Cryptographic Module for CipherCloud Gateway

Version 1.0

FIPS 140-2 Level 1 Security Policy

Version Number: 1.8

Date: August 14, 2014

Table of Contents

1. Module Overview

The Cryptographic Module enables all cryptographic operations performed by the CipherCloud Gateway. The CipherCloud Gateway is a software solution that organizations deploy within their network boundaries or delegate operation to a trusted third party. CipherCloud interfaces with clients (e.g., web browsers, mobile applications, APIs, etc.), and leverages format and operations preserving encryption technology to secure sensitive information in real time, before it's sent to cloud applications (e.g. web servers, API services, databases, etc.), without impacting usability or performance.

The cryptographic module is a software module that is executing in a modifiable operational environment by a general purpose computer.

This software module comprises the following components:

- CCFIPSModule.jar
- Jce.jar
- rt.jar
- sunjce_provider.jar
- local_policy.jar
- US_export_policy.jar

FIPS 140-2 conformance testing was performed at Security Level 1. The following configuration was tested by the lab.

Table 1: Configuration tested by the lab.

Software Component	Operating System	Jave Run-time Environment
CCFIPSModule.jar Jce.jar rt.jar sunjce_provider.jar local_policy.jar US_export_policy.jar	CentOS 6.3	Java JRE 1.6.0

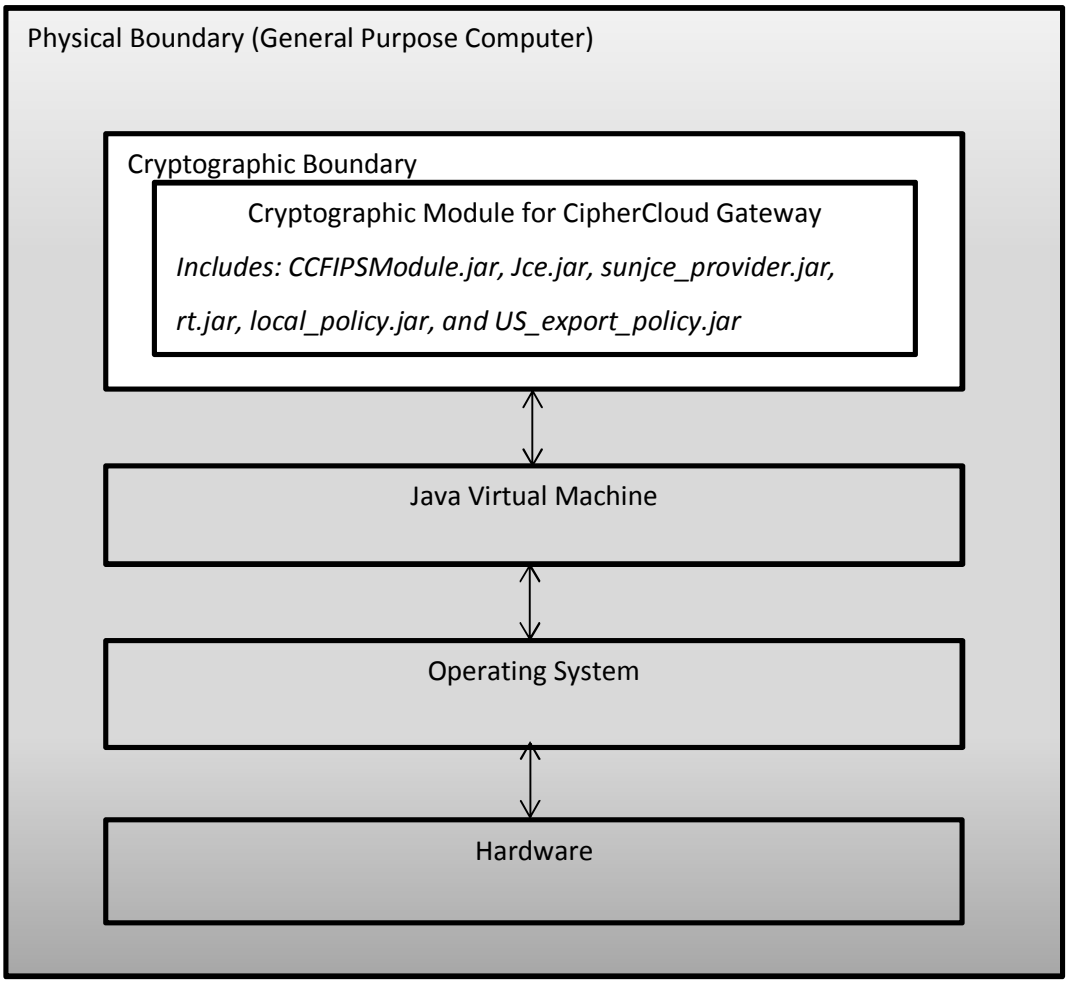
The Cryptographic Module for CipherCloud Gateway meets FIPS 140-2 Level 1 requirements.

Table 2: Module Security Level Statement.

FIPS Security Area	Security Level
Cryptographic Module Specification	1
Module Ports and Interfaces	1
Roles, Services and Authentication	1
Finite State Model	1
Physical Security	N/A

FIPS Security Area	Security Level
Operational Environment	1
Cryptographic Key Management	1
EMI/EMC	1
Self-tests	1
Design Assurance	1
Mitigation of Other Attacks	N/A

Figure 1: Cryptographic Module for CipherCloud Gateway Block Diagram.



