SZ DJI Technology Co., Ltd.

DJI

Core Crypto Engine

Non-Proprietary FIPS 140-2 Security Policy

Version: 1.1

Date: October 21, 2022

Table of Contents

List of Tables

List of Figures

1 Introduction

This document defines the Security Policy for the SZ DJI Technology Co., Ltd. (hereafter, "DJI") Core Crypto Engine module, hereafter denoted the Module. The Core Crypto Engine is a security engine with root of trust and cryptographic accelerator capabilities. It is intended for use in an SOC (System on Chip), where it provides foundational security services for the entire platform, including cryptography, key management, platform identity, secure boot, and secure Life Cycle State (LCS). It offers high-throughput cryptography engines suitable for a diverse set of use cases, such as secure playback of DRM (Digital Rights Management) protected content, drive encryption and more. The module was tested on a DJI H6 SoC, which belongs to the Eagle series of chips.

Table 1 – Cryptographic Module Configurations

Module	HW Version	FW Version	Tested Configuration
Core Crypto Engine	0xDF	TEE 1.1.0, REE 1.1.0 TEE Secure Boot ROM 1.0.0	Linux 4.9 running on a DJI H6 SoC with an ARM Cortex A7

The FIPS 140-2 security levels for the Module are as follows:

Table 2 – Security Level of Security Requirements

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

1.1 Module Description and Cryptographic Boundary

The physical form of the Module is depicted in Figure 1. The Module is a firmware-hybrid, sub-chip module in a single-chip embodiment. The module includes hardware, defined in the RTL (Register Transfer Language), and firmware for execution on the host CPU, making it a hybrid module. The module's RTL is offered for integration as part of a silicon partner's hardware host (an SoC), containing the cryptographic module's hardware alongside the Host CPU, memories and peripherals, making it a sub-chip component.

DJI Core Crypto Engine is integrated into a SoC, where the host processor runs two separate operational environments: a Trusted Execution Environment (TEE) and a Rich Execution Environment (REE). The hardware isolation technology enforces data and control separation between the two environments containing the hardware and the firmware components. The module's hardware and firmware reflects the system architecture, with the components largely divided into two "cores" — the Secure Core (the TEE) and Public Core (the REE). The cores communicate via the Persistent State Interface registers to synchronize their state and pass parameters. The cryptographic boundary is defined by the following components, which are also depicted in Figure 1:

- Core Crypto TEE Firmware,
- Core Crypto REE Firmware,
- Core Crypto Secure Core,
- Core Crypto Public Core

All Core Crypto Engine hardware services are accessed through a firmware layer, providing a high-level interface to its functionality. Each environment's hardware has dedicated components for communication with the corresponding Host environment — bus connectors, register files for control and state passing, descriptor queues for task queuing and high-level control flow, interrupt and completion handling logic.

