Allegro Software Development Corporation

Allegro Cryptographic Engine

Software Version: 1.1.8

FIPS 140-2 Non-Proprietary Security Policy

FIPS Security Level: 2
Document Version: 1.0

Prepared for:

Allegro Software Development Corporation
1740 Massachusetts Avenue
Boxborough, Massachusetts 01719
United States of America

Phone: +1 (978) 264-6600
Email: sales@allegrosoft.com
http://www.allegrosoft.com

Prepared by:

Corsec Security, Inc.
13135 Lee Jackson Memorial Highway, Suite 220
Fairfax, Virginia 22033
United States of America

Phone: +1 (703) 267-6050
Email: info@corsec.com
http://www.corsec.com

Table of Contents

Table of Figures

List of Tables

1 Introduction

1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the Allegro Cryptographic Engine from Allegro Software Development Corporation. This Security Policy describes how the Allegro Cryptographic Engine meets the security requirements of Federal Information Processing Standards (FIPS) Publication 140-2, which details the U.S. and Canadian Government requirements for cryptographic modules. More information about the FIPS 140-2 standard and validation program is available on the National Institute of Standards and Technology (NIST) and the Communications Security Establishment Canada (CSEC) Cryptographic Module Validation Program (CMVP) website at http://csrc.nist.gov/groups/STM/cmvp.

This document also describes how to run the module in a secure FIPS-Approved mode of operation. This policy was prepared as part of the Level 2 FIPS 140-2 validation of the module. The Allegro Cryptographic Engine is referred to in this document as ACE, the crypto-module, or the module.

1.2 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 on the module from the following sources:

- The Allegro website (http://www.allegrosoft.com) contains information on the full line of products from Allegro.
- The CMVP website (http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm) contains contact information for individuals to answer technical or sales-related questions for the module.

1.3 Document Organization

The Security Policy document is one document in a FIPS 140-2 Submission Package. In addition to this document, the Submission Package contains:

- Vendor Evidence document
- Finite State Model document
- Other supporting documentation as additional references

This Security Policy and the other validation submission documentation were produced by Corsec Security, Inc. under contract to Allegro. With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 Submission Package is proprietary to Allegro and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact Allegro.

2 Allegro Cryptographic Engine

2.1 Overview

Allegro Software Development Corporation is a leading provider of software toolkits that are used by manufacturers to enable their machines to be used on the internet. Allegro is also a leading provider of UPnP[1] and DLNA[2] technologies for networked consumer devices. Allegro offers a growing set of toolkits to provide cost-effective solutions to manufacturers. These toolkits are precision-engineered to meet the demands of embedded device developers working on cost-sensitive systems. The toolkits developed by Allegro are flexible enough to support the wide range of low-profile networking stacks, run-time environments, and low-cost microprocessors. Allegro's software is used in data communication products, enterprise products, consumer electronics, medical equipment, and more.

The toolkits developed by Allegro are comprised of Allegro's own cryptographic services. The collection of cryptographic services that include symmetric and asymmetric encryption and decryption, hashing, and digital signature is known as the Allegro Cryptographic Engine (ACE). Allegro Software Development Corporation develops their toolkits in modular and pluggable fashion, giving Original Equipment Manufacturers (OEMs) the ability to compile only exactly what is needed by their application in order to minimize the binary footprint in embedded applications. In order to reduce code duplication of many common cryptographic services, Allegro developed ACE as a single cryptographic provider with an open Application Programming Interface (API) available to both their toolkits as well as OEM applications.

A sample deployment of the Allegro Cryptographic Engine is shown in Figure 1 below. This deployment depicts ACE as it would be used by Allegro's licensed toolkits. ACE also provides a public API from which OEM developers may incorporate ACE into their own applications in order to provide FIPS-Approved cryptographic services.

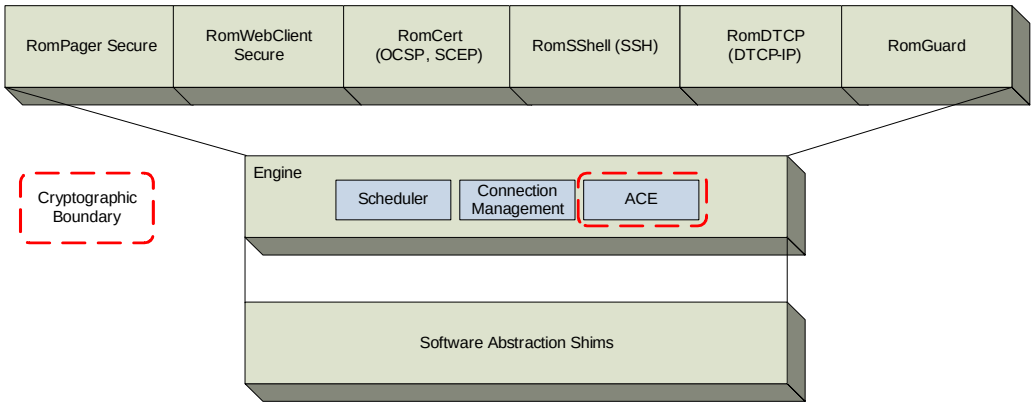

Figure 1 – Allegro Cryptographic Engine Deployment Scenario

[1] UPnP – Universal Plug-n-Play
[2] DLNA – Digital Living Network Alliance

The Allegro Cryptographic Engine is validated at the FIPS 140-2 Section levels shown in Table 1.

