FIPS 140-2 Non-Proprietary Security Policy

IBM Security SiteProtector System Cryptographic Module (Version 3.1.1)

Document Version 1.10

February 12, 2016

Prepared For: IBM Security
Prepared By: SafeLogic Inc.

IBM Security
6303 Barfield Road
Atlanta, GA 30328
www.ibm.com

SafeLogic Inc.
530 Lytton Avenue, Ste. 200
Palo Alto, CA 94301
www.safelogic.com
FIPS 140-2 Non-Proprietary Security Policy: IBM Security SiteProtector System Cryptographic Module (Version 3.1.1)

Abstract

This document provides a non-proprietary FIPS 140-2 Security Policy for the SiteProtector System Cryptographic Module (Version 3.1.1).

Table of Contents

1 Introduction ................................................................. 4
   1.1 About FIPS 140-2 ..................................................... 4
   1.2 About this Document ............................................. 4
   1.3 External Resources ................................................. 4
   1.4 Notices ................................................................. 4
   1.5 Acronyms ............................................................. 5

2 IBM Security SiteProtector System Cryptographic Module (Version 3.1.1) ..................................................... 6
   2.1 Product Overview .................................................. 6
   2.2 Cryptographic Module Specification ..................... 6
   2.3 Validation Level Detail ........................................... 7
   2.4 Cryptographic Algorithms .................................... 7
      2.4.1 Algorithm Implementation Certificates .............. 7
      2.4.2 Non-Approved Algorithms .............................. 9
   2.5 Module Interfaces ................................................. 9
   2.6 Roles, Services, and Authentication ....................... 11
      2.6.1 Operator Services and Descriptions ............... 11
      2.6.2 Module API .................................................. 15
      2.6.3 Operator Authentication ................................ 19
   2.7 Physical Security .................................................. 19
   2.8 Operational Environment ....................................... 20
   2.9 Cryptographic Key Management ............................ 21
   2.10 Self-Tests .......................................................... 27
      2.10.1 Power-On Self-Tests .................................... 27
      2.10.2 Conditional Self-Tests ................................. 28
   2.11 Mitigation of Other Attacks ................................... 29

3 Guidance and Secure Operation ...................................... 31
   3.1 Crypto Officer Guidance ....................................... 31
      3.1.1 Software Packaging ...................................... 31
      3.1.2 Enabling FIPS Mode ...................................... 31
      3.1.3 Additional Rules of Operation ....................... 32
   3.2 User Guidance ..................................................... 33
      3.2.1 General Guidance ......................................... 33
FIPS 140-2 Non-Proprietary Security Policy: IBM Security SiteProtector System Cryptographic Module
(Version 3.1.1)

List of Tables

Table 1 – Acronyms and Terms..................................................................................................................................................5
Table 2 – Validation Level by DTR Section..............................................................................................................................7
Table 3 – Algorithm Certificates (GSKit) ...............................................................................................................................9
Table 4 – Logical Interface / Physical Interface Mapping .......................................................................................................11
Table 5 – Module Services and Descriptions and CSP access ..............................................................................................11
Table 6 - Key Zeroization API .............................................................................................................................................15
Table 7 – Module Keys/CSPs...............................................................................................................................................25

List of Figures

Figure 1 – Module Interfaces Diagram .................................................................................................................................10
1 Introduction

1.1 About FIPS 140-2

Federal Information Processing Standards Publication 140-2 — Security Requirements for Cryptographic Modules specifies requirements for cryptographic modules to be deployed in a Sensitive but Unclassified environment. The National Institute of Standards and Technology (NIST), Communications Security Establishment (CSE) and Cryptographic Module Validation Program (CMVP) runs the FIPS 140-2 program. The CMVP accredits independent testing labs to perform FIPS 140 testing; the CMVP also validates test reports for modules meeting FIPS 140-2 validation. Validated is the term given to a product that is documented and tested against the FIPS 140-2 criteria.


1.2 About this Document

This non-proprietary Cryptographic Module Security Policy for the SiteProtector System Cryptographic Module (Version 3.1.1) from IBM Security provides an overview of the product and a high-level description of how it meets the security requirements of FIPS 140-2. This document contains details on the module’s cryptographic keys and critical security parameters. This Security Policy concludes with instructions and guidance on running the module in a FIPS 140-2 mode of operation.

The IBM Security SiteProtector System Cryptographic Module (Version 3.1.1) may also be referred to as the “module” in this document.

1.3 External Resources

The IBM Security website (http://www.ibm.com) contains information on the full line of products from IBM Security, including a detailed overview of the SiteProtector System Cryptographic Module (Version 3.1.1) solution. The Cryptographic Module Validation Program website (http://csrc.nist.gov/groups/STM/cmvp/) contains links to the FIPS 140-2 certificate and IBM Security contact information.

1.4 Notices

This document may be freely reproduced and distributed in its entirety without modification.
1.5 Acronyms

The following table defines acronyms found in this document:

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES</td>
<td>Advanced Encryption Standard</td>
</tr>
<tr>
<td>CBC</td>
<td>Cipher Block Chaining</td>
</tr>
<tr>
<td>CSE</td>
<td>Communications Security Establishment</td>
</tr>
<tr>
<td>CSP</td>
<td>Critical Security Parameter</td>
</tr>
<tr>
<td>DTR</td>
<td>Derived Testing Requirement</td>
</tr>
<tr>
<td>ECDSA</td>
<td>Elliptic Curve Digital Signature Algorithm</td>
</tr>
<tr>
<td>FIPS</td>
<td>Federal Information Processing Standard</td>
</tr>
<tr>
<td>GPC</td>
<td>General Purpose Computer</td>
</tr>
<tr>
<td>GPOS</td>
<td>General Purpose Operating System</td>
</tr>
<tr>
<td>GSKit</td>
<td>IBM Global Security Kit</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HMAC</td>
<td>Hashed Message Authentication Code</td>
</tr>
<tr>
<td>IBM</td>
<td>International Business Machines</td>
</tr>
<tr>
<td>ISS</td>
<td>Internet Security Systems</td>
</tr>
<tr>
<td>KAT</td>
<td>Known Answer Test</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technology</td>
</tr>
<tr>
<td>RSA</td>
<td>Rivest Shamir Adelman</td>
</tr>
<tr>
<td>SHA</td>
<td>Secure Hashing Algorithm</td>
</tr>
</tbody>
</table>

Table 1 – Acronyms and Terms
2 IBM Security SiteProtector System Cryptographic Module (Version 3.1.1)

2.1 Product Overview

SiteProtector (http://www-03.ibm.com/software/products/en/site-protector-system) is a centralized management system that unifies management and analysis for network, server, and desktop protection agents and small networks or appliances. The SiteProtector is used as the central controlling point for IBM ISS appliances deployed on the network. The SiteProtector performs the following functionality:

- Manages and monitors Sensors and SiteProtector sub-components;
- Enables an administrator to view configuration data of an appliance supported by SiteProtector;
- Displays audit and system data records; and
- Monitors the network connection between SiteProtector and the Sensors it is configured to monitor.