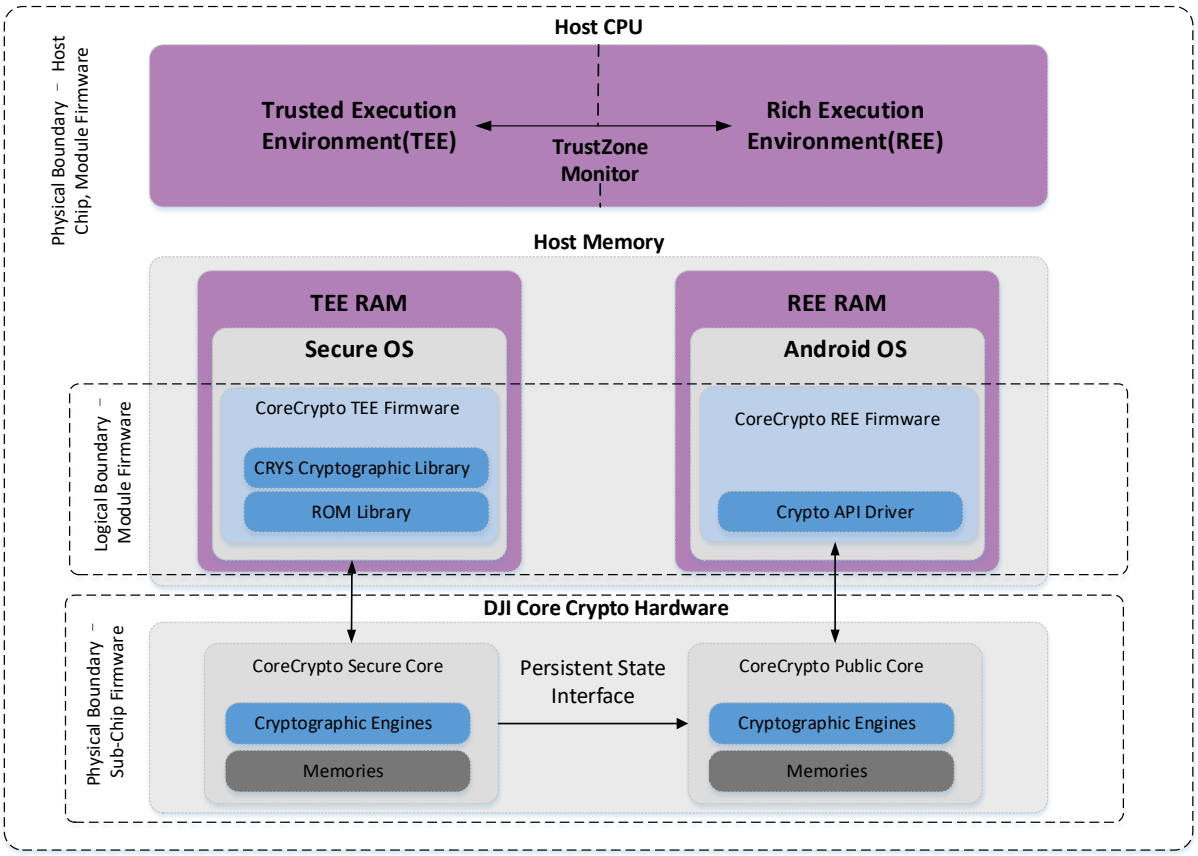

Figure 1 – Module Block Diagram

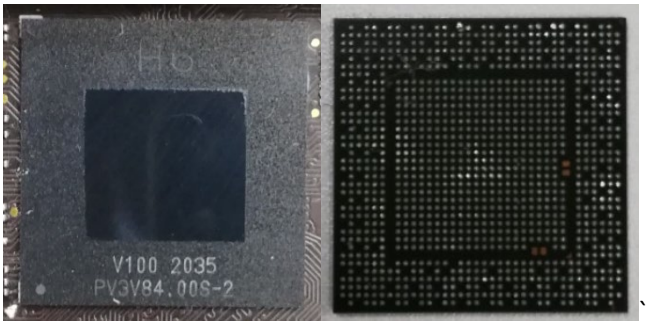

Figure 2 – Module Image

The module's ports and associated FIPS defined logical interface categories are listed in Table 3.

Table 3 – Ports and Interfaces

Port	Logical Interface Type
Power	Power
Clock	Control in

Port	Logical Interface Type
Reset	Control in
Scan	Control in
APB (AMBA Peripheral Bus)	Control in
Firmware API	Control in, Data in, Data out, Status out
Interrupt	Status out
AXI (Advanced extensible Interface)	Data in, Data out

1.2 Modes of Operation

The module supports both an Approved and a Non-Approved mode of operation. During module initialization, a Boolean parameter must be passed in selecting the mode of operation. The *FipsGetState* API can be used to check the mode at runtime and returns Approved, Non-Approved, or Error state; this API is implemented in both the TEE and the REE. While the *FipsGetState* may return the selected mode, it is still the operator's responsibility to ensure the procedural controls listed in this Security Policy are adhered to.

In the Approved mode, the operator is only to utilize approved security functions listed in Table 4. In the Non-Approved mode, the operator is only to utilize functions listed in Table 5. In order to switch between modes of operation, the operator is instructed to perform a power-on reset and initialize the module into the desired mode.

All CSPs (Critical Security Parameters) stored in the module's registers are cleared on power-on reset.

2 Cryptographic Functionality

The Module implements the FIPS Approved and Non-Approved but Allowed cryptographic functions listed in the tables below.

