# *Accellion kiteworks Cryptographic Module*
# *FIPS 140-2 Non-Proprietary Security Policy*

Document *Version 1.1*

## *Accellion, Inc.*

October 31, 2014

**TABLE OF CONTENTS**

# 1 Module Overview

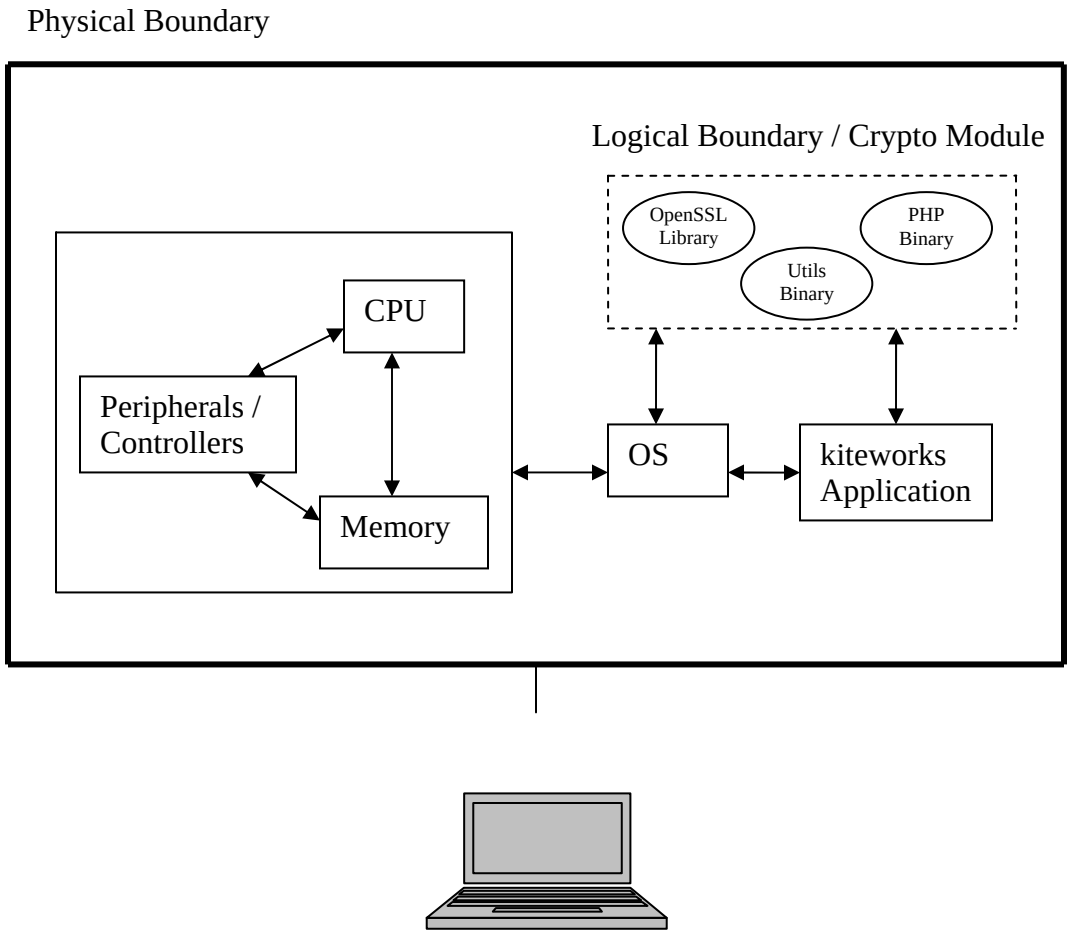
The Accellion kiteworks Cryptographic Module (SW Version KWLIB_1_0_1) is a software only module that operates in a multi-chip standalone embodiment, as defined in the FIPS 140-2 standard. The physical boundary is defined as being the outer perimeter of the general purpose computer on which the software module is installed. The logical boundary is defined as the collection of the shared libraries which are as follows:

- OpenSSL Library: /opt/lib/libcrypto.so.1.0.0 and /opt/lib/libssl.so.1.0.0 [Version: 1.0.1g]
- PHP Binary: /opt/bin/php [Version: 5.5.10]
- Utils Binary: fips_utils [Version: 1.1]

The module implements OpenSSL 1.0.1g which properly handles Heartbeat Extension packets. This Module is not susceptible to the Heartbleed vulnerability.

