# *Accellion Secure File Transfer Cryptographic Module Security Policy*

Document *Version 1.0*

## *Accellion, Inc.*

December 24, 2009

**TABLE OF CONTENTS**

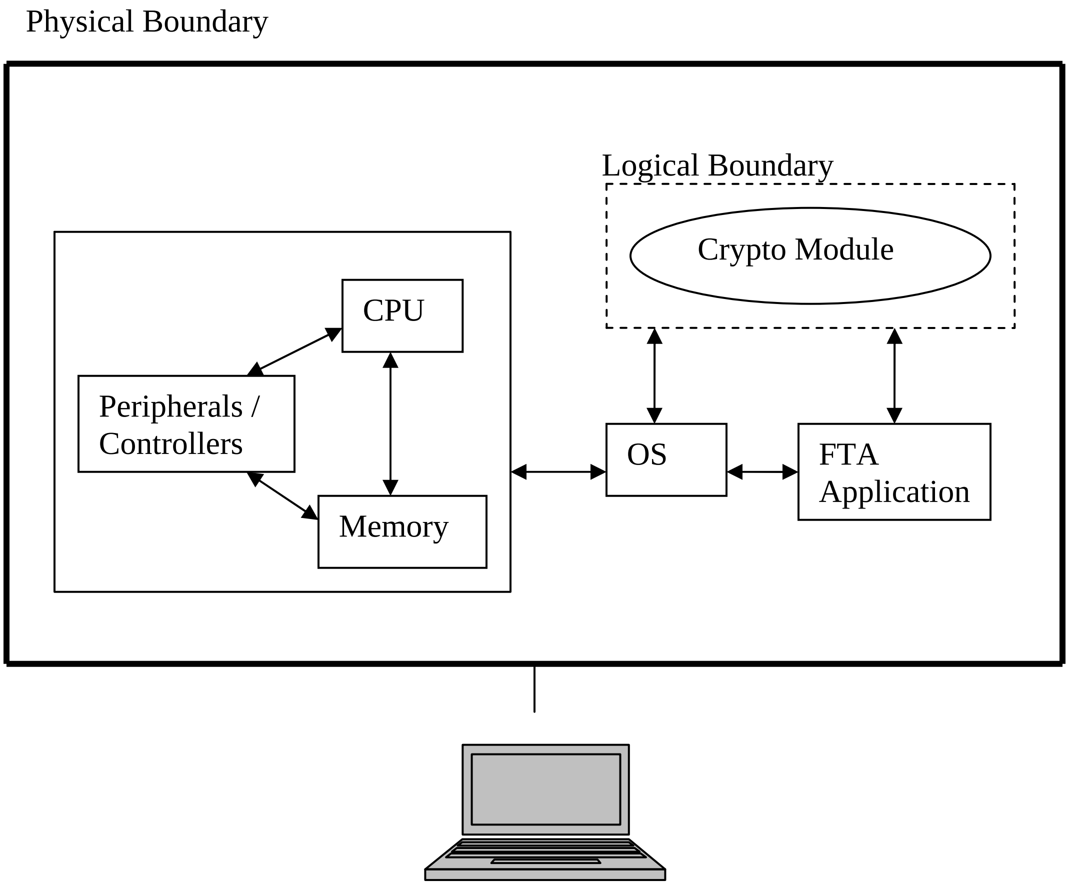
# 1. Module Overview

The Accellion Secure File Transfer Cryptographic Module (SW Version FTALIB_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:

- Rijndael.so
- libmcrypt.so.4.4.7
- libcrypto.so.0.9.8
- libbeecrypt.so.6.4.0
- libphp5.so

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

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



## 2. Security Level

The Accellion Secure File Transfer Cryptographic Module meets the overall requirements applicable to Level 1 security 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 Secure File Transfer Cryptographic Module only supports a FIPS Approved mode of operation; it is placed into FIPS mode when initialized with a valid license key. The user can determine if the cryptographic module is running in FIPS mode via the license page.