2.2 Cryptographic Module Specification

The module is the IBM Security SiteProtector System Cryptographic Module (Version 3.1.1), provides the SiteProtector application with the means to encrypt management session to a managed Sensor. The module is a software-only module installed on a multi-chip standalone device, such as a General Purpose Computer running a General Purpose Operating System and provides cryptographic services to the IBM Security SiteProtector application.

The module is a uniquely identifiable library that is linked into the SiteProtector application. All operations of the module occur via calls from the SiteProtector application, which occur only when an operator is successfully authenticated to the host operating system. As such there are no untrusted services or daemons calling the services of the module. No security functions outside the cryptographic module provide FIPS-relevant functionality to the module.

The module is comprised of the following files:

- \ISS\SiteProtector\Agent Manager\agentmgr.dll
- \ISS\SiteProtector\Agent Manager\issSessionConfigSvcs5.dll
- \ISS\SiteProtector\Application Server\webserver\IHS\bin\issSessionConfigSvcs5.dll
- \ISS\SiteProtector\Application Server\webserver\IHS\modules\mod_ibm_ssl.so
- \ISS\SiteProtector\Event Collector\issSessionConfigSvcs5.dll
- \ISS\SiteProtector\FIPS Service\FipsService.exe
- \ISS\SiteProtector\Application Server\webserver\IHS\gsk8
- \ISS\SiteProtector\GSK\8.0.50.51
This module provides no non-FIPS approved mode of operation, and there is only one FIPS approved mode of operation. Although the module requires no further configuration or compilation, the procedures in the Guidance and Secure Operation must be followed.

### 2.3 Validation Level Detail

The following table lists the level of validation for each area in FIPS 140-2:

<table>
<thead>
<tr>
<th>FIPS 140-2 Section Title</th>
<th>Validation Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cryptographic Module Specification</td>
<td>1</td>
</tr>
<tr>
<td>Cryptographic Module Ports and Interfaces</td>
<td>1</td>
</tr>
<tr>
<td>Roles, Services, and Authentication</td>
<td>2</td>
</tr>
<tr>
<td>Finite State Model</td>
<td>1</td>
</tr>
<tr>
<td>Physical Security</td>
<td>N/A</td>
</tr>
<tr>
<td>Operational Environment</td>
<td>1</td>
</tr>
<tr>
<td>Cryptographic Key Management</td>
<td>1</td>
</tr>
<tr>
<td>Electromagnetic Interference / Electromagnetic Compatibility</td>
<td>1</td>
</tr>
<tr>
<td>Self-Tests</td>
<td>1</td>
</tr>
<tr>
<td>Design Assurance</td>
<td>2</td>
</tr>
<tr>
<td>Mitigation of Other Attacks</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Overall Validation Level**

<table>
<thead>
<tr>
<th>Validation Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

Table 2 – Validation Level by DTR Section

The “Mitigation of Other Attacks” section is not relevant as the module does not implement any countermeasures towards special attacks.

### 2.4 Cryptographic Algorithms

#### 2.4.1 Algorithm Implementation Certificates

The module’s cryptographic algorithm implementations have received the following certificate numbers from the Cryptographic Algorithm Validation Program:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>CAVP Certificate</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA Key Generation</td>
<td>186-4KEY(gen)</td>
<td>Sign / verify operations Key transport</td>
</tr>
<tr>
<td>(2048 or 3072 bits)</td>
<td>1676</td>
<td></td>
</tr>
<tr>
<td>RSA Signature Generation</td>
<td>PKCS#1.5</td>
<td></td>
</tr>
<tr>
<td>(2048 or 3072 bits)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(SHA-224,SHA-256,SHA-384,SHA-512)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Parameters</td>
<td>Cofactor DIFFIE-HELLEMAN Primitive</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>RSA Signature Verificaiton</td>
<td>PKCS#1.5 (1024, 2048, 3072 bits) (SHA-1,SHA-224,SHA-256,SHA-384,SHA-512)</td>
<td></td>
</tr>
<tr>
<td>ECDSA KeyPair Generation</td>
<td>P: 224, 256, 384, 521 K: 233, 283, 409, 571 B: 233, 283, 409, 571</td>
<td>632</td>
</tr>
<tr>
<td>ECDSA Signature Generation</td>
<td>P: 224, 256, 384, 521 K: 233, 283, 409, 571 B: 233, 283, 409, 571</td>
<td></td>
</tr>
<tr>
<td>ECDSA Signature Verification</td>
<td>P: 192, 224, 256, 384, 521 K: 163, 233, 283, 409, 571 B: 163, 233, 283, 409, 571</td>
<td></td>
</tr>
<tr>
<td>ECC CDH Component (SP800-56A)</td>
<td>P: 224, 256, 384, 521</td>
<td>462</td>
</tr>
<tr>
<td>SHA message digest generation</td>
<td>SHA-1, SHA-224, SHA-256, SHA-384, SHA-512</td>
<td></td>
</tr>
<tr>
<td>HMAC</td>
<td>HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512</td>
<td></td>
</tr>
<tr>
<td>AES</td>
<td>AES-128-CMAC, AES-192-CMAC, AES-256-CMAC, ECB, CBC, CFB1, CFB8, CFB128 &amp; OFB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AES_CCM 128, 192, or 256 bit keys (SP800-38C)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AES_GCM 128, 192, or 256 bit keys (FIPS 197, SP800-38D)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AES_XTS 128, 256 bit keys (FIPS SP800-38E)</td>
<td></td>
</tr>
</tbody>
</table>

1 AES XTS mode was CAVS validated but not implemented within the module.
**Triple-DES** | **Triple-DES 192-bit keys in ECB, CBC, CFB64, and OFB mode, CMAC** | **1866** | **Data encryption / decryption**
---|---|---|---
**DRBG 800-90A** | **HMAC_DRBG (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512), HASH_DRBG (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512), CTR_DRBG (AES-128-ECB, AES-192-ECB, AES-256-ECB)** | **737** | **DRBG**

| **Table 3 – Algorithm Certificates (GSKit)** |

**2.4.2 Non-Approved Algorithms**

The module implements the following Non-Approved Algorithms:

- True Random Number Generator (TRNG), a non-deterministic RNG (NDRNG) used to seed the DRBG.
- RSA (key wrapping; key establishment methodology provides 112 or 128 bits of encryption strength).
- Diffie-Hellman (key agreement; key establishment methodology provides 112 or 128 bits of encryption strength)
- EC Diffie-Hellman (CVL Cert. #462, key agreement; key establishment methodology provides between 128 and 256 bits of encryption strength)

**2.5 Module Interfaces**

The figure below shows the module’s physical and logical block diagram:
The interfaces (ports) for the physical boundary include the computer keyboard port, CDROM drive, floppy disk, mouse, network port, parallel port, USB ports, monitor port and power plug. When operational, the module does not transmit any information across these physical ports because it is a software cryptographic module. Therefore, the module’s interfaces are purely logical and are provided through the Application Programming Interface (API) that a calling daemon can operate. The logical interfaces expose services that applications directly call, and the API provides functions that may be called by a referencing application (see Section 2.6 – Roles, Services, and Authentication for the list of available functions).