Table 1 – Security Level Per FIPS 140-2 Section

Section	Section Title	Level
1	Cryptographic Module Specification	2
2	Cryptographic Module Ports and Interfaces	2
3	Roles, Services, and Authentication	2
4	Finite State Model	2
5	Physical Security	N/A
6	Operational Environment	2
7	Cryptographic Key Management	2
8	EMI/EMC[3]	2
9	Self-tests	2
10	Design Assurance	2
11	Mitigation of Other Attacks	N/A

2.2 Module Specification

The Allegro Cryptographic Engine is a software cryptographic module with a multiple-chip standalone embodiment. The overall security level of the module is level 2. ACE is a shared cryptographic library providing symmetric and asymmetric encryption and decryption, hashing, message authentication, key generation, digital signature generation and verification, and other cryptographic functionality. The Allegro Cryptographic Engine consists of 2 files: ACE.dll and ACE.dll.dat.

The module was tested and found to be compliant on a Dell Optiplex 755 GPC[4] running an Intel Core 2 Duo E8400 64-bit processor executing the Windows 7 Ultimate Operating System (OS). The Windows 7 Ultimate OS is required to be running in its Common Criteria (CC) certified configuration in order to operate the cryptographic module in FIPS mode.

The GPC and OS were installed and configured to be CC EAL4+ compliant as specified in the Windows 7 Ultimate and Windows Server 2008 R2 Enterprise Edition Security Target and Validation report.

- Security Target: http://www.commoncriteriaportal.org/files/epfiles/st_vid10390-st.pdf
- Validation Report: http://www.commoncriteriaportal.org/files/epfiles/st_vid10390-vr.pdf

ACE is defined as a software cryptographic module and therefore has a logical boundary in addition to a physical boundary. The physical and logical boundaries are outlined in section 2.2.1 and 2.2.2 respectively.

2.2.1 Physical Cryptographic Boundary

As a software cryptographic module, the physical boundary of the cryptographic module is defined by the hard enclosure around the host system on which it runs. The module supports the physical interfaces of a Dell Optiplex 755 Desktop Computer. These interfaces include the integrated circuits of the system board, the CPU[5], network adapters, RAM[6], hard disk, device case, and power supply. Other devices may be attached to the GPC, such as a display monitor, keyboard, mouse, printer, or storage media. Figure 2 is a

[3] EMI/EMC – Electromagnetic Interference / Electromagnetic Compatibility
[4] GPC – General Purpose Computer
[5] CPU – Central Processing Unit
[6] RAM – Random Access Memory

block diagram representing the Dell Optiplex 755. The physical cryptographic boundary is defined by the red dotted line.

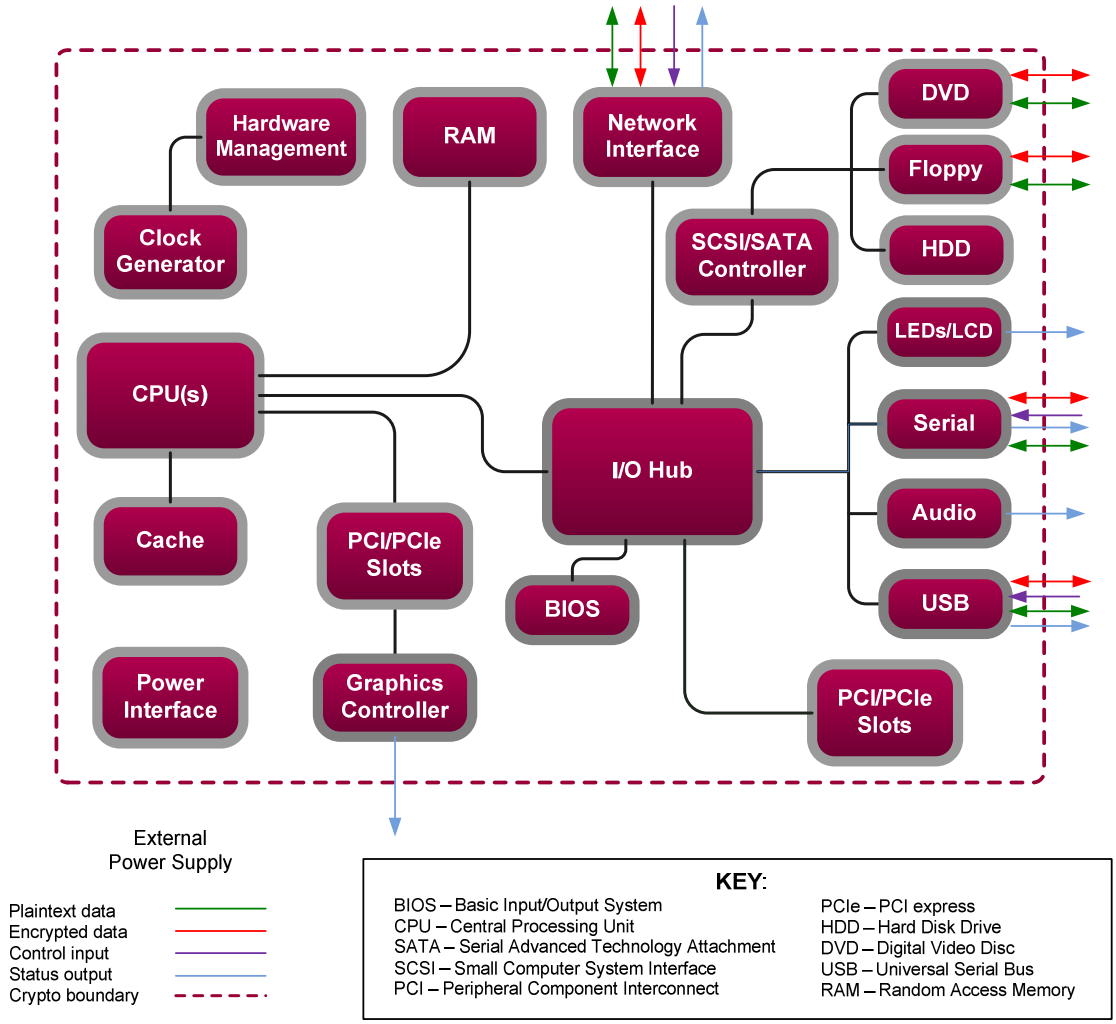

Figure 2 – Allegro Cryptographic Engine Physical Cryptographic Boundary

2.2.2 Logical Cryptographic Boundary

Figure 3 shows a logical block diagram of the module executing in memory and its interactions with surrounding software components, as well as the module's logical cryptographic boundary. The cryptographic module consists of 2 files: ACE.dll and ACE.dll.dat. These files are collectively shown as "Allegro Cryptographic Engine" in the diagram below. The module's services are designed to be called by Allegro's licensed toolkits or by OEM software applications, which define the module's logical interfaces.

