Accellion Cryptographic Module Non-Proprietary Security Policy

Document *Version 2.14*

Accellion, Inc.

September 29, 2015

TABLE OF CONTENTS

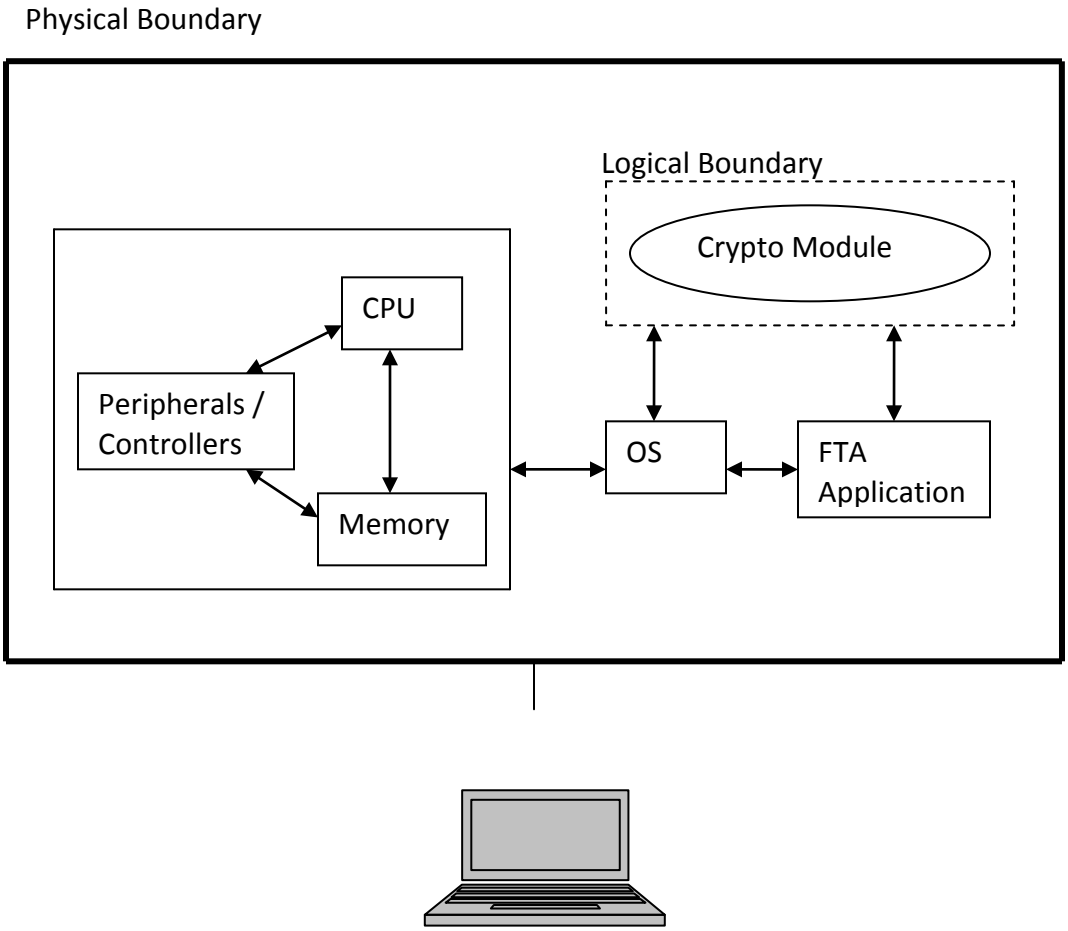
1. Module Overview

The Accellion Cryptographic Module (SW Version FTALIB_4_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:

- PERL AES Library: Rijndael.so [Version: 0.05]
- PHP AES Library: libmcrypt.so.4.4.7 [Version: 2.5.7]
- OpenSSL Object Module: fipscanister.o [Version: 1.0.1]
- Core PHP Library: libphp5.so [Version: 5.2.17p1]
- Utils Binary: fips_utils [Version: 2.1]

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 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 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 and when FIPS Approved/Allowed algorithms are used. It is the operator's responsibility to use only FIPS Approved/Allowed algorithms to stay in a FIPS Approved mode of operation, otherwise the module is placed in a non-Approved mode of operation. The operator can determine if the cryptographic module is running a FIPS certified version by comparing the version on the license page to the version on the FIPS certificate.