The API provided by the module is mapped onto the FIPS 140-2 logical interfaces: data input, data output, control input, and status output. Each of the FIPS 140-2 logical interfaces relates to the module's callable interface, as follows:

<table>
<thead>
<tr>
<th>FIPS 140-2 Interface</th>
<th>Logical Interface</th>
<th>Module Physical Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Input</td>
<td>Input parameters of API function calls</td>
<td>Ethernet/Network port</td>
</tr>
<tr>
<td>Data Output</td>
<td>Output parameters of API function calls</td>
<td>Ethernet/Network port</td>
</tr>
<tr>
<td>Control Input</td>
<td>API function calls</td>
<td>Keyboard and mouse</td>
</tr>
</tbody>
</table>
Status Output

Uses the API function ICC_GetStatus that provides information about the status of the module and returns true or false. Either state is logged. The function is called once the context of the module has been obtained.

Monitor

Power

None

Power supply/connector

Table 4 – Logical Interface / Physical Interface Mapping

The module’s logical interfaces are provided only through the Application Programming Interface (API) that a calling daemon can operate. The module distinguishes between logical interfaces by logically separating the information according to the defined API.

As shown in Figure 1 – Module Interfaces Diagram and Table 5 – Module Services and Descriptions, the output data path is provided by the data interfaces and is logically disconnected from processes performing key generation or zeroization. No key information will be output through the data output interface when the module zeroizes keys.

2.6 Roles, Services, and Authentication

The module supports a Crypto Officer and a User role. The Crypto Officer (i.e., a human operator) can initialize and configure the module while the User role (i.e., SiteProtector) can only access the services of the module. The module does not support a Maintenance role.

2.6.1 Operator Services and Descriptions

<table>
<thead>
<tr>
<th>Service</th>
<th>Notes</th>
<th>Modes</th>
<th>CAVP</th>
<th>Keys and CSPs</th>
<th>Roles</th>
<th>Access Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Symmetric Algorithms</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AES encryption &amp; decryption</td>
<td>128, 192, or 256-bit keys (FIPS 197) Encrypt/Decrypt (with and without hardware support)</td>
<td>CBC, ECB, CFB1, CFB8, CFB128, OFB</td>
<td>3279</td>
<td>AES Symmetric key</td>
<td>Crypto Officer, User</td>
<td>Read/Write</td>
</tr>
<tr>
<td>Triple-DES encryption &amp; decryption</td>
<td>192-bit (of which 168 bits are key bits and the rest are parity bits) keys (SP 800-67) Encrypt/Decrypt</td>
<td>CBC, ECB, CFB64, OFB</td>
<td>1866</td>
<td>Triple-DES Symmetric key</td>
<td>Crypto Officer, User</td>
<td>Read/Write</td>
</tr>
<tr>
<td><strong>Public Key Algorithms</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ECDSA KeyPair Generation</td>
<td>P: 224, 256, 384, 521</td>
<td>N/A</td>
<td>632</td>
<td>ECDSA public and private key</td>
<td>Crypto Officer, User</td>
<td>Write</td>
</tr>
<tr>
<td></td>
<td>K: 233, 283, 409, 571</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>B: 233, 283, 409, 571</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service</td>
<td>Notes</td>
<td>Modes</td>
<td>CAVP</td>
<td>Keys and CSPs</td>
<td>Roles</td>
<td>Access Type</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>--------------------------------------------</td>
<td>-------------------------</td>
<td>------</td>
<td>-------------------------</td>
<td>---------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td><strong>ECDSA PKV</strong></td>
<td>P: 192, 224, 256, 384, 521</td>
<td>N/A</td>
<td>632</td>
<td>ECDSA key material</td>
<td>Crypto Officer User</td>
<td>Write</td>
</tr>
<tr>
<td></td>
<td>K: 163, 233, 283, 409, 571</td>
<td></td>
<td></td>
<td></td>
<td>User</td>
<td></td>
</tr>
<tr>
<td><strong>ECDSA Signature Generation</strong></td>
<td>P: 224, 256, 384, 521</td>
<td>N/A</td>
<td>632</td>
<td>ECDSA private key</td>
<td>Crypto Officer User</td>
<td>Read</td>
</tr>
<tr>
<td></td>
<td>K: 233, 283, 409, 571</td>
<td></td>
<td></td>
<td></td>
<td>User</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B: 233, 283, 409, 571</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ECDSA Signature Verification</strong></td>
<td>P: 192, 224, 256, 384, 521</td>
<td>N/A</td>
<td>632</td>
<td>ECDSA public key</td>
<td>Crypto Officer User</td>
<td>Read</td>
</tr>
<tr>
<td></td>
<td>K: 163, 233, 283, 409, 571</td>
<td></td>
<td></td>
<td></td>
<td>User</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B: 163, 233, 283, 409, 571</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>RSA Key Generation</strong></td>
<td>186-4KEY(gen) (2048 or 3072 bits)</td>
<td>N/A</td>
<td>1676</td>
<td>RSA public and private key</td>
<td>Crypto Officer User</td>
<td>Write</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>User</td>
<td></td>
</tr>
<tr>
<td><strong>RSA Signature Generation</strong></td>
<td>PKCS#1.5 (2048 or 3072 bits)</td>
<td>N/A</td>
<td>1676</td>
<td>RSA private key</td>
<td>Crypto Officer User</td>
<td>Read</td>
</tr>
<tr>
<td></td>
<td>(SHA-224,SHA-256,SHA-384,SHA-512)</td>
<td></td>
<td></td>
<td></td>
<td>User</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(with and without hardware support)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>RSA Signature Verification</strong></td>
<td>PKCS#1.5 (1024, 2048, 3072 bits)</td>
<td>N/A</td>
<td>1676</td>
<td>RSA public key</td>
<td>Crypto Officer User</td>
<td>Read</td>
</tr>
<tr>
<td></td>
<td>(SHA-1,SHA-224,SHA-256,SHA-384,SHA-512)</td>
<td></td>
<td></td>
<td></td>
<td>User</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(with and without hardware support)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>RSA Key Wrapping</strong></td>
<td>Encrypt / Decrypt (2048, 3072 bits)</td>
<td>N/A</td>
<td>1676</td>
<td>RSA public and private key</td>
<td>Crypto Officer User</td>
<td>Read</td>
</tr>
<tr>
<td></td>
<td>Allowed to be used in FIPS mode</td>
<td></td>
<td></td>
<td></td>
<td>User</td>
<td></td>
</tr>
<tr>
<td><strong>Diffie-Hellman (DH)</strong></td>
<td>2048 or 4096 bit modulus</td>
<td>Key agreement and Key Generation</td>
<td>N/A</td>
<td>DH public and private key</td>
<td>Crypto Officer User</td>
<td>Read/Write</td>
</tr>
<tr>
<td></td>
<td>Allowed to be used in FIPS mode</td>
<td></td>
<td></td>
<td></td>
<td>User</td>
<td></td>
</tr>
<tr>
<td><strong>EC Diffie-Hellman (ECDH)</strong></td>
<td>P: 224, 256, 384, 521 (SP 800-56A)</td>
<td>Key agreement and Key Generation</td>
<td>(ECC CDH component test #462)</td>
<td>ECDH public and private key</td>
<td>Crypto Officer User</td>
<td>Read/Write</td>
</tr>
</tbody>
</table>