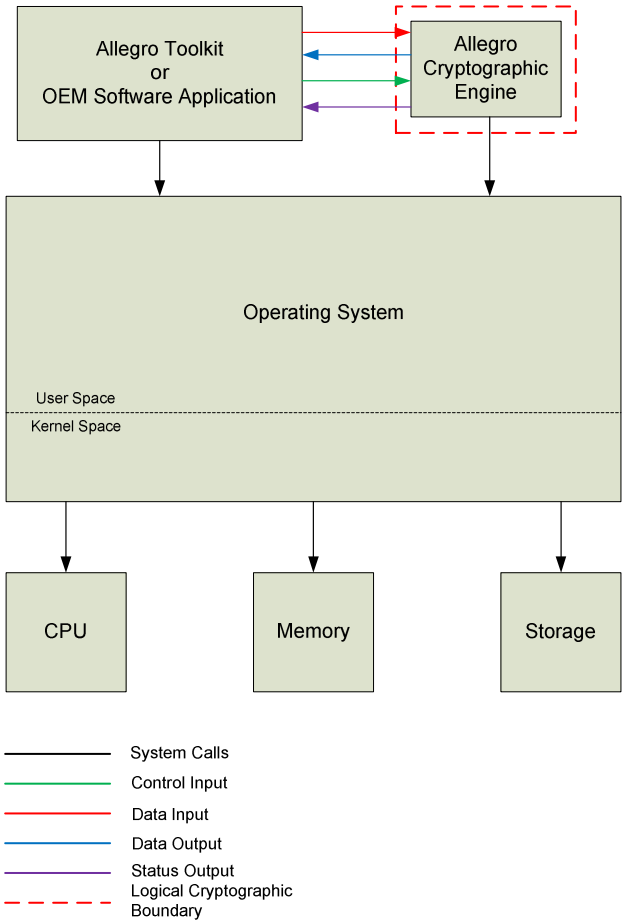

Figure 3 – Allegro Cryptographic Engine Logical Cryptographic Boundary

2.3 Module Interfaces

The module's physical ports can be categorized into the following logical interfaces defined by FIPS 140-2:

- Data input
- Data output
- Control input
- Status output

As a software module, the module doesn't have any physical characteristics. The module's physical and electrical characteristics, manual controls, and physical indicators are those of the host system. The mapping of the module's logical interfaces in the software to the physical interfaces of the Dell Optiplex 755 on which it is installed on is described in Table 2 below.

Table 2 – FIPS Interface Mapping

FIPS Interface	Physical Interface	Logical Interface
Data Input	USB[7] ports (8), network ports (1), serial ports (1), DVD[8] drive (1), 3.5" Floppy Drive (1)	The API calls that accept input data for processing through their arguments.
Data Output	USB ports (8), network ports (1), serial ports (1), DVD drive (1), 3.5" Floppy Drive (1)	The API calls that return by means of their return codes or arguments generated or processed data back to the caller.
Control Input	USB ports (8), network ports (1), serial ports (1), power switch (1)	The API calls that are used to initialize and control the operation of the module.
Status Output	VGA[9] port (1), network ports (1), serial ports (1), USB ports (8), audio ports (2), LED (7)	Return values for API calls.

2.4 Roles and Services

The Allegro Cryptographic Engine supports two roles (as required by FIPS 140-2) that an operator can assume: a Crypto Officer role and a User role. Access to the module is controlled through the authentication mechanism of the Operating system. Once authenticated, each role is assumed implicitly based on the service that is accessed. Services associated with each role are listed in Sections 2.4.1 and 2.4.2.

Please note that the keys and CSPs[10] listed in Table 3, Table 4, and Table 5 indicate the type of access required using the following notation:

- R – Read: The CSP is read.
- W – Write: The CSP is established, generated, modified, or zeroized.
- X – Execute: The CSP is used within an Approved or Allowed security function or authentication mechanism

2.4.1 Crypto Officer Role and Services

The Crypto Officer (CO) role is assumed to perform the initial installation of the module onto the Windows Operating System. The CO is responsible for generating keying material used for encryption, decryption, and signature generation and verification. The CO can also perform on-demand self-tests as well as zeroization of all keying material and other CSPs. Descriptions of the services available to the CO are provided in Table 3 below.

