allowed) Encryption *Input*: plaintext, the modulus n, the public key e *Output*: ciphertext Decryption *Input*: ciphertext, the modulus n, the private key d *Output*: plaintext	X	X	RSA key pair	R
Secure Hash Generation using SHA1, SHA-224, SHA-256, SHA-384, or SHA-512 *Input:* message *Output:* message digest	X	X	None	N/A
Secure Hash Generation using MD5 (non-approved but allowed)	X	X	none	N/A
HMAC generation using HMAC-SHA1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, or HMAC-SHA-512 *Input:* HMAC key, message *Output:* HMAC value of message	X	X	Secret HMAC key	R
RSA signature generation and verification Signature generation *Input:* the modulus n, the private key d, the SHA algorithm (SHA -224/ SHA-256/ SHA-384 /SHA-512), a message m to be signed *Output:* the signature s of the message Signature verification *Input:* the modulus n, the public key e, the SHA algorithm (SHA-1/ SHA -224/ SHA-256/ SHA-384 /SHA-512), a message m, a signature for the message *Output:* pass if the signature is valid, fail if the signature is invalid	X	X	RSA key pair	R

Service	**Roles**		**CSPs and crypto keys**	**Access Type**
	USER	CO		
ECDSA signature generation and verification Signature generation *Input:* message m, q, a, b, X_G, Y_G, n, the SHA algorithm (SHA-224/ SHA-256/ SHA-384/ SHA-512), sender's private key d *Output:* signature of m as a pair of r and s Signature verification *Input:* received message m', signature in form on r' and s' pair, q, a, b, X_G, Y_G, n, sender's public key Q, he SHA algorithm (SHA-1/ SHA-224/ SHA-256/ SHA-384/ SHA-512) *Output:* pass if the signature is valid, fail if the signature is invalid	X	X	ECDSA key pair	R
Random number generation *Input:* Entropy Input, Nonce, Personalization String *Output:* Returned Bits	X	X	Entropy input string, Nonce, V and Key	R W
ECDSA key pair generation Input: q, FR, a, b, domain_parameter_seed, G, n, h. Output: private key d, public key Q	X	X	ECDSA key pair	W
Release all resources of symmetric crypto function context (i.e. Symmetric Key Zeroization) *Input:* context Output: N/A	X	X	AES / Triple-DES key	Z
Release all resources of hash context (i.e. MAC Key Zeroization) *Input:* context Output: N/A	X	X	HMAC key	Z
Release all resources of asymmetric crypto function context (i.e. Asymmetric Key Zeroization) *Input:* context Output: N/A	X	X	Asymmetric keys (RSA/ ECDSA)	Z
Reboot (implicit Power-on Self-test) Input: N/A *Output:* pass if the Self-test is successful, fail if the Self-test is unsuccessful	X	X	None	N/A
Show Status Input: N/A *Output:* Status of module	X	X	None	N/A

Table 6 Approved and Allowed Services in Approved Mode

Service		**Roles**	
		User	**CO**
Integrated Encryption Scheme on elliptic curves encryption / decryption		X	X
DES Encryption / Decryption		X	X
Triple-DES Encryption/ Decryption	asm_arm implementation CTR mode (non-compliant)	X	X
	Two-Key Triple-DES (non-approved)		
CAST5 Encryption /Decryption		X	X
Blowfish Encryption /Decryption		X	X

Service	**Roles**	
	User	**CO**
RC2 Encryption /Decryption	X	X
RC4 Encryption /Decryption	X	X
MD2 Message Digest Generation	X	X
MD4 Message Digest Generation	X	X
RIPEMD Message Digest Generation	X	X
RSA PKCS#1 (v1.5 and PSS) Signature Generation / Verification Key size: 1024-4096 bits in multiple of 32 bits not listed in table 3	X	X
RSA ANSI X9.31 Signature Generation, Signature Verification Key size (modulus): 1024-4096 bits in multiple of 32 bits not listed in table 3 Public key exponent values: 65537 or larger	X	X
RSA PKCS#1 (v1.5 and PSS) and ANSI X9.31 Signature Generation, Key Size < 2048 Signature Verification, Key Size < 1024	X	X
RSA PKCS#1 v1.5 and KST RSA-OAEP Key Wrapping Key size < 2048	X	X
RSA PKCS#1 v1.5 Signature Verification Key size= 1536 bits	X	X
ECDSA Key Pair Generation for compact point representation of points	X	X
ECDSA Key Generation: Curve P-192 Key Verification: Curve P-192 Signature Generation: Curve P-192 Signature Verification: Curve P-192	X	X
Ed25519 Key agreement/ Signature Generation/ Signature Verification	X	X
SP800-56C Key Derivation	X	X
ANSI X9.63 Hash Based Key Derivation	X	X
RFC6637 Key Derivation	X	X
AES-CMAC MAC Generation / Verification	X	X
OMAC MAC Generation	X	X