Table 4 – Approved Algorithms

Cert #	Algorithm	Mode	Description	Functions/Caveats
A2277, A2278	AES [197]	ECB [38A]	Key Sizes: 128, 192, 256	Encrypt, Decrypt
		CBC [38A]	Key Sizes: 128, 192, 256	Encrypt, Decrypt
		CTR [38A]	Key Sizes: 128, 192, 256	Encrypt, Decrypt
		OFB [38A]	Key Sizes: 128, 192, 256	Encrypt, Decrypt
		CMAC [38B]	Key Sizes: 128, 192, 256	Message Authentication (Generate Only)
VA	CKG [IG D.12]	[133] Section 5.1 Asymmetric signature key generation using unmodified DRBG output		Key Generation
		[133] Section 6.1 Direct symmetric key generation using unmodified DRBG output		
		[133] Section 6.2.2 Derivation of symmetric keys from a pre-shared key		
A2277	DRBG [90A]	CTR	Use_df AES-256	Deterministic Random Bit Generation. No assurance of the minimum strength of generated keys.
A2277	ECDSA [186]	-	P-224, P-256, P-384, P-521	KeyGen
			P-192, P-224, P-256, P-384, P-521	PKV
			P-224 SHA(224, 256, 384, 512) P-256 SHA(256, 384, 512) P-384 SHA(384, 512) P-521 SHA(512)	SigGen
			P-224 SHA(224, 256, 384, 512) P-256 SHA(256, 384, 512) P-384 SHA(384, 512) P-521 SHA(512)	SigVer
A2277	HMAC [198]	SHA-1	Key Sizes: 256 - 2048 λ = *32, 48, 64*, 80, 96, 128, 160	Message Authentication
		SHA-224	Key Sizes: 256 - 2048 λ = *32, 48, 64, 80, 96,* 128, 192, 256	
		SHA-256	Key Sizes: 256 - 2048 λ = *32, 48, 64, 80, 96,* 256, 320, 384, 448, 512	
		SHA-384	Key Sizes: 256 - 2048 λ = *32, 48, 64, 80, 96,* 256, 320, 384, 448, 512	
		SHA-512	Key Sizes: 256 - 2048	

Cert #	Algorithm	Mode	Description	Functions/Caveats
			λ = *32, 48, 64, 80, 96,* 256, 320, 384, 448, 512	
A2278	HMAC [198]	SHA-1	Key Sizes: 256 - 1536 λ = *32, 48, 64*, 80, 96, 128, 160	Message Authentication
		SHA-224	Key Sizes: 256 - 1536 λ = *32, 48, 64, 80, 96,* 128, 192, 256	
		SHA-256	Key Sizes: 256 - 1536 λ = *32, 48, 64, 80, 96,* 256, 320, 384, 448, 512	
A2277	KBKDF [108]	Counter	CMAC (AES-128, AES-256)	Key Based Key Derivation
A2277	RSA [186]	FIPS 186-4	2048, 3072-bit	Key Gen
		PKCS1_v1.5	n = 2048 SHA(224, 256, 384, 512) n = 3072 SHA(224, 256, 384, 512)	SigGen, SigVer
		PSS	n = 2048 SHA(224, 256, 384, 512) n = 3072 SHA(224, 256, 384, 512)	SigGen, SigVer
A2279	RSA [186]	PSS	n = 2048 SHA(256)	SigVer, Firmware Integrity Test
A2277	SHS [180]	SHA-1 SHA-224 SHA-256 SHA-384 SHA-512		Message Digest
A2278	SHS [180]	SHA-1 SHA-224 SHA-256		Message Digest

Table 5 – Non-Approved Cryptographic Functions (Non-Approved mode only)

Algorithm	Description
AES CBC-CS	All usage
AES CBC-MAC	All usage
AES CCM	All usage
AES GCM and GMAC	All usage
AES XCBC-MAC	All usage
DESECDSA	All usage with Non-Approved key sizes less than P-224 and non-NIST recommended curves, such as P224K1 and P256K1
ECIES	All usage
HMAC-MD5	All usage
IV/GEN RNG	All usage

Algorithm	Description
KAS	All usage of Diffie-Hellman and EC Diffie-Hellman
KDF1 and KDF2 (ISO/IEC-18033-2)	All usage
MD5	All usage
RSA	All usage with keys less than 2048 bits or greater than 3072
RSA	All usage of RSAES-PKCS1-v1.5 and RSAES-OAEP encryption and decryption
SHA-1	When used for signature generation
Triple-DES	All usage

2.1 Critical Security Parameters

All CSPs used by the Module are described in this section. All usage of these CSPs by the Module (including all CSP lifecycle states) is described in the services detailed in Section 3.