### *Approved mode of operation*

The Accellion Secure File Transfer Cryptographic Module supports the following FIPS Approved algorithms:

- AES ECB mode with 128 bit keys for decryption of the file (Cert. #843)
- AES CBC mode with 128 bit keys for decryption of the license (Cert. #844)
- AES CBC mode with 128 and 256 bit keys for encryption and decryption in the TLS (Cert. #845)
- Triple-DES TCBC mode for encryption and decryption in the TLS (Cert #771)
- HMAC-SHA-1 for message authentication (Cert. #639)
- DSA with 1024 bit keys for digital signature verification (Cert. #307)
- SHA-1 for hashing (used with TLS implementation) (Cert. #836)

- SHA-1 for hashing (used with HMAC implementation) (Cert. #1051)
- SHA-1 for hashing (used with DSA implementation) (Cert. #842)

The Accellion Secure File Transfer Cryptographic Module supports the following FIPS allowed algorithms and protocols:

- TLS/SSL 3.1 for secure communications and key establishment
- NDRNG to generate passwords (2 implementations, one for PHP and one for Perl)
- AES key wrap per the AES Key Wrap Specification (Cert. #845, key wrapping; key establishment methodology provides 128 or 256 bits of encryption strength)
- Triple-DES (Cert. #771, key wrapping; key establishment methodology provides 80 bits of encryption strength)
- RSA (key wrapping; key establishment methodology provides 80 bits of encryption strength)

The Accellion Secure File Transfer Cryptographic Module supports the following non-FIPS Approved algorithms which do not support any security relevant operations:

- Blowfish for encryption
- MD5 for hashing

## 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, data output, control input, and status output. The data input interface consists of the input parameters of the shared libraries' functions. The data output interface consists of the output parameters of the shared libraries' functions. The control input interface consists of the actual functions of the shared libraries. The status output interface includes the return values of the functions of the shared libraries.

## 5. Identification and Authentication Policy

The Accellion Secure File Transfer Cryptographic Module supports two distinct operator roles (User, Cryptographic Officer). In compliance with FIPS 140-2 Level 1 standards, the module does not support user 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

***Roles and Services***

**Table 2 – Services Authorized for Roles**

| Role | Authorized Services |
|---|---|
| User: | • Symmetric encryption/decryption: This service provides encryption/decryption functionality for AES and TDES ciphers.<br>• Key wrapping for key transport: This service provides key wrapping functionality for SSL 3.1 or TLS connection.<br>• Digital signature: This service provides functionality to verify digital signatures using DSA cipher.<br>• Keyed Hash (HMAC): This service provides keyed hash functionality using HMAC-SHA1 cipher.<br>• Message Digest (SHS): This service provides functionality to generate message digest using SHA1 cipher. |
| Cryptographic Officer | • Install the module on the host computer system |

Self-Tests: On bootup of the host computer system, automatically runs the self-tests necessary for FIPS 140-2.

Zeroization: All the CSPs can be zeroized through Accellion's Secure File Transfer application.

Show Status: The operator can obtain the current status of the module.

***Definition of Critical Security Parameters (CSPs)***

The following are CSPs contained in the module:

- Key Encryption Key (KEK): This is an AES 128 bit key used for encryption/decryption of AES 128 file decryption key.
- License Key: This is an AES 128 key used to decrypt the license file.
- Accellion TLS Key: This key is used for TLS connections (the factory shipped 1024 bit RSA key is replaced by the customer).

- Customer TLS Key: This key is used for TLS connections, 1024 bit RSA key.
- TLS Session Key: TDES or AES 128/256 bit key used in TLS session.
- File Decryption Key: This is an AES 128 key used to decrypt a file stored on the Secure File Transfer Appliance's hard disk.
- HMAC Key: This key is used by the login API.
- HMAC Software Integrity Key: This key is used to calculate the HMAC-SHA1 digest of the module which is then used in the software integrity test.

### *Definition of Public Keys:*

The following are the public keys contained in the module:

- RSA Public Key: Checks the signature of the license.
- RSA Public Key – TLS: 1024 bit RSA key used in TLS which can be replaced by the customer.
- DSA Public Key for Software Load: A DSA 1024 bit key used to authenticate software loads.