[7] USB – Universal Serial Bus
[8] DVD – Digital Video Disc
[9] VGA – Video Graphics Array
[10] CSP – Critical Security Parameter

Table 3 – Crypto Officer Services

Service	Description	CSP and Type of Access
AcRunSelfTest()	Run cryptographic self-tests on-demand	None
AcGenerateKey()	Generate symmetric keys	AES[11] Key – W AES GCM[12] Key – W AES GCM IV[13] – W AES CMAC[14] Key – W XTS[15,16,17]-AES Key – W Triple-DES[18] Key – W KEK[19] – W
AcGenerateKeyPair()	Generate asymmetric key pairs	RSA[20] Private Key – W RSA Public Key – W DSA[21] Private Key – W DSA Public Key – W ECDSA[22] Private Key – W ECDSA Public Key – W DH[23] Private Components – W DH Public Components – W ECDH[24] Private Components – W ECDH Public Components – W
AcBuildKeyPairFromParams()	Generate asymmetric key pairs given more specific key parameters	RSA Private Key – W RSA Public Key – W DSA Private Key – W DSA Public Key – W ECDSA Private Key – W ECDSA Public Key – W DH Private Components – W DH Public Components – W ECDH Private Components – W ECDH Public Components – W
AcDeriveKey()	Derive a key using pre-generated data	TLS[25] Session Key – W PBKDF2[26] DPK[27] – W
AcReleaseHandle()	Zeroize Keys	All Keys – W

[11] AES – Advanced Encryption System
[12] GCM – Galois Counter Mode
[13] IV – Initialization Vector
[14] CMAC – Cipher-based Message Authentication code
[15] XTS – XEX-Based Tweaked-Codebook Mode with Ciphertext Stealing
[16] XEX – XOR-Encrypt-XOR
[17] XOR – Exclusive Or
[18] DES – Data Encryption Standard
[19] KEK – Key Encrypting Key
[20] RSA – Rivest, Shamir, Adleman
[21] DSA – Digital Signature Algorithm
[22] ECDSA – Elliptic Curve Digital Signature Algorithm
[23] DH – Diffie-Hellman
[24] ECDH – Elliptic Curve Diffie-Hellman
[25] TLS – Transport Layer Security
[26] PBKDF2 – Password-Based Key Derivation Function 2
[27] DPK – Data Protection Key

2.4.2 User Role and Services

The User role performs general security services, including cryptographic operations and other Approved security functions such as random number generation, encryption and decryption, message authentication, and signature generation and verification. Descriptions of the services available to the User role are provided in Table 4 below.

Table 4 – User Services

Service	Description	CSP and Type of Access
AcGenerateRandomNumbers()	Generate random data	DRBG[28] Seed – W/R/X DRBG Entropy – R/X DRBG 'V' Value – W/R DRBG 'C' Value – W/R
AcDigest() AcDigestInit() AcDigestUpdate() AcDigestFinal()	Create message digest from input data	None
AcDigestClone()	Create message digest from previously digested data	None
AcDigestSize()	Calculate the size (in bytes) of a message digest	AES CMAC Key – R AES GCM Key – R
AcKeyedDigestInit() AcDigestUpdate() AcDigestFinal()	Create a keyed message digest of input data	HMAC Key – R/X AES Key – R/X AES GCM Key – R/X AES GCM IV – R/X AES CMAC Key – R/X
AcSign() AcSignInit() AcSignUpdate() AcSignFinal()	Sign a block of data	RSA Private Key – R/X DSA Private Key – R/X ECDSA Private Key – R/X
AcSignDigestBuffer()	Calculate a digital signature for a previously computed digest	RSA Private Key – R/X DSA Private Key – R/X ECDSA Private Key – R/X
AcVerify() AcVerifyInit() AcVerifyUpdate() AcVerifyFinal()	Verify a signed block of data	RSA Public Key – R/X DSA Public Key – R/X ECDSA Public Key – R/X
AcVerifyDigestBuffer()	Verify a digital signature for a previously computed digest	RSA Public Key – R/X DSA Public Key – R/X ECDSA Public Key – R/X
AcEncryptInit() AcEncryptUpdate() AcEncryptFinal()	Encrypt or decrypt a block of data[29]	AES Key – R/X AES GCM Key – R/X XTS-AES Key – R/X AES CMAC Key – R/X Triple-DES Key – R/X

[28] DRBG – Deterministic Random Bit Generator

[29] An encryption flag in the data structure is set to "TRUE" to indicate encryption and set to "FALSE" to indicate decryption

Service	Description	CSP and Type of Access
AcWrapKey()	Wrap/encrypt a symmetric key	KEK – R/X
AcUnwrapKey()	Unwrap/decrypt an encrypted symmetric key	KEK – R/X
AcKeySize()	Return the key size for a selected Key	All CSPs – R
AcKeyExchange()	Establish a shared secret using DH or ECDH	DH Private Components – R/X DH Public Components – R/X ECDH Private Components – R/X ECDH Public Components – R/X

2.4.3 Unauthorized Operator Services

The module provides two services that require an operator to authenticate to the OS, but do not require the operator to assume an authorized role. The following services can be called to either begin using the module for use in a FIPS-Approved mode of operation or to unload the module from memory and fix an error state. The services listed in Table 5 do not affect the overall security of the module and therefore do not require an authorized role to be accessed.

Table 5 – Unauthorized Operator Services

Service	Description	CSP and Type of Access
AllegroTaskInit()	Initialize the module for use in FIPS mode	None
AllegroTaskDeInit()	Disable Crypto Services; Zeroization	All CSPs – W

2.4.4 Authentication

The Allegro Cryptographic Engine itself does not support an approved authentication mechanism. ACE is being evaluated at Level 2 for Operational Environment requirements. As such, the cryptographic module may rely on the authentication mechanism provided by the Operating System. The Windows 7 Ultimate Operating System meets the functional requirements specified in the U.S. Government Approved Protection Profile for General-Purpose Operating Systems.

2.4.4.1 Role-Based Authentication

The module uses role-based authentication. Windows 7 Ultimate Operating System provides an identity-based authentication mechanism. This satisfies the role-based authentication requirements of the module per section 3.3 of "Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program" only when the operating system is running according to the Common Criteria configuration guide. When the operator authenticates to the OS with a correct password, they will implicitly assume the role of CO or User based on the service that is accessed. Each service offered by the cryptographic module has been assigned a role. To perform a service, the operator calls a cryptographic module API and by doing so implicitly selects the role. Sections 2.4.1 and 2.4.2 outline the responsibilities of the CO and User and list the services associated with each role.