2. Modes of Operation

In the FIPS approved mode of operation the operator must only use FIPS-approved security functions listed in the Section 2.1. When the class CCFIPSModule is instantiated, the module gets loaded and the power-up self-tests are performed.

In the non-FIPS mode of operation the module performs non-approved functions listed in the Section "2.2 All Other Algorithms" of this security policy. These functions shall not be used in FIPS approved mode of operation.

2.1 Approved Cryptographic Functions

The following approved cryptographic algorithms are used in FIPS approved mode of operation.

Table 3: Approved Cryptographic Functions.

Algorithm	CAVP Certificate
AES (ECB, CBC and CFB)	2339
SHS (SHA1, SHA256 and SHA512)	2017
HMAC (HMAC-SHA1, HMAC-SHA256)	1449
PBKDF2 (SP 800-132) (HMAC-SHA1, HMAC-SHA256)	vendor affirmed
SP 800-90A CTR DRBG	303

2.2 All other algorithms

In the FIPS approved mode of operation the operator must not use the functions listed in the Table 4. These functions are available in the User role.

Table 4: Non-Approved Cryptographic Functions and Services.

Algorithm	Non-Approved Service
AES (PCBC, CTR, CTS, OFB, OFB8, and OFB128 modes; non-compliant)	Encryption/Decryption
ARC4	Encryption/Decryption
Blowfish	Encryption/Decryption
DES	Encryption/Decryption
Diffie-Hellman (non-compliant)	Key Agreement
DSA (non-compliant)	Digital Signature Generation/Verification
RC2	Encryption/Decryption
RSA	Encryption/Decryption
Triple-DES (non-compliant)	Encryption/Decryption
PBEWithMD5AndDES	Encryption/Decryption
PBEWithMD5AndTripleDES	Encryption/Decryption
PBEWithSHA1AndDESede	Encryption/Decryption
PBEWithSHA1AndRC2_40	Encryption/Decryption
MD2	Hash Generation

Algorithm	Non-Approved Service
MD5	Hash Generation
SHA-384 (non-compliant)	Hash Generation
HMAC-MD5	HMAC Generation
HMAC SHA-384 (non-compliant)	HMAC Generation
HMAC SHA-512 (non-compliant)	HMAC Generation

3. Ports and interfaces

The logical interfaces to the module are implemented via an Application Programming Interface (API). The following table describes each logical interface.

Table 5: FIPS 140-2 Logical Interfaces.

Logical Interface	Description
Data Input	Input parameters that are supplied to the API commands
Data Output	Output parameters that are returned by the API commands
Control Input	API commands
Status Output	Exceptions and return status provided by API commands

4. Roles and Services

The module supports a Crypto Officer role and a User Role. The Crypto Officer installs and loads the module. The Crypto Officer also uses the services provided by the module. The User uses the cryptographic services provided by the module. The module provides the following services.

Table 6: Roles and Services

Service	Corresponding Roles	Types of Access to Cryptographic Keys and CSPs R – Read or Execute W – Write or Create Z – Zeroize
AES Encryption/Decryption	User Crypto Officer	AES Key: R
Secure Hash Generation	User Crypto Officer	None
HMAC Generation	User Crypto Officer	HMAC Key: R
PBKDF2 Key Derivation	User Crypto Officer	AES Key: W HMAC Key: W

Service	Corresponding Roles	Types of Access to Cryptographic Keys and CSPs R – Read or Execute W – Write or Create Z – Zeroize
		Password: R
Random Number Generation Using DRBG	User Crypto Officer	DRBG seed: R, W PBKDF2 salt: W AES key: W HMAC key: W
Self-test	User Crypto Officer	N/A
Zeroization	User Crypto Officer	All: Z
Session Key Zeroization	User Crypto Officer	Active Key: Z
Show Status	User Crypto Officer	N/A
Installation	Crypto Officer	N/A

Non-Approved cryptographic services are implementations of Non-Approved algorithms. They are listed in the Section 2.2.

5. Cryptographic Keys and CSPs

The table below describes cryptographic keys and CSPs used by the module.

Table 7: Cryptographic Keys and CSPs

Key	Description/Usage	Origin	Zeroization
AES Key	Used during AES encryption and decryption	Generated using CTR-DRBG or derived from password using PBKDF2[1]	Zeroized during power cycle or reboot
HMAC Key	Used during calculation of HMAC	Generated using CTR-DRBG or derived from password using PBKDF2[1]	Zeroized during power cycle or reboot
DRBG Seed	Used during generation of random numbers	Generated using NDRNG	Zeroized during power cycle or reboot
Password	Used to derive key using PBKDF2	Entered into the module by the operator	Zeroized during power cycle or reboot

[1] The operator shall not use the keys generated using PBKDF for any purposes other than protection of data stored inside the module.

The Keys and CSPs are stored in plaintext form in volatile memory within the software module. The software module does not store the Keys or CSPs in non-volatile storage.

Keys and CSPs used in the FIPS Approved mode of operation shall not be used while in the non-FIPS mode of operation. Keys or CSPs shall not be established while in the non-FIPS mode of operation.

6. Self-tests

The module performs the following power-up and conditional self-tests. Upon failure or a power-up or conditional self-test the module halts its operation and throws an exception.

The following table describes self-tests implemented by the module.

Table 8: Self-Tests

Algorithm	Test
AES	KAT (encryption/decryption)
SHA1	KAT
SHA256	KAT
SHA512	KAT
HMAC-SHA256	KAT (combined with integrity test)
PBKDF2	KAT
SP800-90A CTR_DRBG	KAT
	Continuous Random Number Generator test
NDRNG	Continuous Random Number Generator test