Table 6 – Critical Security Parameters (CSPs)

CSP	Description / Usage	Generation	Entry/Output	Storage	Zeroization
DRBG-EI	DRBG entropy input. 256 - 2048 with the nonce being 128 – 1024 bits. The entropy source must provide at least 112 bits of entropy to the DRBG.	Imported during manufacturing	Entry: N/A per IG 7.7; provided to the sub-chip module using the single chip internal paths. Output: N/A.	Plaintext in RAM	Power cycle
DRBG-State	CTR_DRBG internal state (V and Key – see 800-90A)	Per SP800-90A using the DRBG-EI	Entry: N/A. Plaintext from within Single Chip Internal Paths (IG 7.7) by User Output: N/A. Plaintext from within Single Chip Internal Paths (IG 7.7) by User during key generation	Plaintext in RAM	Power cycle
User Symmetric Keys	AES 128/192/256-bit keys with any supported mode used for the encryption or decryption of	DRBG	Entry: N/A. Plaintext from within Single Chip Internal Paths (IG 7.7) by User	Plaintext in RAM	Power cycle and after each use

CSP	Description / Usage	Generation	Entry/Output	Storage	Zeroization
	operator supplied data.		Output: N/A. Plaintext from within Single Chip Internal Paths (IG 7.7) by User during key generation		
User Asymmetric Key	RSA 2048/3072 or ECDSA P-224/-256/-384/-521 private keys used for digital signature generation.	DRBG	Entry: N/A. Plaintext from within Single Chip Internal Paths (IG 7.7) Output: N/A. Plaintext from within Single Chip Internal Paths (IG 7.7) by User during key generation	Plaintext in RAM	Power cycle and after each use
User Integrity Key	HMAC keys with any supported hash used for message integrity.	DRBG	Entry: N/A. Plaintext from within Single Chip Internal Paths (IG 7.7) Output: N/A. Plaintext from within Single Chip Internal Paths (IG 7.7) by User during key generation	Plaintext in RAM	Power cycle and after each use
Kdr	Device Root key. AES-256 bit key used to derive device-specific keys (Krpmb) using SP800-108 KBKDF.	N/A. Installed during manufacturing	N/A	Plaintext in OTP and HW Register	OTP Zeroization command (RMA)
Krpmb	RPMB shared HMAC-256 bit key used for providing message integrity over RPMB data frames.	N/A. Derived from Kdr using KBKDF	N/A	Plaintext in RAM	Power cycle

2.2 Public Keys

Table 7 – Public Keys

Key	Description / Usage	Generation	Entry/Output	Storage
Boot-PubK	Boot Public Keys (Qty. 2). RSA 2048-bit Public Key used to verify the firmware during Secure Boot.	N/A. Installed during manufacturing	N/A	Plaintext in Flash. SHA-256 hash of Boot-PubK is also stored in OTP
User Asymmetric Public Key	RSA 2048/3072-bit or ECDSA P-224/-256/-384/-521 public keys used for digital signature verification.	DRBG	Entry: N/A. Plaintext from within Single Chip Internal Paths (IG 7.7) Output: N/A. Plaintext from within Single Chip Internal Paths (IG 7.7) by User during key generation	Plaintext in RAM

3 Roles, Authentication and Services

3.1 Assumption of Roles

The module supports two distinct operator roles, User and Cryptographic Officer (CO). The module does not support authentication and as such, roles are implicitly assumed based on the services invoked.

Table 8 lists all operator roles supported by the module. The Module does not support a maintenance role or bypass capability. The Module does not support concurrent operators.

Table 8 – Roles Description

Role ID	Role Description
CO	Cryptographic Officer – Assumed by the manufacturer and is responsible for initializing the module for operational use, as well as life cycle management.
User	User – The operator exercising the cryptographic services provided by the firmware components of the TEE and REE, once the module is operational.