**Hash Functions**
<table>
<thead>
<tr>
<th>Service</th>
<th>Notes</th>
<th>Modes</th>
<th>CAVP</th>
<th>Keys and CSPs</th>
<th>Roles</th>
<th>Access Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHA-1 message digest generation</td>
<td>FIPS 180-4 (not valid for signature generation)</td>
<td>N/A</td>
<td>2717</td>
<td>None</td>
<td>Crypto Officer User</td>
<td>N/A</td>
</tr>
<tr>
<td>SHA-224, SHA-256, SHA-384, SHA-512 message digest generation</td>
<td>FIPS 180-4, SHA-2 algorithms (with and without hardware support)</td>
<td>N/A</td>
<td>2717</td>
<td>None</td>
<td>Crypto Officer User</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Message Authentication Codes (MACs)**

| HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512 | FIPS 198, 198-1 (with and without hardware support)                   | N/A   | 2076 | HMAC-SHA-1 key, HMAC-SHA-224 key, HMAC-SHA-256 key, HMAC-SHA-384 key, HMAC-SHA-512 key | Crypto Officer User    | Write       |

**Encryption Functions**

| AES-128-CMAC, AES-192-CMAC, AES-256-CMAC | 128, 192, or 256 bit keys (FIPS 197) Encrypt/Decrypt (with and without hardware support) | N/A   | 3279 | CMAC-AES-128 key, CMAC-AES-192 key, CMAC-AES-256 key                           | Crypto Officer User    | Write       |
| Triple-DES CMAC (CMAC with three key Triple-DES) | 192-bit keys (FIPS 197)                                                                  | CBC   | 1866 | CMAC-Triple-DES key (192-bit)                                                 | Crypto Officer User    | Write       |
| AES_CCM                                      | 128, 192, or 256 bit keys (SP800-38C) (with and without hardware support)                | N/A   | 3279 | AES_CCM key                                                                    | Crypto Officer User    | Write       |
| AES_GCM                                      | 128, 192, or 256 bit keys (FIPS 197, SP800-38D) (with and without hardware support)     | N/A   | 3279 | AES_GCM key                                                                    | Crypto Officer User    | Write       |

**Random Bit Generator**
<table>
<thead>
<tr>
<th>Service</th>
<th>Notes</th>
<th>Modes</th>
<th>CAVP</th>
<th>Keys and CSPs</th>
<th>Roles</th>
<th>Access Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRBG 800-90A</td>
<td>SP 800-90A (with and without hardware support).</td>
<td>HMAC_DRBG (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512), HASH_DRBG (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512), CTR_DRBG (AES-128-ECB, AES-192-ECB, AES-256-ECB)</td>
<td>737</td>
<td>Seed</td>
<td>Crypto Officer User</td>
<td>Write</td>
</tr>
</tbody>
</table>

**FIPS 140-2 Functions**

<table>
<thead>
<tr>
<th>Service</th>
<th>Notes</th>
<th>CAVP</th>
<th>Keys and CSPs</th>
<th>Roles</th>
<th>Access Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure</td>
<td>Initializes the module for FIPS mode of operation</td>
<td>N/A</td>
<td>None</td>
<td>Crypto Officer</td>
<td>Execute</td>
</tr>
<tr>
<td>Self-Test</td>
<td>Performs self-tests on critical functions of module. Run during startup and periodically during operations to prevent any secure data from being released and to ensure all components are functioning correctly</td>
<td>N/A</td>
<td>None</td>
<td>Crypto Officer</td>
<td>Execute</td>
</tr>
<tr>
<td>Show Status</td>
<td>Shows status of the module</td>
<td>N/A</td>
<td>None</td>
<td>Crypto Officer</td>
<td>Execute</td>
</tr>
<tr>
<td>Zeroization</td>
<td>Zeroizes keys. Ephemeral CSPs are zeroized by the RAM clearing processes, and static CSPs are zeroized by uninstalling the module and formatting the hard drive</td>
<td>N/A</td>
<td>None</td>
<td>Crypto Officer</td>
<td>Execute</td>
</tr>
</tbody>
</table>

Table 5 – Module Services and Descriptions and CSP access
Secret keys, public/private keys, and CSPs are protected from unauthorized disclosure, unauthorized modification, and unauthorized substitution because only authorized users are allowed access to the GPOS and SiteProtector application. The SiteProtector application ensures that no keys or CSPs leave the physical boundary of the module in plaintext. The module does not output intermediate key values, nor does it generate keys with non-Approved key generation methods.

Ephemeral CSPs are zeroized by the RAM clearing processes, and static CSPs are zeroized by uninstalling the module and formatting the hard drive. All keys and CSPs are stored in memory, and zeroization has been implemented to ensure no traces are left of any CSPs upon termination of the service using the CSP. Zeroization has been implemented by overwriting the allocated memory buffer with zeros before freeing the memory to other uses. Any service using a CSP will zeroize the CSP upon normal termination and when transitioning into error states. Zeroization is initiated by terminating the process and powering off the module. Zeroization will complete before any other malicious command could compromise the keys currently being zeroized because the module will not process additional commands until it finishes executing the current command.