Table 7 Non-Approved Services in Non-Approved Mode

4.3 Operator authentication

Within the constraints of FIPS 140-2 level 1, the module does not implement an authentication mechanism for operator authentication. The assumption of a role is implicit in the action taken. The module relies upon the operating system for any operator authentication.

5 Physical Security

The Apple corecrypto Kernel Space Module for ARM (ccv10) is a software module intended to operate on a multi-chip standalone platform. The FIPS 140-2 physical security requirements do not apply to this module since it is a software module.

6 Operational Environment

6.1 Applicability

The Apple corecrypto Kernel Space Module for ARM (ccv10) operates in a modifiable operational environment per FIPS 140-2 level 1 specifications. The module is included in the OS executing on the hardware specified in section 2.1.3.

6.2 Policy

The operating system is restricted to a single operator (single-user mode; concurrent operators are explicitly excluded).

FIPS Self-Test functionality is invoked along with mandatory FIPS 140-2 tests when the module is loaded into memory by the operating system.

7 Cryptographic Key Management

The following Table summarizes the cryptographic keys and CSPs used in the Apple corecrypto Kernel Space Module for ARM (ccv10) with the ley lengths supported, the available methods for key generation, key entry and key output and zeroization.

Name	Key / CSP Size	Generation	Entry / Output	Zeroization
AES Keys	128, 192, 256 bits	N/A. Supplied by the caller	Entry : calling application (see 7.4) Output: N/A	automatic zeroization when structure is deallocated or when the system is powered down (see 7.6).
Triple-DES Keys	192 bits			
HMAC Keys	min 112 bits			
RSA key pair	2048, 3072, 4096			
ECDSA key pair	P-224, P-256, P-384, P-521 curves	The private keys are generated using FIPS186-4 Key Generation method, and the random value used in the key generation is generated using SP800-90A DRBG.	Entry : calling application (see 7.4) Output: calling application (see 7.4)	
Entropy input string		Obtained from the NDRNG	Entry: OS Output: N/A	
DRBG nonce		Obtained from the NDRNG		
DRBG V, Key		Derived from DRBG input string as defined by SP800-90A	Entry: N/A Output: N/A	

Table 8 Keys and CSPs

7.1 Random Number Generation

The module uses a FIPS 140-2 approved deterministic random bit generator based on a block cipher as specified in NIST SP 800-90A. The default Approved DRBG used for random number generation is a CTR_DRBG using AES-256 with derivation function and without prediction resistance. The module also employs a HMAC_DRBG for random number generation. The deterministic random bit generators are seeded by the read_random interface. The read_random is the Kernel Space interface that receives random bits from an entropy source composed by a Fortuna PRNG and the NDRNG from the ARM-based processor. The entropy source provides 256-bits of security strength in seeding and reseeding the module approved DRBGs.

7.2 Key / CSP Generation

The following approved key generation methods are used by the module:

- The module does not implement symmetric key generation.
- In accordance with FIPS 140-2 IG D.12, the cryptographic module performs Cryptographic Key Generation (CKG)) for asymmetric keys as per [SP 800-133] (vendor affirmed) compliant with [FIPS 186-4] and using DRBG compliant with [SP 800-90A]. A seed (i.e., the random value) used in asymmetric key generation is obtained from [SP 800-90A] CTR_DRBG. The generated seed is an unmodified output from the DRBG. The module does not output any information or intermediate results during the key generation process. The DRBG itself is single-threaded.