3.2 Services

All services implemented by the Module are listed in the table below. In addition, the relationship between access to CSPs and the different module services is also specified. The modes of access shown in the table are defined as:

- G = Generate: The service generates the CSP.
- O = Output: The service outputs the CSP using the single chip internal paths.
- E = Execute: The service uses the CSP in an algorithm.
- I = Input: The service inputs the CSP using the single chip internal paths.
- Z = Zeroize: The service zeroizes the CSP.

Table 9 – Authorized Services

Service	Description	CSP Access Rights	User	CO	REE	TEE
AES	Encryption, Decryption	User keys - I	**X**		**X**	**X**
SHA	Message Integrity	---	**X**		**X**	**X**
HMAC	Message Authentication	User keys - I	**X**		**X**	**X**
KBKDF	SP800-108 KBKDF	User keys - I, O KDR - E	**X**			**X**
RPMB Key Derivation	Replay Protected Memory Block (RPMB) Derive KRPMB from KDR using KBKDF.	KDR – E KRPMB - O	**X**			**X**

RPMB Page MAC	Calculate HMAC SHA-256 for RPMB Frame authentication	KRPMB - I	**X**			**X**
ECC Key Gen	ECC key generation	User key - O	**X**			**X**
Entropy Load	Load entropy	KRPMB - I	**X**			**X**
ECC PKV	Public key validation	User keys - I	**X**			**X**
ECDSA	Signature Generation/Verification	User keys - I	**X**			**X**
RSA Key Gen	Key pair generation	User keys - O	**X**			**X**
RSA	PSS Sign/Verify	User keys - I	**X**			**X**
DRBG	DRBG Instantiation, Reseed, Generation	DRBG context (V/Key) – I, O	**X**			**X**
DRBG Testing	Enable KAT mode	---	**X**			**X**
LCS Read	Reports the current Life Cycle State (LCS)	---	**X**			**X**
RMA (Zeroize)	Return Merchandising Authorization (RMA). Change LCS to RMA and zeroizes module.	Boot Key (Hash) - O RSA public key - I K_{CE} - Z K_{DR} - Z		**X**		**X**
TEE Library Initialization	Performs self-tests.	---	**X**			**X**
Secure Timer	Get Timestamp, compare Timestamp	---	**X**			**X**
Secure Boot—Read Configuration	Read Public Key Hash	Boot Key (Hash) - O	**X**			**X**
Secure Boot—Firmware Cert Verify	Firmware certificate chain verification using 2048-bit RSA-PSS; verifies boot certificate and firmware contents.	RSA public key - I	**X**			**X**
Identify SOC	Return SOC ID, unique device ID	K_{DR} - E Boot Key (Hash) - O	**X**			**X**
Show Status (FipsGetState)	Indicates the FIPS configuration: Approved, Non-Approved, or Error state	---	**X**		**X**	**X**
Self-Tests	Performs all power-on self-tests and is invoked by power cycling.	---	**X**			

4 Self-Tests

The module performs self-tests to ensure the proper operation of the module. Per FIPS 140-2 these are categorized as either power-up self-tests or conditional self-tests. Power up self–tests are available on demand by power cycling the module.

A platform-global variable is used to synchronize the state of the power-on self-tests between the TEE and the REE, preventing data output before self-tests complete.

A failure of any of these tests will cause the module to enter the Error state, which is indicated by the status output. On success, the module will enter the Approved or Non-Approved state as requested by the Host and will change its status output accordingly.

All algorithm Known Answer Tests (KATs) must be completed successfully prior to any other use of cryptography by the Module. If one of the KATs fails, the Module enters error state and outputs status indicating the specific self-test error code, otherwise it indicates successful completion by CC_FIPS_STATE_CRYPTO_APPROVED.

The module performs the following algorithm KATs on power-up (Note: Additional KATs have been implemented for non-Approved functions, but only those relevant to FIPS 140-2 conformance have been listed here):