### *Definition of CSPs Modes of Access*

Table 5 defines the relationship between access to CSPs and the different module services. The modes of access shown in the table are defined as follows:

- Use (U): This operation uses the identified CSP.
- Store (S): This operation stores the identified CSP into persistent storage.
- Zeroize (Z): This operation actively overwrites the identified CSP.

**Table 5 – CSP Access Rights within Roles & Services**

| Role | Service | CSPs | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Key Encryption Key | License Key | Accellion/Customer TLS Key | TLS Session Key | File Decryption Key | HMAC Key | HMAC Software Integrity Key |
| User | Symmetric encryption/decryption | U | U | | U | U | | |
| User | Key wrapping for key transport | | | U | U | | | |
| User | Digital signature | | | | | | | |
| User | Keyed Hash (HMAC) | | | | | | U | |
| User | Message Digest (SHS) | | | | | | | |
| Cryptographic Officer | Module Installation | S | S | S | S | S | S | S |
| NA | Self Tests | | | | | | | U |
| NA | Zeroize | Z | Z | Z | Z | Z | Z | |
| NA | Show Status | | | | | | | |

# 7. Operational Environment

The Accellion Secure File Transfer 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 following shared libraries:

- Rijndael.so
- libmcrypt.so.4.4.7
- libcrypto.so.0.9.8
- libbeecrypt.so.6.4.0
- libphp5.so

When a crypto module is implemented in Accellion's SFTA environment, the SFTA application is the user of the cryptographic module. The SFTA application makes the calls to the cryptographic module. Therefore, the SFTA 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 SFTA application is serving multiple clients.

The Accellion Secure File Transfer Cryptographic Module has been tested on Linux OS derived from Red Hat Enterprise Version 5.1

# 8. Security Rules

The Secure File Transfer Appliance 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/SSL3.1 algorithm.
4. The cryptographic module shall perform the following tests:

    A. <u>Power up Self-Tests:</u>

    1. Cryptographic algorithm tests:
        a. AES ECB decryption KAT (for decryption of the file)
        b. AES CBC decryption KATs (for decryption of the license) (2 tests)
        c. AES CBC encryption/decryption KATs (for encryption/decryption in TLS) (2 tests)
        d. TDES encryption/decryption KAT (used with TLS implementation)
        e. HMAC-SHA-1 KAT
        f. DSA verify KAT
        g. SHA-1 KAT (used with TLS implementation)
        h. SHA-1 KAT (used with HMAC implementation)
        i. SHA-1 KAT (used with DSA implementation)
    2. Software Integrity Test – HMAC-SHA-1 used

3. Critical Functions Tests: None

B. Conditional Self-Tests:

1. NDRNG Continuous RNG Test (used with PHP)
2. NDRNG Continuous RNG Test (used with Perl)
3. Software Load Test using DSA with SHA-1

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.
6. Prior to each use, the internal RNG shall be tested using the conditional test specified in FIPS 140-2 §4.9.2.
7. Status information shall not contain CSPs or sensitive data that if misused could lead to a compromise of the module.

# 9. Physical Security Policy

***Physical Security Mechanisms***

The Accellion Secure File Transfer Cryptographic Module is a software module intended for use with Linux OS derived from Red Hat Enterprise Version 5.1; 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 |
| **ALCS** | Accellion Local Cluster Service |
| **API** | Application Program Interface |
| **CO** | Cryptographic Officer |
| **CSP** | Critical Security Parameter (as defined in FIPS 140-2) |
| **DES** | Data Encryption Standard |
| **DSA** | Digital Signature Algorithm |
| **EMC** | Electromagnetic Compatibility |
| **EMI** | Electromagnetic Interference |
| **FIPS** | Federal Information Processing Standard |
| **HMAC** | Keyed-Hash Message Authentication Code |
| **MD5** | Message-Digest Algorithm 5 |
| **NDRNG** | Nondeterministic Random Number Generator |
| **RNG** | Random Number Generator |
| **RPM** | Red Hat Package Manager |
| **RSA** | Rivest, Shamir and Adleman Algorithm |
| **SHA** | Secure Hash Algorithm |
| **SSH** | Secure Shell |
| **SSL** | Secure Sockets Layer |
| **TLS** | Transport Layer Security |
| **USB** | Universal Serial Bus |