3.1 Approved mode of operation

3.1.1 Approved Algorithms

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

PERL AES Library

- AES ECB mode with 128 and 256 bit keys for decryption of the file (Cert. #2317)

PHP AES Library

- AES CBC mode with 128 and 256 bit keys for encryption and decryption. Used for decryption of the license, encryption/decryption of the administrator session, and other general encryption/decryption (Cert. #2318)

OpenSSL Object Module

- AES CBC mode with 128 and 256 bit keys for encryption and decryption in TLS (Cert. #3326)
- AES GCM mode with 128 and 256 bit keys for encryption and decryption in TLS (Cert. #3326)
- Triple-DES TCBC mode using 3-keys for encryption and decryption in TLS (Cert. #1898)
- HMAC-SHA-1, HMAC-SHA-256 and HMAC-SHA-384 for MAC and key derivation in TLS, with minimum key size of 112 bits (Cert. #2117)
- SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512 for hashing and key derivation in TLS (Cert. #2758)
- CVL SP 800-135 TLSv1.0/v1.1/v1.2 KDF for deriving TLS Session Keys (CVL Cert. #482)
- FIPS 186-4 RSA with 1024/2048/3072 bit keys for digital signature verification (Cert. #1707)
- FIPS 186-4 RSA with 2048/3072 bit keys for digital signature generation (Cert. #1707)
- FIPS 186-4 ECDSA Key pair generation (NIST defined P 256/384 curves) (Cert. #655)
- SP 800-56A KAS (NIST defined P 256/384 curves) (CVL Cert. #481)
- SP 800-90A CTR_DRBG using AES 256, security strength is 256 bits (Cert. #772)

Core PHP Library

- HMAC-SHA-1 for API token authentication, with minimum key size of 112 bits (Cert. #2118)
- SHA-1 for hashing and within HMAC (Cert. #2759)

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_ECDHE_RSA_WITH_AES_128_CBC_SHA (with RSA modulus >=2048 bits and NIST defined P 256/384 curves)
  - TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (with RSA modulus >=2048 bits and NIST defined P 256/384 curves)
  - TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA (with RSA modulus >=2048 bits and NIST defined P 256/384 curves)
- 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)
  - TLS_RSA_WITH_AES_128_GCM_SHA256 (with RSA modulus >=2048 bits)
  - TLS_RSA_WITH_AES_256_GCM_SHA384 (with RSA modulus >=2048 bits)
  - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (with RSA modulus >=2048 bits and NIST defined P 256/384 curves)
  - TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (with RSA modulus >=2048 bits and NIST defined P 256/384 curves)
  - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (with RSA modulus >=2048 bits and NIST defined P 256/384 curves)
  - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (with RSA modulus >=2048 bits and NIST defined P 256/384 curves)
- AES key wrap per the AES Key Wrap Specification (Cert. #3326, key wrapping; key establishment methodology provides 128 bits of encryption strength)
- RSA (key wrapping using 2048 or 3072-bit keys; key establishment methodology provides 112 or 128 bits of encryption strength)
- NDRNG to seed the *SP 800-90A* DRBG
- MD5 within TLS key derivation (as allowed per IG D.9)

3.2 Non-approved mode of operation

The module utilizes open source libraries which contain many non-Approved algorithms that are not allowed for use in the Approved mode. 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 Object Module

- AES (not tested; non-compliant):
  - CBC mode: 192
  - CBC-HMAC-SHA-1
  - CFB, CFB1, CFB8, CTR, ECB, OFB modes
  - XTS

    - CMAC
- DRBG (not tested; non-compliant): HASH, HMAC, Dual EC, CTR with 128/192 or no derivation function
- DSA (not tested; non-compliant)
- ECDSA (not tested; non-compliant):
    - Signature Generation
    - Signature Verification
- HMAC (not tested; non-compliant): HMAC-SHA-224 and HMAC-SHA-512
- RNG (not tested; non-compliant): ANSI x931
- RSA (<112 bits of strength; 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 (not tested; 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

Core PHP Library

- 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 (not tested; non-compliant)
  - SHA-384 (not tested; 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 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

6.1 Roles and Services

Table 2 – Services Authorized for Roles

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.
	X		X		AES key wrapping for key transport	This service provides key wrapping functionality for file decryption key
	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.