7.3 Key / CSP Establishment

The module provides the following key establishment services in the Approved Mode:

- the AES Key wrapping using KW, CCM and GCM modes.
- The RSA key wrapping using PKCS#1 v1.5 or RSA-OAEP is non-approved but allowed method.

The encryption strengths for the key establishment methods are determined in accordance with FIPS 140-2 Implementation Guidance IG 7.5 and NIST [SP 800-57 (Part1)].

- AES key wrapping is used for key establishment methodology that provides between 128 and 256 bits of encryption strength.
- RSA key wrapping is used for key establishment methodology that provides between 112 and 152 bits of encryption strength.

7.4 Key / CSP Entry and Output

All keys are entered from, or output to, the invoking kernel service running on the same device. All keys entered into the module are electronically entered in plain text form. Keys are output from the module in plain text form if required by the calling kernel service. The same holds for the CSPs.

7.5 Key / CSP Storage

The Apple corecrypto Kernel Space Module for ARM (ccv10) considers all keys in memory to be ephemeral. They are received for use or generated by the module only at the command of the calling kernel service. The same holds for CSPs.

The module protects all keys, secret or private, and CSPs through the memory protection mechanisms provided by the OS, including the separation between the kernel and user-space. No process can read the memory of another process. No user-space application can read the kernel memory.

7.6 Key / CSP Zeroization

Keys and CSPs are zeroized when the appropriate context object is destroyed or when the device is powered down. Additionally, the user can zeroize the entire device directly (locally) or remotely, returning it to the original factory settings.

8 Electromagnetic Interference/Electromagnetic Compatibility (EMI/EMC)

The EMI/EMC properties of the Apple corecrypto Kernel Space Module for ARM (ccv10) are not meaningful for the software library. The devices containing the software components of the module have their own overall EMI/EMC rating. The validation test environments have FCC, part 15, Class B rating.

9 Self-Tests

FIPS 140-2 requires that the module perform self-tests to ensure the integrity of the module and the correctness of the cryptographic functionality at start up. In addition, the random bit generator requires continuous verification. The FIPS Self Tests functionality runs all required module self-tests. This functionality is invoked by the OS Kernel startup process upon device initialization. If the self-tests succeed, the Apple corecrypto Kernel Space Module for ARM (ccv10) instance is maintained in the memory of the OS Kernel on the device and made available to each calling kernel service without reloading.

All self-tests performed by the module are listed and described in this section.

9.1 Power-Up Tests

The following tests are performed each time the Apple corecrypto Kernel Space Module for ARM (ccv10) starts and must be completed successfully for the module to operate in the FIPS Approved Mode. If any of the following tests fails, the device shuts down automatically. To run the self-tests on demand, the user may reboot the device.

9.1.1 Cryptographic Algorithm Tests

Algorithm	Modes	Test
Triple-DES	ECB	KAT (Known Answer Test) Separate encryption / decryption operations are performed
AES implementations selected by the module for the corresponding environment AES-128	ECB, CBC, XTS, GCM, CCM	KAT Separate encryption / decryption operations are performed
DRBG (CTR_DRBG and HMAC_DRBG; tested separately)	N/A	KAT
HMAC-SHA implementations selected by the module for the corresponding environment HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-512	N/A	KAT
ECDSA	Signature Generation, Signature Verification	PCT
RSA	Signature Generation, Signature Verification	KAT

Table 9 Cryptographic Algorithm Tests

9.1.2 Software / firmware integrity tests

A software integrity test is performed on the runtime image of the Apple corecrypto Kernel Space Module for ARM (ccv10). The corecrypto's HMAC-SHA256 is used as an Approved algorithm for the integrity test. If the test fails, then the device powers itself off.

9.1.3 Critical Function Tests

No other critical function test is performed on power up.

9.2 Conditional Tests

The following sections describe the conditional tests supported by the Apple corecrypto Kernel Space Module for ARM (ccv10).

9.2.1 Continuous Random Number Generator Test

The Apple corecrypto Kernel Space Module for ARM (ccv10) performs a continuous random number generator test on the noise source (i.e., NDRNG), whenever it is invoked to seed the SP800-90A DRBG.

9.2.2 Pair-wise Consistency Test

The Apple corecrypto Kernel Space Module for ARM (ccv10) generates asymmetric ECDSA key pairs and performs all required pair-wise consistency tests with the newly generated key pairs.