2.4.4.2 Authentication Mechanism Strength

As specified in Section 3.1.1.1, the minimum password length to access the Windows 7 OS is sixteen (16) characters. The password may contain any combination of upper- and lower-case letters, numbers, and printable symbols; allowing for 94 possible characters. Therefore, there is at minimum, $94^{16} = 3.7x10^{31}$

possible password combinations. This means there is a 1 in $3.7x10^{31}$ chance that one random access attempt will succeed. This surpasses the 1 in 1,000,000 requirement of the FIPS 140-2 standard.

In order to surpass the 1 in 100,000 likelihood that random, successive authentication attempts will succeed or a false acceptance will occur in a one minute period, an attacker would be required to make $94^{16}/100,000 = 3.7x10^{26}$ attempts in one minute. By default, Windows 7 OS will only allow 10 failed authentication attempts per minute before locking the account. Therefore, this requirement is satisfied.

2.5 Physical Security

The Allegro Cryptographic Engine is a software module, which FIPS defines as a multiple-chip standalone cryptographic module. As such, it does not include physical security mechanisms. Thus, the FIPS 140-2 requirements for physical security are not applicable.

2.6 Operational Environment

The Allegro Cryptographic Engine (Software Version: 1.1.8) was tested and found to be compliant with FIPS 140-2 requirements on a Dell Optiplex 755 GPC running an Intel Core 2 Duo E8400 64-bit processor executing the Windows 7 Ultimate Operating System. The Windows 7 Ultimate Operating System meets the functional requirements specified in the U.S. Government Approved Protection Profile for General-Purpose Operating Systems in a Networked Environment. The Windows 7 Ultimate Operating System was Common Criteria certified at EAL[30]4+ (CCTL[31] Validation Report Number: CCEVS-VR-VID10390-2010). The Windows 7 Ultimate OS must be set up in its CC-evaluated configuration to run the Allegro Cryptographic Engine in its FIPS-Approved mode of operation. See Section 3.1.1 for more information regarding this set-up.

2.7 Cryptographic Key Management

The Allegro Cryptographic Engine implements the FIPS-Approved algorithms listed in Table 6 below.

Table 6 – FIPS-Approved Algorithm Implementations

Algorithm	Certificate Number
AES ECB[32], CBC[33], CTR[34], CFB[35], CFB8, CFB128, OFB[36], CCM[37] encryption/decryption and wrap/unwrap with 128-,192-, and 256-bit keys	2671
AES GCM encryption/decryption and message authentication with 128-, 192-, and 256-bit keys	2671
XTS-AES encryption/decryption with XTS_128- and XTS_256-bit keys[38]	2671
Triple-DES ECB, CBC, CFB, CFB8, CFB64, OFB encryption/decryption; 3-key	1602
RSA (FIPS186-3) (ANSI[39] X9.31) key pair generation with 2048- and 3072-bit keys; Signature Generation and Verification	1374

[30] EAL – Evaluation Assurance Level
[31] CCTL – Common Criteria Testing Lab
[32] ECB – Electronic Code Book
[33] CBC – Cipher Block Chaining
[34] CTR - Counter
[35] CFB – Cipher Feedback
[36] OFB – Output Feedback
[37] CCM – Counter with CBC-MAC
[38] The vendor affirms that the length of data in all instances of AES-XTS does not exceed 2^{20} blocks
[39] ANSI – American National Standards Institute

Algorithm	Certificate Number
RSA (FIPS186-3) (PKCS[40] #1 v1.5) signature generation and verification; Encapsulation/Unencapsulation with 2048- and 3072- bit keys	1374
RSA (FIPS186-3) (PSS[41]) signature generation and verification	1374
DSA (FIPS186-3) key pair generation with 2048- and 3072-bit keys; signature generation and verification	810
DSA (FIPS186-3) PQG Generation and Verification	810
ECDSA (FIPS186-3) key pair generation with NIST curves: P-192, P-224, P-256, P-384, and P-512; signature generation and verification	465
SHA[42]-1, SHA-224, SHA-256, SHA-384, and SHA-512	2243
HMAC[43]with SHA-1, SHA-224, SHA-256. SHA-384, and SHA-512	1661
CMAC[44] generation and verification with AES 128-, 192-, and 256-bit keys	2671
Diffie-Hellman FFC[45] Key Agreement Scheme (SP800-56A)	148
EC[46] Diffie-Hellman ECC[47] Key Agreement Scheme (SP800-56A) with NIST curves: P-192, P-224, P-256, P-384, and P-521	148
SP[48] 800-90A Hash_DRBG[49]	430

Caveats:

- Additional information concerning SHA-1 and RSA key transport and specific guidance on transitions to the use of stronger cryptographic keys and more robust algorithms is contained in NIST Special Publication 800-131A.
- The module implements MD5[50] for use with SSL3.1[51]/TLS1.0 communications, which is allowed in the FIPS-Approved mode of operation.
- The module generates cryptographic keys whose strengths are modified by available entropy.
- The module generates keys per Scenario 1 of IG 7.l8

The module employs the following key establishment methodologies, which are allowed for use in a FIPS-Approved mode of operation:

- RSA (2048- or 3072-bit keys; key wrapping; key establishment methodology provides 112 to 128 bits of encryption strength)
- AES (Cert # 2671, key wrapping; key establishment methodology provides between 128 and 256 bits of encryption strength)
- Diffie-Hellman (CVL Cert. #148, key agreement; key establishment methodology provides between 80 and 128 bits of encryption strength)

[40] PKCS – Public-Key Cryptography Standards
[41] PSS – Probabilistic Signature Scheme
[42] SHA – Secure Hash Algorithm
[43] HMAC – (keyed-) Hash Message Authentication Code
[44] CMAC – Cipher-based Message Authentication Code
[45] FFC – Finite Field Cryptography
[46] EC – Elliptic Curve
[47] ECC – Elliptic Curve Cryptography
[48] SP – Special Publication
[49] DRBG – Deterministic Random Bit Generator
[50] MD5 – Message Digest v5
[51] SSL3.1 – Secure Socket Layer v3.1

- EC Diffie-Hellman (CVL Cert. #148, key agreement; key establishment methodology provides between 80 and 256 bits of encryption strength)

Allegro Software Development Corporation affirms compliance with SP 800-132 for the full implementation of PBKDF2. The Allegro Cryptographic Engine implements option 1(a) from section 5.4 of the Special Publication. Please refer to section 3.2.1.1 for Cryptographic Officer guidance specific to this function.