The primary purpose for this device is to provide data security for file transfers and sharing.

**Figure 1 – Block Diagram of the Cryptographic Module**



# 2 Security Level

The Accellion kiteworks Cryptographic Module meets the overall requirements applicable to Security Level 1 of FIPS 140-2.

**Table 1 - Module Security Level Specification**

| Security Requirements Section | Level |
|---|---|
| Cryptographic Module Specification | 1 |
| Module Ports and Interfaces | 1 |
| Roles, Services and Authentication | 1 |
| Finite State Model | 1 |
| Physical Security | N/A |
| Operational Environment | 1 |
| Cryptographic Key Management | 1 |
| EMI/EMC | 1 |
| Self-Tests | 1 |
| Design Assurance | 2 |
| Mitigation of Other Attacks | N/A |

# 3 Modes of Operation

The Accellion kiteworks Cryptographic Module has a FIPS Approved mode of operation and a non-FIPS Approved mode of operation. It is placed into FIPS mode of operation when initialized with a valid license key. The operator can determine if the cryptographic module is running a FIPS certified version by comparing the version on the "About" page to the version on the FIPS certificate.

## 3.1 Approved Mode of Operation

### *3.1.1 Approved Algorithms*

The Accellion kiteworks Cryptographic Module supports the following FIPS Approved algorithms within the libraries:

OpenSSL Library

- AES CBC mode with 128 and 256 bit keys for encryption and decryption in TLS (Cert. #2850)
- Triple-DES TCBC mode using 3-keys for encryption and decryption in TLS (Cert. #1703)
- HMAC-SHA-1 and HMAC-SHA-256 for MAC and key derivation in TLS (Cert. #1790)
- HMAC-SHA-512 for token authentication in inter-server communication (Cert. #1790)

- SHA-1 and SHA-256 for hashing and key derivation in TLS (Cert. #2392)
- SHA-224 and SHA-384 for hashing (Cert. #2392)
- SHA-512 for hashing and within HMAC (Cert. #2392)
- SP 800-135 TLSv1.0/v1.1/v1.2 KDF for deriving TLS Session Keys (CVL Cert. #286)
- FIPS 186-4 RSA with 1024/2048/3072 bit keys for digital signature verification (Cert. #1492)
- FIPS 186-4 RSA with 2048 bit keys for digital signature generation (Cert. #1492)
- SP 800-90A CTR_DRBG using AES 256, security strength is 256 bits (Cert. #503)

PHP Binary

- HMAC-SHA-1 for OAuth Signature Flow Authentication (Cert. #1791)
- HMAC-SHA-512 for token authentication in inter-server communication (Cert. #1791)
- SHA-1/SHA-512 for hashing and within HMAC (Cert. #2393)
- SHA-256 for hashing (Cert. #2393)

### *3.1.2 Non-Approved but Allowed Functions*

The module supports the following FIPS non-Approved but allowed algorithms and protocols:

- TLS v1.0/SSL v3.1 and TLS v1.1 for secure communications and key establishment (acting as client or server) with the following cipher suites:
  - TLS_RSA_WITH_AES_128_CBC_SHA (with RSA modulus >=2048 bits)
  - TLS_RSA_WITH_AES_256_CBC_SHA (with RSA modulus >=2048 bits)
  - TLS_RSA_WITH_3DES_EDE_CBC_SHA (with RSA modulus >=2048 bits)
- TLS v1.2 for secure communications and key establishment (acting as client or server) with the following cipher suites:
  - TLS_RSA_WITH_AES_128_CBC_SHA256 (with RSA modulus >=2048 bits)
  - TLS_RSA_WITH_AES_256_CBC_SHA256 (with RSA modulus >=2048 bits)
- RSA (key wrapping using 2048 to 16,384-bit keys; key establishment methodology provides between 112 and 256 bits of encryption strength)
- NDRNG to seed the *SP 800-90A* DRBG
- MD5 within TLS key derivation (as allowed per IG D.9)