9.2.3 SP800-90A Health Tests

The Apple corecrypto Kernel Space Module for ARM (ccv10) performs the health tests as specified in section 11.3 of [SP800-90A].

9.2.4 Critical Function Test

No other critical function test is performed conditionally.

10 Design Assurance

10.1 Configuration Management

Apple manages and records source code and associated documentation files by using the revision control system named “Git.”

Apple module hardware data, which includes descriptions, parts data, part types, bills of materials, manufacturers, changes, history, and documentation are managed and recorded. Additionally, configuration management is provided for the module’s FIPS documentation.

The following naming/numbering convention for documentation is applied.

<evaluation>_<module>_<os>_<mode>_<doc name>_<doc version (##.##)>

Example: FIPS_CORECRYPTO_IOS_tvOS_KS_SECPOL_4.0

Document management utilities provide access control, versioning, and logging. Access to the Git repository (source tree) is granted or denied by the server administrator in accordance with company and team policy.

10.2 Delivery and Operation

The corecrypto KEXT is built into the OS. For additional assurance, it is digitally signed. The Approved Mode is configured by default and can only be transitioned into the non-Approved mode by calling one of the non-Approved algorithms listed in Table 4. This transition is only temporary as the module returns to the Approved mode after that call.

10.3 Development

The Apple crypto module (like any other Apple software) undergoes frequent builds utilizing a “train” philosophy. Source code is submitted to the Build and Integration group (B & I). Integration (merge) can only take place after the corecrypto project successfully passes internal integration and system tests on all platforms. B & I builds, integrates and does basic sanity checking on the operating systems and apps that they produce. Copies of older versions are archived offsite in underground granite vaults.

10.4 Guidance

The following guidance items are to be used for assistance in maintaining the module’s validated status while in use.

10.4.1 Cryptographic Officer Guidance

The Approved Mode of operation is configured in the system by default and can only be transitioned into the non-Approved mode by calling one of the non-Approved algorithms listed in Table 4. If the device starts up successfully then corecrypto KEXT has passed all self-tests and is operating in the Approved Mode.

10.4.2 User Guidance

As above, the Approved Mode of operation is configured in the system by default and can only be transitioned into the non-Approved mode by calling one of the non-Approved algorithms listed in Table 4. If the device starts up successfully then corecrypto KEXT has passed all self-tests and is operating in the Approved Mode.

Kernel programmers that use the module API shall not attempt to invoke any API call directly and only adhere to defined interfaces through the kernel framework.

11 Mitigation of Other Attacks

The module protects against the utilization of known Triple-DES weak keys. The following keys are not permitted:

```
{0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01},
{0xFE,0xFE,0xFE,0xFE,0xFE,0xFE,0xFE,0xFE},
{0x1F,0x1F,0x1F,0x1F,0x0E,0x0E,0x0E,0x0E},
{0xE0,0xE0,0xE0,0xE0,0xF1,0xF1,0xF1,0xF1},
{0x01,0xFE,0x01,0xFE,0x01,0xFE,0x01,0xFE},
{0xFE,0x01,0xFE,0x01,0xFE,0x01,0xFE,0x01},
{0x1F,0xE0,0x1F,0xE0,0x0E,0xF1,0x0E,0xF1},
{0xE0,0x1F,0xE0,0x1F,0xF1,0x0E,0xF1,0x0E},
{0x01,0xE0,0x01,0xE0,0x01,0xF1,0x01,0xF1},
{0xE0,0x01,0xE0,0x01,0xF1,0x01,0xF1,0x01},
{0x1F,0xFE,0x1F,0xFE,0x0E,0xFE,0x0E,0xFE},
{0xFE,0x1F,0xFE,0x1F,0xFE,0x0E,0xFE,0x0E},
{0x01,0x1F,0x01,0x1F,0x01,0x0E,0x01,0x0E},
{0x1F,0x01,0x1F,0x01,0x0E,0x01,0x0E,0x01},
{0xE0,0xFE,0xE0,0xFE,0xF1,0xFE,0xF1,0xFE},
{0xFE,0xE0,0xFE,0xE0,0xFE,0xF1,0xFE,0xF1}.
```