The module supports the CSPs listed below in Table 7.

Table 7 – List of Cryptographic Keys, Cryptographic Key Components, and CSPs

CSP	Key Type	Generation[52] / Input	Output	Storage	Zeroization	Use
AES Key	AES 128-, 192-, or 256-bit key	Internally Generated via approved DRBG; or Input via API in plaintext	Output encrypted via KEK	Keys are not persistently stored by the module[53]	Unload module; API call; Remove Power	Encrypt and decrypt blocks of data
AES GCM Key	AES 128-, 192-, or 256-bit key	Internally Generated via approved DRBG; or Input via API in plaintext	Output encrypted via KEK	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Encrypt and decrypt blocks of data; Keyed Message Authentication Code
AES GCM IV[54]	>= 96 bits of random data	Internally Generated via approved DRBG	Never	Keys are not persistently stored by the module	Unload module; API call; Remove Power	IV input to AES GCM function
XTS-AES Key	AES XTS_128- or AES XTS_256-bit key	Internally Generated via approved DRBG	Never	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Storage encryption or decryption
AES CMAC Key	AES 128-, 192-, or 256-bit key	Internally Generated via approved DRBG; or Input via API in plaintext	Output encrypted via KEK	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Keyed Message Authentication Code

[52] Post processing is not performed on the output of the DRBG during key generation
[53] Keys are temporarily stored in the volatile memory of the host platform
[54] External generation of the IV is not permitted per IG A.5.

CSP	Key Type	Generation[52] / Input	Output	Storage	Zeroization	Use
Triple-DES Key	Triple-DES 168-bit key	Internally Generated via approved DRBG; or Input via API in plaintext	Output encrypted via KEK	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Encrypt and decrypt blocks of data
HMAC Key	80- to 512-bit HMAC Key	Internally Generated via approved DRBG; or Input via API in plaintext	Output encrypted via KEK	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Keyed Message Authentication Code
Key Encryption Key (KEK)	AES 128-, 192-, 256-bit key or RSA 2048-, 3072-bit key	Internally Generated via PBKDF2; or Input via API in plaintext	Output in plaintext via GPC INT[55] Path	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Key Wrapping / Key Unwrapping
PBKDF2 DPK	112-bits of random data	Internally Generated	Output encrypted via KEK	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Protection of stored data
RSA Private Key	2048- or 3072-bit RSA2 Private Key	Internally Generated via approved DRBG; or Input via API in plaintext	Output encrypted via KEK	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Signature Generation; Decryption
RSA Public Key	2048- or 3072-bit RSA2 Public Key	Internally Generated via approved DRBG; or Input via API in plaintext	Output in plaintext via GPC INT Path	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Signature Verification; Encryption

[55] INT - Internal

CSP	Key Type	Generation[52] / Input	Output	Storage	Zeroization	Use
DSA Private Key	224- or 256-bit DSA2 Private Key	Internally Generated via approved DRBG; or Input via API in plaintext	Output encrypted via KEK	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Signature Generation;
DSA Public Key	2048- or 3072-bit DSA2 Public Key	Internally Generated via approved DRBG; or Input via API in plaintext	Output in plaintext via GPC INT Path	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Signature Verification;
ECDSA Private Key	NIST Recommended Curve Sizes: P-192, P-224, P-256, P-384, or P-521	Internally Generated via approved DRBG; or Input via API in plaintext	Output encrypted via KEK	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Signature Generation;
ECDSA Public Key	NIST Recommended Curve Sizes: P-192, P-224, P-256, P-384, or P-521	Internally Generated via approved DRBG; or Input via API in plaintext	Output in plaintext via GPC INT Path	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Signature Verification;
DH Private Components	160-, 224- or 256- bit Diffie Hellman Private Key	Internally Generated via approved DRBG; or Input via API in plaintext	Output encrypted via KEK	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Establish Secure SSH[56] Session

[56] SSH – Secure Shell

CSP	Key Type	Generation[52] / Input	Output	Storage	Zeroization	Use
DH Public Components	1024-, 2048-, or 3072-bit Diffie-Hellman Public Key	Internally Generated via approved DRBG; or Input via API in plaintext	Output in plaintext via GPC INT Path	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Establish Secure SSH Session
ECDH Private Components	NIST Recommended Curve Sizes: P-192, P-224, P-256, P-384, or P-521	Internally Generated via approved DRBG; or Input via API in plaintext	Output encrypted via KEK	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Establish Secure SSH Session
ECDH Public Components	NIST Recommended Curve Sizes: P-192, P-224, P-256, P-384, or P-521	Internally Generated via approved DRBG; or Input via API in plaintext	Output in plaintext via GPC INT Path	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Establish Secure SSH Session
TLS Session Key	Shared TLS symmetric key	Internally Generated via approved DRBG	Output encrypted via KEK	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Encrypt/Decrypt communications over TLS
TLS Integrity Key	HMAC SHA-1 key	Internally Generated via approved DRBG; or Input via API in plaintext	Output in plaintext via GPC INT Path	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Protects the integrity of data sent over TLS
DRBG Seed	440-or 888-bits of random material[57]	Internally Generated	Never	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Seeding material for Hash_DRBG