The module supports the following FIPS non-Approved but allowed algorithms that do not support any security relevant operations:

- MD5 for hashing (no security claimed)

### 3.2 Non-Approved Mode of Operation

The module utilizes open source libraries which contain many algorithms that have not been CAVP tested and are therefore considered non-Approved. It is not recommended that the operator utilizes algorithms other than those listed above. If any non-Approved algorithm listed below is used, the module enters a non-Approved mode of operation:

OpenSSL Library

- AES (non-compliant):
  - CBC mode: 192
  - CBC-HMAC-SHA-1
  - CFB, CFB1, CFB8, CTR, ECB, OFB modes
  - XTS
  - GCM
  - CMAC
- DRBG (non-compliant): HASH, HMAC, Dual EC, CTR with 128/192 or no DF
- DSA (non-compliant)
- ECDSA (non-compliant)
- HMAC (non-compliant): HMAC-SHA-224, HMAC-SHA-384
- RNG (non-compliant): ANSI x931
- RSA (non-compliant):
  - Signature Generation: 1024 with any SHA, 2048 with SHA-1, 3072 with SHA-1
  - Key Wrapping: <2048
  - Any function using hashes: MD4, MD5, MDC2, RIPEMD-160
- Triple-DES (non-compliant):
  - 2-key
  - 3-key with CFB, CFB1, CFB8, OFB
  - CMAC
- PKCS #3 Diffie-Hellman Key Agreement
- Blowfish
- CAMELLIA
- CAST5
- DES
- DESX
- IDEA
- MDC2
- MD4
- RC2
- RC4
- RC4-HMAC-MD5

- RIPEMD-160
- SEED
- SSLeay
- Whirlpool

PHP Binary

- PHP Built-in functions to generate random numbers:
  - rand(). Details: http://sg3.php.net/manual/en/function.rand.php
  - mtrand(). Details: http://sg3.php.net/manual/en/function.mt-rand.php
- Hashing algorithms:
  - MD2
  - MD4
  - MD5
  - SHA-224 (non-compliant)
  - SHA-384 (non-compliant)
  - RIPEMD (RIPEMD-128, -160, -256, -320)
  - Whirlpool
  - Tiger (Tiger-128,3, -160,3, -192,3, -128,4, -160,4, -192,4)
  - snefru, snefru256
  - gost
  - adler32
  - crc32, crc32b
  - fnv132, fnv164
  - joaat
  - haval (haval-128,3, -160,3, -192,3, -224,3, -256,3, -128,4, -160,4, 192,4, -224,4, -256,4, -128,5, -160,5, 192,5, 224,5, 256,5)
- HMAC (non-compliant):
  - If using any of above listed hashing algorithms.
  - If key size is <112 bits.

# 4 Ports and Interfaces

The physical ports of the module are provided by the general purpose computer on which the module is installed. The module supports the following logical interfaces:

- Data input: The data input interface consists of the input parameters of the shared libraries' functions.
- Data output: The data output interface consists of the output parameters of the shared libraries' functions.
- Control input: The control input interface consists of the actual functions of the shared libraries.
- Status output: The status output interface includes the return values of the functions of the shared libraries.

# 5 Identification and Authentication Policy

The Accellion kiteworks Cryptographic Module supports two distinct operator roles (User, Cryptographic Officer). In compliance with FIPS 140-2 Level 1 standards, the module does not support operator authentication for those roles. However, only one role may be active at a time and the module does not allow concurrent operators.

The User and Cryptographic Officer roles are implicitly assumed by the entity accessing services implemented by the module.

User Role: Initialize the module and perform any of the module services. This role has access to all of the services provided by the module.

Cryptographic Officer Role: Installation of the module on the host computer system.