Key zeroization services are performed via the following API functions:

<table>
<thead>
<tr>
<th>Key Zeroization Services</th>
<th>API functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean up memory locations used by low-level arithmetic functions</td>
<td>ICC_BN_clear_free() ICC_BN_CTX_free()</td>
</tr>
<tr>
<td>Clean up symmetric cipher context</td>
<td>ICC_EVP_CIPHER_CTX_free()</td>
</tr>
<tr>
<td>Clean up RSA context</td>
<td>ICC_RSA_free()</td>
</tr>
<tr>
<td>Clean up Diffie-Hellman context</td>
<td>ICC_DH_free()</td>
</tr>
<tr>
<td>Clean up asymmetric key contexts</td>
<td>ICC_EVP_PKEY_free()</td>
</tr>
<tr>
<td>Clean up HMAC context</td>
<td>ICC_HMAC_CTX_free()</td>
</tr>
<tr>
<td>Clean up ECDSA and ECDH contexts</td>
<td>ICC_EC_KEY_free()</td>
</tr>
<tr>
<td>Clean up CMAC context</td>
<td>ICC_CMAC_CTX_free()</td>
</tr>
<tr>
<td>Clean up AES-GCM context</td>
<td>ICC_AES_GCM_CTX_free()</td>
</tr>
<tr>
<td>Clean up RNG context</td>
<td>ICC_RNG_CTX_free()</td>
</tr>
</tbody>
</table>

Table 6 - Key Zeroization API

It is the calling application’s responsibility to appropriately utilize the provided zeroization methods (i.e. API functions) as listed in the table above to clean up involved cryptographic contexts before they are released.

2.6.2 Module API

The following list enumerates the API functions supported. Functions marked with (CO) are crypto officer functions.
• ICC_EC_GROUP_set_asn1_flag
• ICC_EVP_CIPHER_CTX_flags
• ICC_EVP_CIPHER_CTX_set_flags
• ICC_GetStatus
• ICC_Init (CO)
• ICC_SetValue (CO)
• ICC_GetValue
• ICC_Attach (CO)
• ICC_Cleanup
• ICC_SelfTest
• ICC_GenerateRandomSeed
• ICC_OBJ_nid2sn
• ICC_EVP_get_digestbyname
• ICC_EVP_get_cipherbyname
• ICC_EVP_MD_CTX_new
• ICC_EVP_MD_CTX_free
• ICC_EVP_MD_CTX_init
• ICC_EVP_MD_CTX_cleanup
• ICC_EVP_MD_CTX_md
• ICC_EVP_CIPHER_CTX_new
• ICC_EVP_CIPHER_CTX_free
• ICC_EVP_CIPHER_CTX_init
• ICC_EVP_CIPHER_CTX_cleanup
• ICC_EVP_CIPHER_CTX_set_key_length
• ICC_EVP_CIPHER_CTX_set_padding
• ICC_EVP_CIPHER_CTX_cipher
• ICC_DES_random_key
• ICC_DES_set_odd_parity
• ICC_EVP_EncryptInit
• ICC_EVP_EncryptUpdate
• ICC_EVP_EncryptFinal
• ICC_EVP_DecryptInit
• ICC_EVP_DecryptUpdate
• ICC_EVP_DecryptFinal
• ICC_EVP_OpenInit
• ICC_EVP_OpenUpdate
• ICC_EVP_OpenFinal
• ICC_EVP_SignInit
• ICC_EVP_SignUpdate
• ICC_EVP_SignFinal
FIPS 140-2 Non-Proprietary Security Policy: IBM Security SiteProtector System Cryptographic Module
(Version 3.1.1)

• ICC_EVP_VerifyInit
• ICC_EVP_VerifyUpdate
• ICC_EVP_VerifyFinal
• ICC_EVP_ENCODE_CTX_new
• ICC_EVP_ENCODE_CTX_free
• ICC_EVP_EncodeInit
• ICC_EVP_EncodeUpdate
• ICC_EVP_EncodeFinal
• ICC_EVP_DecodeInit
• ICC_EVP_DecodeUpdate
• ICC_EVP_DecodeFinal
• ICC_RAND_bytes
• ICC_RAND_seed
• ICC_EVP_PKEY_decrypt
• ICC_EVP_PKEY_encrypt
• ICC_EVP_PKEY_new
• ICC_EVP_PKEY_free
• ICC_EVP_PKEY_size
• ICC_RSA_new
• ICC_RSA_generate_key
• ICC_RSA_check_key
• ICC_EVP_PKEY_set1_RSA
• ICC_EVP_PKEY_get1_RSA
• ICC_RSA_free
• ICC_RSA_private_encrypt
• ICC_RSA_private_decrypt
• ICC_RSA_public_encrypt
• ICC_RSA_public_decrypt
• ICC_i2d_RSAPrivateKey
• ICC_i2d_RSAPublicKey
• ICC_d2i_PrivateKey
• ICC_d2i_PublicKey
• ICC_EVP_PKEY_set1_DH
• ICC_EVP_PKEY_get1_DH
• ICC_DH_new
• ICC_DH_new_generate_key
• ICC_DH_check
• ICC_DH_free
• ICC_DH_size
• ICC_DH_compute_key
• ICC_DH_generate_parameters
• ICC_DH_get_PublicKey
• ICC_id2_DHparams
• ICC_d2i_DHparams
• ICC_RSA_size
• ICC_BN_CTX_new
• ICC_BN_CTX_free
• ICC_BN_mod_exp
• ICC_HMAC_CTX_new
• ICC_HMAC_CTX_free
• ICC_HMAC_Init
• ICC_HMAC_Update
• ICC_HMAC_Final
• ICC_BN_div
ICC_ECDSA_SIG_new
ICC_ECDSA_SIG_free
ICC_i2d_ECDSA_SIG
ICC_d2i_ECDSA_SIG
ICC_ECDSA_sign
ICC_ECDSA_verify
ICC_ECDSA_size
ICC_EVP_PKEY_set1_EC_KEY
ICC_EVP_PKEY_get1_EC_KEY
ICC_EC_KEY_new_by_curve_name
ICC_EC_KEY_new
ICC_EC_KEY_free
ICC_EC_KEY_generate_key
ICC_EC_KEY_get0_group
ICC_EC_METHOD_get_field_type
ICC_EC_GROUP_method_of
ICC_EC_POINT_new
ICC_EC_POINT_free
ICC_EC_POINT_get_affine_coordinates_G_Fp
ICC_EC_POINT_set_affine_coordinates_G_Fp
ICC_EC_POINT_get_affine_coordinates_G_F2m
ICC_AES_GCM_DecryptUpdate
ICC_AES_GCM_EncryptFinal
ICC_AES_GCM_DecryptFinal
ICC_AES_GCM_GenerateIV_NIST
ICC_GHASH
ICC_AES_CCM_Encrypt
ICC_AES_CCM_Decrypt
ICC_get_RNGbyname
ICC_RNG_CTX_new
ICC_RNG_CTX_free
ICC_RNG_CTX_Init
ICC_RNG_Generate
ICC_RNG_ReSeed
ICC_RNG_CTX_ctrl
ICC_RSA_sign
ICC_RSA_verify
ICC_EC_GROUP_get_degree
ICC_EC_GROUP_get_curve_GFp
ICC_EC_GROUP_get_curve_GF2m
ICC_EC_GROUP_get0_generator
ICC_i2o_ECPublicKey
ICC_o2i_ECPublicKey
ICC_BN_cmp
ICC_BN_add
ICC_BN_sub
ICC_BN_mod_mul
ICC_EVP_PKCS82PKEY
ICC_EVP_PKEY2PKCS8
ICC_PKCS8_PRIV_KEY_INFO_free
ICC_d2i_PKCS8_PRIV_KEY_INFO
ICC_i2d_PKCS8_PRIV_KEY_INFO
ICC_d2i_ECPKParameters
•ICC_i2d_ECPKParameters
•ICC_EC_GROUP_free
•ICC_EC_KEY_set_group
•ICC_EC_KEY_dup
•ICC_RSA_X931_derive_ex
•ICC_Init
•ICC_EC_GROUP_set_asn1_flag
•ICC_OPENSSL_cpuid_override
•ICC_OPENSSL_cpuid
•ICC_EVP_CIPHER_CTX_flags
•ICC_EVP_CIPHER_CTX_set_flags
•ICC_EVP_CIPHER_CTX_copy
•ICC_OPENSSL_HW_rand
•ICC_OPENSSL_rdtscX
•ICC_BN_is_prime_fasttest_ex