- Firmware Integrity:
    - TEE and REE Firmware is verified using RSA-2048 PSS Signature Verification: This also acts as the KAT for this Bootloader implementation of RSA.
- AES Encrypt/Decrypt KATs (Certs. #A2277 and #A2278): ECB, CBC, OFB, and CTR using 128, 192, and 256-bit keys
- AES-CMAC KAT (Certs. #A2277 and #A2278): 128, 192, and 256-bit keys
- DRBG (CTR_DRBG) KAT (Cert. #A2277): Instantiate, Reseed, Generate
- ECDSA Sign/Verify KAT (Cert. #A2277): P-256 with SHA-256
- HMAC-SHA-256 KAT (Cert. #A2277)
- HMAC SHA-1 and SHA-256 KAT (Cert. #A2278)
- KBKDF KAT (Cert. #A2277): SP800-108
- RSA Sign/Verify KATs (Cert. #A2277 and #A2279): 2048-bit keys using PKCS#1v2.1 PSS
- SHA-1, -256, and -512 KATs (Certs. #A2277 and #A2278)

The module performs the following conditional self-tests as indicated.

- DRBG Continuous Random Number Generator Test
- DRBG: SP800-90A Health Tests.
- ECDSA Pairwise consistency test on ECDSA key pair generation
- RSA Pairwise consistency test on RSA key pair generation

5 Physical Security Policy

The Core Crypto Engine is a sub-chip module. The module is synthesized in a single host chip with standard passivation and a production grade enclosure that prevents access to the interior of the module and conforms to Level 1 requirements for physical security

6 Operational Environment

The Module has a non-modifiable operational environment under the FIPS 140-2 definitions. Firmware updates are not supported.

7 Mitigation of Other Attacks Policy

The Module has not been designed to mitigate attacks beyond the scope of FIPS 140-2 requirements.

8 Security Rules and Guidance

This section documents the security rules for the secure operation of the cryptographic module to implement the security requirements of FIPS 140-2.

1. The module provides two distinct operator roles: User and Cryptographic Officer.
2. The module allows the operator to initiate power-up self-tests by power cycling power or resetting the module.
3. The operator shall not invoke any cryptographic functions listed in Table 5, which specifies the non-Approved security functions only for use in the non-Approved mode.
4. Power up self-tests do not require any operator action.
5. Data output are inhibited during key generation, self-tests, zeroization, and error states.
6. Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the module.
7. There are no restrictions on which keys or CSPs are zeroized by the zeroization service.
8. The module does not support concurrent operators.
9. The module does not support a maintenance interface or role.
10. The module does not support manual key entry.
11. The module does not have any proprietary external input/output devices used for entry/output of data.
12. The module does not output intermediate key values.
13. The module does not provide bypass services or ports/interfaces.
14. In order to Zeroize the module through the RMA service:
    a. The operator must set the state value on the OTP dedicated field with the RMA state flag
    b. The operator puts the module through a power-on reset. During boot, the RMA state becomes available.
    c. The operator invokes the RMA service to enter the RMA state.
    d. The operator puts the module through a power-on reset again. Following this second power-on reset, the module is zeroized and is no longer operable.
    e. The operator can use the LCS Read service to verify that the device has indeed entered RMA mode.

9 References and Definitions

The following standards are referred to in this Security Policy.

Table 10 – References

Abbreviation	Full Specification Name
[FIPS140-2]	*Security Requirements for Cryptographic Modules*, May 25, 2001
[IG]	*Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program, February 14, 2022*
[108]	*NIST Special Publication 800-108, Recommendation for Key Derivation Using Pseudorandom Functions (Revised), October 2009*
[133]	*NIST Special Publication 800-133 Revision 2, Recommendation for Cryptographic Key Generation, June 2020*
[186]	*National Institute of Standards and Technology, Digital Signature Standard (DSS), Federal Information Processing Standards Publication 186-4, July, 2013.*
[197]	*National Institute of Standards and Technology, Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, November 26, 2001*
[198]	*National Institute of Standards and Technology, The Keyed-Hash Message Authentication Code (HMAC), Federal Information Processing Standards Publication 198-1, July, 2008*
[180]	*National Institute of Standards and Technology, Secure Hash Standard, Federal Information Processing Standards Publication 180-4, August, 2015*
[38A]	*National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation, Methods and Techniques, Special Publication 800-38A, December 2001*
[38A-A]	*National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-38A, Addendum, October 2010*
[38B]	*National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, Special Publication 800-38B, May 2005*
[38C]	*National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality, Special Publication 800-38C, May 2004*
[38E]	*National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices, Special Publication 800-38E, January 2010*
[90A]	*National Institute of Standards and Technology, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, Special Publication 800-90A, June 2015*