# 6 Access Control Policy

## 6.1 Roles and Services

**Table 2 – Roles and Services**

| Crypto Officer Role | User Role | No Role Required | FIPS Approved Mode | Non-FIPS Approved Mode | Service Name | Service Description |
|---|---|---|---|---|---|---|
| X | | | X | | Module Installation | Install the module on the host computer system |
| | X | | X | | Symmetric Encryption/Decryption | This service provides encryption/decryption functionality for AES and Triple-DES ciphers. |

| Crypto Officer Role | User Role | No Role Required | FIPS Approved Mode | Non-FIPS Approved Mode | Service Name | Service Description |
|---|---|---|---|---|---|---|
| | X | | X | | RSA Key Wrapping/Unwrapping for Key Transport | This service provides key wrapping/unwrapping functionality. |
| | X | | X | | Digital Signature Verification | This service provides functionality to verify digital signatures. |
| | X | | X | | Digital Signature Generation | This service provides functionality to generate digital signatures. |
| | X | | X | | Keyed Hash (HMAC) | This service provides keyed hash functionality using HMAC-SHA1 or HMAC-SHA-256 or HMAC-SHA-512. |
| | X | | X | | Message Digest (SHS) | This service provides functionality to generate message digests using SHA-1 or SHA-256 or SHA-224 or SHA-384 or SHA-512. |
| | X | | X | | TLS Session Establishment | This service establishes TLS sessions and generates TLS session keys. It uses a restricted set of cipher suites that only use FIPS Approved algorithms. |
| | X | | X | | MD5 | This service generates MD5 hashes (no security is given by this service). |
| | X | | X | | Misc/Helper Service | This service helps in performing the services mentioned above |
| | X | | | X | Non-Approved OpenSSL APIs | This is a catch-all service for all other OpenSSL APIs that are present but use non-CAVP tested and non-Approved algorithms. It is not recommended that the User utilize these services. |
| | X | X | X | | Self-Tests | On bootup of the host GPC and/or module instantiation, automatically runs the self-tests necessary for FIPS 140-2. |

| Crypto Officer Role | User Role | No Role Required | FIPS Approved Mode | Non-FIPS Approved Mode | Service Name | Service Description |
|---|---|---|---|---|---|---|
| | X | X | X | | Zeroization | All the CSPs can be zeroized through an API call or by powering down the GPC. |
| | X | X | X | | Show Status | The operator can obtain the current status of the module. |

*Note: Detailed services listings and API mappings per FIPS IG 3.5 can be found at: www.accellion.com/KW_FIPS_Service_API_mapping.pdf.*

### 6.2 Definition of Critical Security Parameters (CSPs)

The following are CSPs contained in the module:

- Accellion TLS Key: This is a RSA 2048 bit private key used for TLS connections. This key is shipped from the factory and is to be replaced by the Customer TLS Key.
- Customer TLS Key: This is a RSA 2048 to 16384 bit private key used for TLS connections. This key is loaded by the customer and replaces the Accellion TLS Key.
- TLS Session Keys: The set of ephemeral Triple-DES or AES 128/256 and HMAC keys created for each TLS session.
- Entropy Input to DRBG: This is the seed to the DRBG.
- DRBG State (V and Key): These are the state variables for the DRBG.
- License Key: This is an AES 256 bit key used to decrypt the license file.
- MySQL Password Key: This is an AES 256 bit key used to encrypt the MySQL DB password.
- User ECM Password Key: This is an AES 256 bit key used to encrypt user's ECM password.
- Settings Key: This is an AES 256 bit key used to encrypt settings that contain sensitive information like password before storing it in DB.
- HMAC OAuth Key: This key is used to calculate the HMAC-SHA-1 digest in one of the OAuth authentication flows.
- HMAC Token Key: This key is used to calculate the HMAC-SHA-512 digest which is used in the token authentication in inter-server communication.
- HMAC Software Integrity Key: This key is used to calculate the HMAC-SHA-1 digest of

the module which is then used in the software integrity test.

- SAML Key: This is a RSA 2048 bit private key used to digitally sign the SAML response.
- Galera Key: This is a RSA 2048 bit private key used in MySQL TLS session.