[57] Depending on hash algorithm used (See SP 800-90A, Table 2)

CSP	Key Type	Generation[52] / Input	Output	Storage	Zeroization	Use
DRBG Entropy	256-bit random material	Externally Generated[58]	Never	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Entropy material for Hash_DRBG
DRBG 'C' Value	Internal Hash_DRBG state value	Internally Generated	Never	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Used for Hash_DRBG
DRBG 'V' Value	Internal Hash_DRBG state value	Internally Generated	Never	Keys are not persistently stored by the module	Unload module; API call; Remove Power	Used for Hash_DRBG

[58] The module employs the random number generator of the Windows 7 Ultimate Operating System

2.8 EMC/EMI

The Allegro Cryptographic Engine is a software module. Therefore, the only electromagnetic interference produced is that of the Dell Optiplex 755 Desktop Computer; on which the module resides and executes. FIPS 140-2 requires that the host systems on which FIPS 140-2 testing is performed meet the Federal Communications Commission (FCC) EMI and EMC requirements for business use as defined in Subpart B, Class A of FCC 47 Code of Federal Regulations Part 15. However, all systems sold in the United States must meet these applicable FCC requirements.

Confirmation of this testing is provided at:
http://support.dell.com/support/edocs/systems/op755/en/UG/HTML/fcc.htm

2.9 Self-Tests

The Allegro Cryptographic Engine performs power-up self-tests automatically each time the GPC is powered on and the module is loaded into memory. Conditional self-tests are performed each time the module needs to generate a new random number or a new asymmetric key pair, or when establishing a new Diffie-Hellman Key Agreement. The module's random bit generator will perform critical function tests as needed to assure its security. While the module is performing these self-tests, all data output interfaces are inhibited.

Should any self-test fail, the module's data output interfaces will be inhibited. Only control input and status output commands will be allowed to execute. To correct a power-up self-test, on-demand self-test, or conditional self-test error, the module must be reloaded into memory by either restarting the module or by calling the AllegroTaskInit() service after the module has been de-initialized.

2.9.1 Power-Up Self-Tests

The Allegro Cryptographic Engine performs the following self-tests at power-up:

- Software integrity check using HMAC SHA-256 Message Authentication Code
- Known Answer Tests (KATs)
  - AES KAT[59] [60],
  - AES CMAC KAT
  - Triple-DES KAT
  - RSA KAT[61]
  - DSA Sign/Verify Pairwise Consistency Test
  - ECDSA Sign/Verify Pairwise Consistency Test
  - SHA-1 KAT
  - HMAC with SHA-1 KAT
  - SHA-224, SHA-256, SHA-384, SHA-512 KAT
  - HMAC with SHA-224, SHA-256, SHA-384, SHA-512 KAT
  - Diffie-Hellman Primitive 'Z' Computation KAT
  - EC Diffie-Hellman Primitive 'Z' Computation KAT
  - SP 800-90A Hash_DRBG KAT

All Known Answer Tests may be called on-demand by calling the AcRunSelfTest() service.

2.9.2 Conditional Self-Tests

The Allegro Cryptographic Engine performs the following conditional self-tests when needed:

- Conditional Self Tests (CSTs)

[59] Includes CBC, CFB1, CFB8, CFB128, ECB, OFB, CTR, GCM, and CCM modes in 128-, 192-, and 256-bit sizes
[60] Includes XTS mode in 128 and 256-bit sizes
[61] Sign/Verify KAT using 2048 bit key, SHA-256 hash

- RSA Pairwise Consistency Test
- DSA Pairwise Consistency Test
- ECDSA Pairwise Consistency Test
- Diffie-Hellman Public Key Assurance Test
- EC Diffie-Hellman Public Key Assurance Test
- SP 800-90A Hash_DRBG Continuous Random Number Generation Test
- Continuous Random Number Generator Test for Entropy Source

2.9.3 Critical Functions Self-Tests

Critical function tests are performed conditionally by the module any time a random number is generated using the SP 800-90A Hash_DRBG. The SP 800-90A Hash_DRBG contains two critical functions; DRBG Instantiate and DRBG Reseed. The instantiation test is tested during power-up and any time that a new DRBG instance is created. The reseed test is performed conditionally when the reseed counter has reached its pre-determined maximum value and the DRBG needs to be reseeded. If any of these critical function tests fail, the module will transition to a soft error state. Follow the guidance in Section 2.9 to correct the error state.

The Allegro Cryptographic Engine performs the following critical function tests:

- SP 800-90A DRBG Instantiation Critical Function Test
- SP 800-90A DRBG Reseed Critical Function Test

2.10 Design Assurance

Allegro uses Perforce Server as their configuration management tool to track the progress and design of their source code and product manuals. To ensure secure delivery of the Allegro Cryptographic Engine, Allegro places the module onto a DVD and ships the DVD via FedEx. Tracking numbers are used to track the progress of the shipment to the customer. FedEx requires the recipient of the product to sign for the package to ensure the product arrives securely to the intended recipient.

2.11 Mitigation of Other Attacks

This section is not applicable. The modules do not claim to mitigate any attacks beyond the FIPS 140-2 Level 2 requirements for this validation.

3 Secure Operation

The Allegro Cryptographic Engine meets Level 2 requirements for FIPS 140-2. The sections below describe how to place and keep the module in FIPS-Approved mode of operation.

3.1 Initial Setup

Initial setup for the Allegro Cryptographic Engine consists of installing Windows 7 Ultimate in its CC evaluated configuration, creating a new user account on the Operating System and providing that user account with a password. After creating a new user account and password, the CO shall follow the CO Guidance in Section 3.2.1 to use ACE in its FIPS-Approved mode of operation.