2.6.3 Operator Authentication

Operators authenticate to the module via the Microsoft Windows Server 2012 R2 (General Purpose Operating System), which implements a username/password authentication mechanism and enforces operator authentication prior to the operator utilizing any system services. Further, the Windows login authentication mechanism required to access the module distinguishes operators that have administrator rights on the computer system. The modules rely on this mechanism to distinguish an operator between the two supported roles. The module itself does not contain authentication data.

The GPOS will allow an operator to change roles only if the User knows the Crypto Officer password and vice versa. The operating system is responsible for ensuring previous authentication data is cleared upon powering off of the module.

Passwords for the Crypto-Officer and User role must be a minimum of 8 characters (see Secure Operation section of this document). The password can consist of alphanumeric values, a-z A-Z 0-9, yielding 62 choices per character. The probability of a successful random attempt is 1/62^8, which is less than 1/1,000,000.

The GPOS module will lock an account after 5 failed authentication attempts; thus, the maximum number of attempts in one minute is 5. Therefore, the probability of a success with multiple consecutive attempts in a one minute period is 5/62^8 which is less than 1/100,000.

2.7 Physical Security

This section of requirements does not apply to this module. The module is a software-only module and does not implement any physical security mechanisms.
2.8 Operational Environment

The cryptographic module were tested and validated on the following hardware platform:

- Intel Core i7-2600 @ 3.4GHz (1-CPU / 4-core)

The module runs on Microsoft Windows Server 2012 R2 Standard (Single-user mode), Version 6.3. The module’s software is entirely encapsulated by the cryptographic boundary (shown in Figure 1).

The GPC(s) used during testing are assumed to have met Federal Communications Commission (FCC) FCC Electromagnetic Interference (EMI) and Electromagnetic Compatibility (EMC) requirements for business use as defined by 47 Code of Federal Regulations, Part15, Subpart B.
# Cryptographic Key Management

The table below provides a complete list of Critical Security Parameters used within the module:

<table>
<thead>
<tr>
<th>Key/CSP Name</th>
<th>Description / Use</th>
<th>Generation</th>
<th>Storage</th>
<th>Establishment / Export</th>
<th>Interface</th>
<th>Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES Session Key</td>
<td>AES 128, 192, 256 encryption &amp; decryption of management traffic</td>
<td>Internal generation at installation by DRBG</td>
<td><strong>Storage</strong>: RAM plaintext</td>
<td><strong>Agreement</strong>: Via secure TLS tunnel</td>
<td><strong>Entry</strong>: NA</td>
<td><strong>Output</strong>: NA</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Type</strong>: Ephemeral</td>
<td></td>
<td></td>
<td><strong>Decrypt</strong></td>
<td><strong>Encrypt</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Association</strong>: The system is the one and only owner. Relationship is maintained by the operating system via protected memory.</td>
<td></td>
<td></td>
<td><strong>User</strong></td>
<td><strong>R W D</strong></td>
</tr>
<tr>
<td>Triple-DES Session Key</td>
<td>Triple-DES 192 encryption &amp; decryption of management traffic</td>
<td>Internal generation at installation by DRBG</td>
<td><strong>Storage</strong>: RAM plaintext</td>
<td><strong>Agreement</strong>: Via secure TLS tunnel</td>
<td><strong>Entry</strong>: NA</td>
<td><strong>Output</strong>: NA</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Type</strong>: Ephemeral</td>
<td></td>
<td></td>
<td><strong>Decrypt</strong></td>
<td><strong>Encrypt</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Association</strong>: The system is the one and only owner. Relationship is maintained by the operating system via protected memory.</td>
<td></td>
<td></td>
<td><strong>User</strong></td>
<td><strong>R W D</strong></td>
</tr>
<tr>
<td>HMAC key</td>
<td>HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512 for message verification</td>
<td>Internal generation at installation by DRBG</td>
<td><strong>Storage</strong>: RAM plaintext</td>
<td><strong>Agreement</strong>: NA</td>
<td><strong>Entry</strong>: NA</td>
<td><strong>Output</strong>: None</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Type</strong>: Ephemeral</td>
<td></td>
<td></td>
<td><strong>Establish Session</strong></td>
<td><strong>Crypto Officer</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Association</strong>: The system is the one and only owner. Relationship is maintained by the operating system via protected memory.</td>
<td></td>
<td></td>
<td><strong>User</strong></td>
<td><strong>R W D</strong></td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>Storage</td>
<td>Agreement</td>
<td>Configure</td>
<td>Crypto Officer</td>
<td>User</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------</td>
<td>-----------</td>
<td>------------</td>
<td>----------------</td>
<td>------</td>
</tr>
<tr>
<td><strong>Crypto Officer Password</strong></td>
<td>Alphanumeric passwords externally generated by a human user for authentication to the operating system.</td>
<td>on disk/obfuscated</td>
<td>NA</td>
<td>Configure</td>
<td>Crypto Officer</td>
<td>R W D</td>
</tr>
<tr>
<td><strong>User Password</strong></td>
<td>Alphanumeric passwords externally generated by a human user for authentication to the operating system.</td>
<td>on disk/obfuscated</td>
<td>NA</td>
<td>Configure</td>
<td>Crypto Officer</td>
<td>R W</td>
</tr>
<tr>
<td><strong>DRBG Seed Key</strong></td>
<td>256-bit value to seed the FIPS-approved DRBG</td>
<td>RAM plaintext</td>
<td>NA</td>
<td>Establish Session</td>
<td>Crypto Officer</td>
<td>None</td>
</tr>
<tr>
<td><strong>Entropy Input String</strong></td>
<td>Input value for entropy calculation</td>
<td>RAM plaintext</td>
<td>NA</td>
<td>Establish Session</td>
<td>Crypto Officer</td>
<td>None</td>
</tr>
</tbody>
</table>