### 6.3 Definition of Public Keys

The following are the public keys contained in the module:

- RSA Public Key – SAML: This is a RSA 2048 bit public key used to check the signature of the SAML response.
- RSA Public Key – Galera: This is a RSA 2048 bit public key used in MySQL TLS session.
- RSA Public Key - License: This is a RSA 1024 bit public key used to check the signature of the license. (Note: 1024 bit modulus is still allowed for legacy use with signature verification).
- RSA Public Key – TLS: This is a RSA 2048 bit public key used in TLS which can be replaced by the customer.
- Third Party RSA Public Keys – TLS: This is a RSA 2048 to 16384 bit public key which is sent from third party users and used in TLS communications.
- Third Party RSA Public Keys – SAML: This is a RSA 1024 to 16384 bit public key which is generated outside the module and provided by the operator. (Note: 1024 bit modulus is still allowed for legacy use with signature verification).

### 6.4 Definition of CSPs/Public Key Modes of Access

Table 3 defines the relationship between access to CSPs/Public Keys and the different module services. The modes of access shown in the table are defined as follows:

- Generate (G): This operation generates the identified CSP.
- Use (U): This operation uses the identified CSP.
- Input (I): This operation receives the identified CSP from outside the GPC.
- Output (O): This operation outputs the identified CSP outside of the GPC.
- Zeroize (Z): This operation removes the identified CSP.

**Table 3 – CSP/Public Key Access Rights within Roles & Services**

| Services | CSPs | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Accellion TLS Key | Customer TLS Key | TLS Session Keys | Entropy Input to DRBG | DRBG State (V and Key) | License Key | MySQL Password Key | User ECM Password Key | Settings Key | HMAC OAuth Key | HMAC Token Key | HMAC Software Integrity Key | SAML Key | Galera Key | RSA Public Key – SAML | RSA Public Key - Galera | RSA Public Key - License | RSA Public Key - TLS | Third Party RSA Public Keys – TLS | Third Party RSA Public Keys - SAML |
| Module Installation | U | U | U | U | U | U | U | U | U | U | U | U | U | U | U | U | U | U | | |
| Symmetric Encryption/Decryption | | I | U | | | U | U | U | U | I | | | | | | | | I | | I |
| RSA Key Wrapping/Unwrapping for Key Transport | U | U | | | | | | | | | | | | U | | U | | U | U | |
| Digital Signature Verification | | | | | | | | | | | | | | | U | | U | | | U |
| Digital Signature Generation | | | | | | | | | | | | | U | | | | | | | |
| Keyed Hash (HMAC) | | | | | | | | | | U | U | | | | | | | | | |
| Message Digest (SHS) | | | | | | | | | | | | | | | | | | | | |
| TLS Session Establishment | U | U | G | G | G | | | | | | | | | | | | | U | UI | |
| MD5 | | | | | | | | | | | | | | | | | | | | |
| Misc/Helper Service | | | | | | | | | | | | | | | | | | | | |
| Non-Approved OpenSSL APIs | | | | | | | | | | | | | | | | | | | | |
| Self-Tests | | | | | | | | | | | | U | | | | | | | | |
| Zeroization | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | Z | | | | | | |
| Show Status | | | | | | | | | | | | | | | | | | | | |

# 7 Operational Environment

The Accellion kiteworks Cryptographic Module is a software module that runs on an underlying modifiable operational environment and is installed on a general purpose computer. The module is composed of the shared libraries described in Section 1 above.

When a crypto module is implemented in Accellion's kiteworks environment, the kiteworks application is the user of the cryptographic module. The kiteworks application makes the calls to the cryptographic module. Therefore, the kiteworks application is the single user of the cryptographic module, and satisfies the FIPS 140-2 requirement for a single user mode of operation, even when the kiteworks application is serving multiple clients.

The Accellion kiteworks Cryptographic Module has been tested on CentOS 6.4 on VMware ESXi 5.1.0.