3.1.1 Operating System Configuration

The Windows 7 Ultimate Operating System will provide the operational environment as well as the authentication mechanism required for the module to meet Level 2 FIPS security specifications. Windows 7 Ultimate shall be installed in its CC evaluated configuration. The CO shall refer to the guidance supplementation mentioned in the "Microsoft Windows Common Criteria Evaluation for Microsoft Windows 7 and Microsoft Windows Server 2008 R2 Security Target" in order to properly configure the Operating System.

To run ACE in its FIPS-Approved mode of operation, a new user account shall be created on the OS. After logging into the Admin account, the CO will create a new user account following the guidelines of the OS user manual. When creating a new user, the CO shall require a password to log into the account. The CO shall refer to all administrative and guidance documents in order to create a new user account on the Operating System.

3.1.1.1 Password Requirements

The password shall be, at minimum, 16 characters in length and shall consist of a combination of upper- and lower-case letters, numbers, and printable symbols. The CO shall not include a password hint/reminder. After creating the new user account, the CO shall log into the new user account to install the Allegro Cryptographic Engine.

3.2 Secure Management

The Cryptographic Officer is in charge of the secure management and handling of the ACE cryptographic module. The Allegro Cryptographic Engine is shipped on a DVD and delivered via FedEx. A tracking number is provided to the CO in order to track the progress of the shipment and ensure secure delivery of the module. The CO shall sign for the DVD upon arrival and shall maintain control of the DVD throughout its lifetime. Following the secure delivery of the module, the CO shall follow the steps outlined in Section 3.1 for proper configuration of the Operating System prior to installing the module onto the OS.

3.2.1 CO Guidance

As explained the sections above, the CO is in charge of setting up the Windows 7 Ultimate Operating System and receiving the DVD containing the cryptographic module, installation guides, user guides, and other supporting documentation. The CO shall follow the installation procedures detailed in the included installation guides to properly install the Allegro Cryptographic Engine onto the operating system. The module is shipped in its FIPS-Approved mode of operation. No further configuration is needed by the CO in order to operate the module in its FIPS-Approved mode of operation. During normal operation, the User may check the status of the module by attempting to run a service. If the service executes, the module is operating in FIPS mode.

3.2.1.1 Guidance for Password-Based Key Derivation Function

Passwords passed to the PBKDF2 implemented by the Allegro Cryptographic Engine shall be, at minimum, 6 characters and shall consist of upper- and lower-case letters and numbers. The probability of guessing this password at random is 1 in 62^6 or 1 in $5.68x10^{10}$. The Data Protection Key (DPK) derived from the PBKDF2 shall be used for storage purposes only.

3.2.2 User Guidance

The user shall adhere to the guidelines of this Security Policy. The User does not have any ability to install or configure the module. Operators in the User role are able to use the services available to the User role listed in Table 4. The use of MD5 shall be limited to SSL3.1/TLS1.0 communications. The restricted use of this function is enforced by the module. The user is responsible for reporting to the Cryptographic Officer if any irregular activity is noticed. During operation, the User may check the status of the module by attempting to run a service. If the service executes, the module is operating in FIPS mode.

4 Acronyms

Table 8 lists the acronyms used throughout this Security Policy.

Table 8 – Acronyms

Acronym	Definition
ACE	Allegro Cryptographic Engine
AES	Advanced Encryption System
ANSI	American National Standards Institute
API	Application Programming Interface
CBC	Cipher Block Chaining
CCM	Counter with CBC-MAC
CCTL	Common Criteria Testing Lab
CFB	Cipher Feedback
CMAC	Cipher-based Message Authentication Code
CMVP	Cryptographic Module Validation Program
CO	Cryptographic Officer
CPU	Central Processing Unit
CSEC	Communications Security Establishment Canada
CSP	Critical Security Parameter
CST	Conditional Self-Test
CTR	Counter
DES	Data Encryption Standard
DH	Diffie-Hellman
DLNA	Digital Living Network Alliance
DPK	Data Protection Key
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
DVD	Digital Video Disc
EAL	Evaluation Assurance Level
EC	Elliptic Curve
ECB	Electronic Code Book
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EMC	Electromagnetic Compatibility

Acronym	Definition
EMI	Electromagnetic Interference
FCC	Federal Communications Commission
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standard
GCM	Galois/Counter Mode
GPC	General Purpose Computer
HMAC	(keyed-) Hash Message Authentication Code
INT	Internal
KAT	Known Answer Test
KEK	Key Encrypting Key
MAC	Message Authentication Code
MD5	Message Digest v5
NIST	National Institute of Standards and Technology
OEM	Original Equipment Manufacturer
OFB	Output Feedback
OS	Operating System
PBKDF2	Password-Based Key Derivation Function 2
PKCS	Public Key Cryptography Standard
PSS	Probabilistic Signature Scheme
RAM	Random Access Memory
RSA	Rivest Shamir and Adleman
SATA	Serial Advanced Technology Attachment
SCSI	Small Computer System Interface
SHA	Secure Hash Algorithm
SP	Special Publication
SSH	Secure Shell
SSL	Secure Socket Layer
TLS	Transport Layer Security
Triple-DES	Triple Data Encryption Standard
UPnP	Universal Plug-n-Play
USB	Universal Serial Bus
XEX	XOR-Encrypt-XOR
XOR	Exclusive Or
XTS	XEX-Based Tweaked-Codebook Mode with Ciphertext Stealing

Prepared by:
Corsec Security, Inc.

13135 Lee Jackson Memorial Highway, Suite 220
Fairfax, Virginia 22033
United States of America

Phone: +1 (703) 267-6050
Email: info@corsec.com
http://www.corsec.com