**Protected memory.**

**Type:** Static

**Association:** controlled by the operating system

**Entry:** Manual entry via operating system

**Output:** NA
<table>
<thead>
<tr>
<th>Mechanism</th>
<th>V and C values</th>
<th>Storage: RAM plaintext</th>
<th>Type:</th>
<th>Association: The system is the one and only owner. Relationship is maintained by the operating system via protected memory.</th>
<th>Agreement:</th>
<th>Establish Session:</th>
<th>Crypto Officer:</th>
<th>User:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hash_DRBG</td>
<td>V and C values</td>
<td>Generated internally by non-Approved RNG</td>
<td>Ephemeral</td>
<td></td>
<td>NA</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>HMAC_DRBG</td>
<td>V and Key values</td>
<td>Generated internally by non-Approved RNG</td>
<td>Ephemeral</td>
<td></td>
<td>NA</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>CTR_DRBG</td>
<td>V and Key values</td>
<td>Generated internally by non-Approved RNG</td>
<td>Ephemeral</td>
<td></td>
<td>NA</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>RSA Private Key</td>
<td>Private asymmetric key for sign / verify operations and key establishment(^2) for SiteProtector TLS connections.</td>
<td>Internal generation</td>
<td>Storage: RAM plaintext</td>
<td>Type: Static</td>
<td>Association: The system is the one and only owner. Relationship is maintained by the operating system via protected memory.</td>
<td>Agreement: NA</td>
<td>Entry: NA</td>
<td>Output: Key handle from API request is output only to the SiteProtector application</td>
</tr>
<tr>
<td>RSA Public Key</td>
<td>Public asymmetric key for sign / verify operations and key establishment for SiteProtector TLS connections Encryption/Decryption of the Premaster Secret for entry/output</td>
<td>Internal generation</td>
<td>Storage: RAM plaintext</td>
<td>Type: Static</td>
<td>Association: The system is the one and only owner. Relationship is maintained by the operating system via X509 certificates.</td>
<td>Agreement: NA</td>
<td>Entry: NA</td>
<td>Output: Key handle from API request is output only to the SiteProtector application</td>
</tr>
<tr>
<td>ECDHE Private Key</td>
<td>Private asymmetric key for key establishment(^3) for</td>
<td>Internal generation</td>
<td>Storage: RAM plaintext</td>
<td>Type: Static</td>
<td>Agreement: NA</td>
<td>Entry: NA</td>
<td>Establish Session: Crypto Officer R W D</td>
<td></td>
</tr>
</tbody>
</table>

\(^2\) Key establishment methodology provides 112 or 128 bits of encryption strength

\(^3\) Key establishment methodology provides 112 or 128 bits of encryption strength
### Table 7 – Module Keys/CSPs

<table>
<thead>
<tr>
<th>ECDHE Public Key</th>
<th>SiteProtector TLS connections.</th>
<th>Internal generation</th>
<th>Storage: RAM plaintext</th>
<th>Agreement: NA</th>
<th>Entry: NA</th>
<th>Output: Key handle from API request is output only to the SiteProtector application</th>
<th>Establish Session</th>
<th>Crypto Officer R W D</th>
<th>User R</th>
</tr>
</thead>
</table>

Secret keys, public/private keys, and CSPs are protected from unauthorized disclosure, unauthorized modification, and unauthorized substitution because only authorized users are allowed access to the GPOS and SiteProtector application. The SiteProtector application ensures that no keys or CSPs leave the physical boundary of the module in plaintext. The module does not output intermediate key values, nor does it generate keys with non-Approved key generation methods.

Ephemeral CSPs are zeroized by the RAM clearing processes, and static CSPs are zeroized by uninstalling the module and formatting the hard drive. All keys and CSPs are stored in memory, and zeroization has been implemented to ensure no traces are left of any CSPs upon termination of the service using the CSP. Zeroization has been implemented by overwriting the allocated memory buffer with zeros before freeing the memory to other uses. Any service using a CSP will zeroize the CSP upon normal termination and when transitioning into error states. Zeroization is initiated by terminating the process and powering off the module. Zeroization will complete before any other malicious command could compromise the keys currently being zeroized because the module will not process additional commands until it finishes executing the current command.
The TLS protocol has not been reviewed or tested by the CAVP and CMVP. Please see NIST document SP800-131A for guidance regarding the use of non FIPS-approved algorithms.
2.10 Self-Tests

The module includes an array of self-tests that are run during startup and periodically during operations to prevent any secure data from being released and to ensure all components are functioning correctly. In the event of any self-test failure, the module/SiteProtector application will output an error to the audit log and will shutdown. In addition to self-test failures, successful loading of the module is also logged. To access status of self-tests, success or failure, the application provides access to the audit log. Status is viewable via operating environment’s audit mechanism and by verifying proper loading and operation of the SiteProtector application. While the module is running self-tests, the module will not output data. The SiteProtector application makes calls to the SiteProtector System Cryptographic Module (Version 3.1.1), and data will not be returned until the self-tests complete.

No keys or CSPs will be output when the module is in an error state. The module will halt and the process will terminate; as such, no data will be output via the data output interface. Additionally, the module does not support a bypass function, and the module does not allow plaintext cryptographic key components or other unprotected CSPs to be output on physical ports. No external software or firmware is allowed to be loaded in a FIPS mode of operation.

The following sections discuss the module’s self-tests in more detail.

2.10.1 Power-On Self-Tests

Power-on self-tests are run upon every initialization of the module and if any of the tests fail, the module will not initialize. The module will enter an error state and no services can be accessed by the users. The module implements the following power-on self-tests:

- Module integrity check for the GSKit cryptographic library is via 2048-bit CAVS-validated RSA public key (PKCS#1.5) and a single HMAC SHA-1 digest calculated over the module at the time it is created. This RSA public key is stored inside the static stub and relies on the operating system for protection. Self-test and library verification is performed at library load by hooking the shared library’s ‘call on load’ entry points.