# 8 Security Rules

The Accellion kiteworks Cryptographic Module's design corresponds to the cryptographic module's security rules. This section documents the security rules enforced by the cryptographic module to implement the security requirements of this FIPS 140-2 Level 1 module.

1. The cryptographic module shall provide the following distinct operator roles:
   - User role
   - Cryptographic Officer role
2. When the module has not been placed in a valid role, the operator shall not have access to any cryptographic services.
3. The cryptographic module shall encrypt message traffic using the TLS v1.0/SSL v3.1, TLS v1.1, or TLS v1.2 protocol.
4. The cryptographic module shall perform the following tests:

   A. Power up Self-Tests:

      1. Cryptographic algorithm tests:

      OpenSSL Library

         a. AES CBC 128/256 bit encryption KAT and decryption KAT
         b. Triple-DES TCBC 3-key encryption KAT and decryption KAT
         c. HMAC-SHA-1, HMAC-SHA-256, and HMAC-SHA-512 KATs
         d. SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512 KATs
         e. SP 800-135 TLS KDF KAT
         f. FIPS 186-4 RSA 1024 bit signature verification KAT
         g. FIPS 186-4 RSA 2048 bit signature generation KAT
         h. DRBG KAT
         i. RSA key wrapping and unwrapping KAT

PHP Binary

j. HMAC-SHA-1 and HMAC-SHA-512 KATs
k. SHA-1, SHA-256, and SHA-512 KATs

2. Software Integrity Test – HMAC-SHA-1
3. Critical Functions Tests: None

B. Conditional Self-Tests:

1. NDRNG Continuous Test
2. DRBG Continuous Test

5. At any time the cryptographic module is in an idle state, the operator shall be capable of commanding the module to perform the power-up self-test (i.e., by rebooting the GPC). If a power up self-test or the integrity test fails, the module provides the following error: "Failed FIPS test when running. Going into FIPS error state." The module is then uninstalled and the GPC is shutdown.
6. If the DRBG continuous self-test fails, the module provides the following error: "DRBG continuous test failed, going to fips error state". If the NDRNG continuous self-test fails, the module provides the following error: "NDRNG continuous test failed, going to fips error state". In either case, the module is then uninstalled and the GPC is shutdown.
7. Status information shall not contain CSPs or sensitive data that if misused could lead to a compromise of the module.
8. Third party clients using this module shall restrict their TLS ciphers suites to those listed in Section 3 of this document.
9. Third party clients using this module shall restrict their TLS RSA modulus to be >=2048 bits.
10. Customer TLS Keys used to replace the Accellion TLS Key shall be restricted to RSA with a modulus >= 2048 bits.

# 9 Physical Security Policy

The Accellion kiteworks Cryptographic Module is a software module intended for use with CentOS 6.4 on VMware ESXi 5.1.0; therefore, the physical security requirements of FIPS 140-2 are not applicable.

# 10 Mitigation of Other Attacks Policy

The module has not been designed to mitigate specific attacks outside of the scope of FIPS 140-2.

# 11 Definitions and Acronyms

| | |
|---|---|
| **AES** | Advanced Encryption Standard |
| **API** | Application Program Interface |
| **CO** | Cryptographic Officer |
| **CSP** | Critical Security Parameter (as defined in FIPS 140-2) |
| **DES** | Data Encryption Standard |
| **DRBG** | Deterministic Random Bit Generator |
| **EMC** | Electromagnetic Compatibility |
| **EMI** | Electromagnetic Interference |
| **FIPS** | Federal Information Processing Standard |
| **HMAC** | Keyed-Hash Message Authentication Code |
| **KDF** | Key Derivation Function |
| **MD5** | Message-Digest Algorithm 5 |
| **NDRNG** | Non-Deterministic Random Number Generator |
| **RSA** | Rivest, Shamir and Adleman Algorithm |
| **KW** | kiteworks |
| **SHA** | Secure Hash Algorithm |
| **SSL** | Secure Sockets Layer |
| **Triple-DES** | Triple-Data Encryption Standard (Algorithm) |
| **TLS** | Transport Layer Security |