- Module integrity checks for other SiteProtector modules are by digital signature verification based on a 3072-bit CAVS-validated RSA public key using SHA-256 hashing. The signatures are created when the modules are created by IBM. Signature verification is done by the SiteProtector FIPS service before shared library is loaded.

- RSA signature generation with 2048 modulus KAT

- RSA signature verification with 2048 modulus KAT

- ECDSA pairwise consistency test with P-384
2.10.2 Conditional Self-Tests

Conditional self-tests are on-demand tests and tests run continuously during operation of the module. If any of these tests fail, the module will enter an error state and no services can be accessed by the users. The module can be re-initialized to clear the error and resume FIPS mode of operation. The module performs the following conditional self-tests:
FIPS 140-2 Non-Proprietary Security Policy: IBM Security SiteProtector System Cryptographic Module (Version 3.1.1)

- Pairwise consistency test for RSA (Signature Generation, Signature Verification, Key Generation, Key Wrapping)
- Pairwise consistency test for ECDSA (KeyPair Generation, PKV, Signature Generation, Signature Verification)
- DRBG 800-90A
  - Health Tests compliant with SP 800-90A – Section 11.3.
  - The DRBG 800-90A generates a minimum of 8 bytes per request. If less than 8 bytes are requested, the rest of the bytes is discarded and the next request will generate new random data.
  - The first 8 bytes of every request is compared with the last 8 bytes requested, if the bytes match an error is generated.
  - For the first request made to any instantiation of a DRBG 800-90A, two internal 8 byte cycles are performed.
  - The DRBG 800-90A relies on the environment (i.e. proper shutdown of the shared libraries) for resistance to retrospective attacks on data.
  - The DRBG 800-90A performs known answer tests when first instantiated and health checks at intervals as specified in the standard.
- True Random Number Generator (TRNG)
  - A non-deterministic RNG (NDRNG) is used to seed the DRBG. Every time a new seed or n bytes is required (either to initialize the DRBG, reseed the DRBG periodically or reseed the DRBG by user’s demand), the cryptographic module performs a comparison between the SHA-256 message digest using the new seed and the previously calculated digest. If the values match, the TRNG generates a new stream of bytes until the continuous DRBG test passes.

The module will inhibit data output via the output interface when conditional tests are performed. Once the tests have passed and the keys have been generated, the module will pass the key to the calling daemon.

### 2.11 Mitigation of Other Attacks

The module does not mitigate other attacks.
3 Guidance and Secure Operation

This section describes how to configure the module for FIPS-approved mode of operation. Operating the module without maintaining the following settings will remove the module from the FIPS-approved mode of operation.

3.1 Crypto Officer Guidance

3.1.1 Software Packaging

The module is included with SiteProtector Version 3.1.1 and is available for direct download. The SiteProtector application (and subsequently the module) is to be installed on a Windows Server 2012 R2 (Single-user mode) operating system.

3.1.2 Enabling FIPS Mode

To meet the cryptographic security requirements, especially for secure communication, certain restrictions on the installation and use of SiteProtector must be followed. The steps below will ensure that the module implements all required self-tests and uses only approved algorithms.

Only the Express install package is supported. Other installation options are not valid. To install SiteProtector, please follow these steps:

Installation

1. Install the packages on the machine intended to run SiteProtector and in this order:
   1) SiteProtectorExpress-Setup.exe
   2) EventArchiver-Setup.exe (optional for UCR validation)
   3) FIPSService-Setup.exe.
2. All SiteProtector components must be installed on a single hardware / OS platform. The only exception to this rule is that the management Console may be installed and used remotely.
3. The installation must be a new install. Upgrading from a previous version of SiteProtector is not valid.

Configure

1. Install the License.
   A license is required to update SiteProtector components to the latest version.
   To install the license in the SiteProtector Console:
   a. Select Tools -> Licenses -> Agent/Module
   b. Select the Licenses tab (2nd tab)
   c. Select the Add button.
   d. Location and select the provided SiteProtector license key.

4 Note that upgrading from a previous Express Install is not supported and a clean install is required.
FIPS 140-2 Non-Proprietary Security Policy: IBM Security SiteProtector System Cryptographic Module (Version 3.1.1)

e. Press the F5 key to refresh the License tab until all licenses display the state "Key Good".
f. Select the OK button to close the License dialog.

2. Reboot the machine.

3. Apply the component updates.
Several SiteProtector components will display update status of "Out of Date".
The component updates should be installed in this order:
1. Database / Product Maintenance
2. Database / Product Features
3. SiteProtector Core (additional database updates may need now appear and requires updating)
4. Event Collector
5. Agent Manager

To install an update to a SiteProtector component from the Console, in the Agent view:
a. Right-click the component.
b. Select Updates -> Apply XPU...
c. Select Accept on the license agreement page.
d. Select Next on the Schedule Update page.
e. Select Finish on the Select XPU page.
Note: The X-press Update Server and Event Archiver (optional) will update automatically.

When all update statuses show "Current", the component versions should be:
- Database 3.1.1.12 (XPU 1.468) [or greater]
- Event Collector 3.1.1.2 [or greater]
- FIPS Service 3.1.1 [or greater]
- SiteProtector Core 3.1.1.2 [or greater]
- Security Fusion Module 3.0 [or greater]
- Agent Manager 3.1.1.7 [or greater]
- X-press Update Server 3.1.1.2 [or greater]
- Event Archiver 3.1.1.2 (optional) [or greater]

All SiteProtector components must be installed on a single hardware / OS platform.
The installation must be a new install. Upgrading from a previous version of SiteProtector is not valid.

Remote management is allowed as long as the module implements IBM® Java JCE FIPS 140-2 Cryptographic Module (Software Version: 1.7), FIPS 140-2 Certificate #1993. Optionally, download Console-Setup.exe and install on a different machine than SiteProtector is installed.

3.1.3 Additional Rules of Operation

1. All host system components that can contain sensitive cryptographic data (main memory, system bus, disk storage) must be located in a secure environment.
2. The writable memory areas of the Module (data and stack segments) are accessible only by the SiteProtector application so that the Module is in "single user" mode, i.e. only the SiteProtector application has access to that instance of the Module.

3. The operating system is responsible for multitasking operations so that other processes cannot access the address space of the process containing the Module.

3.2 User Guidance

3.2.1 General Guidance

The User must configure and enforce the following initialization procedures in order to operate in FIPS approved mode of operation: the end user of the operating system is responsible for zeroizing CSPs by via wipe/secure delete procedures.