Crypto Officer Role	User Role	No Role Required	FIPS Approved Mode	Non-FIPS Approved Mode	Service Name	Service Description
	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-384.
	X		X		Message Digest (SHS)	This service provides functionality to generate message digests using SHA-1 or SHA-224 or SHA-256 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		Key Agreement	This service performs SP 800-56A key agreement using P-256 and P-384 curves.
	X		X		Misc/Helper Service	This service helps in performing the services mentioned above
	X			X	Non-Approved APIs	This is a catch-all service for all other APIs that are present in the module 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.
	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/FTA_FIPS_Service_API_mapping_4_0_1.pdf.

6.2 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 the File Decryption Key.
- License Key: This is an AES 256 bit key used to decrypt the license file.
- Session Key: This is an AES 256 bit key used to encrypt/decrypt administrator session.
- 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 or 3072 bit private key used for TLS connections. This key is loaded by the customer over network port on GPC via TLS 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.
- File Decryption Key: This is an AES 128 or 256 bit key used to decrypt a file stored on the Secure File Transfer Appliance's hard disk. Can be updated over network port on GPC via TLS.
- HMAC Key: This >= 112 bit key is used by the login API through an API based authentication. This is one of the available authentication modes. Can be updated over network port on GPC via TLS.
- HMAC Software Integrity Key: This >= 112 bit key is used to calculate the HMAC-SHA1 digest of the module which is then used in the software integrity test.
- Entropy Input to DRBG: This is the seed to the DRBG.
- DRBG State (V and Key): These are the state variables for the DRBG.
- ECDH Key: This is an EC DH (NIST defined P 256/384 curves) private key agreement key.
- ECDH Shared Secret: This is shared secret derived at the end of the ECDH key agreement process.

Definition of Public Keys:

The following are the public keys contained in the module:

- RSA Public Key: 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 or 3072 bit public key used in TLS which can be replaced by the customer.
- Third Party RSA Public Keys – TLS: This is a RSA 2048 or 3072 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 3072 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).
- ECDH Public Key: This is EC DH (NIST defined P 256/384 curves) public key agreement key.
- Third Party ECDH Public Key: This is a third party EC DH (NIST defined P 256/384 curves) public key agreement key

6.3 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 actively overwrites the identified CSP.

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

Services	Key Encryption Key	License Key	Session Key	Accellion TLS Key	Customer TLS Key	TLS Session Keys	File Decryption Key	HMAC Key	HMAC Software Integrity Key	Entropy Input to DRBG	DRBG State (V and Key)	ECDH Key	ECDH Shared Secret	RSA Public Key	RSA Public Key - TLS	Third Party RSA Public Keys – TLS	Third Party RSA Public Keys – SAML	ECDH_Public_Key	Third Party ECDH_Public_Key
Symmetric encryption/decryption	U	U	U		I	U	UI O	I							I	I			
AES key wrapping for key transport	U						IO												
RSA Key wrapping/unwrapping for key transport				U	U	U									U	U			
Digital Signature Verification														U			UI		
Digital Signature Generation				U	U														
Keyed Hash (HMAC)						G		U											
Message Digest (SHS)																			
TLS Session Establishment										G	G		U						
Key Agreement												GU	G					GU O	UI
Misc/Helper Service																			
Non-Approved APIs																			
Module Installation	U	U	U	U	U	U		U	U	U	U	U	U	U	U			U	U
Self-Tests									U										
Zeroization	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z						
Show Status																			

7. Operational Environment

The Accellion 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 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 Cryptographic Module has been tested on Red Hat Enterprise Linux 5 on VMware ESXi 5.1.0 running on a Dell Inc. PowerEdge R320.

8. Security Rules

The Accellion 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:

      PERL AES Library

         a. AES ECB decryption KATs (for decryption of the file) (2 tests for both 128 bit and 256 bit)

      PHP AES Library

         b. AES CBC encryption/decryption KATs (for decryption of the license, encryption/decryption of the administrator session, and other general encryption/decryption) (2 tests)

      OpenSSL Object Module

         c. AES CBC 128/256 bit encryption KAT and decryption KAT
         d. AES GCM 128/256 bit encryption KAT and decryption KAT

e. Triple-DES TCBC 3-key encryption KAT and decryption KAT
f. HMAC-SHA-1,HMAC-SHA-256 and HMAC-SHA-384 KATs
g. SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512 KATs
h. SP 800-135 TLS KDF KAT
i. FIPS 186-4 RSA 2048 bit signature verification KAT
j. FIPS 186-4 RSA 2048 bit signature generation KAT
k. DRBG KAT
l. RSA 2048 key wrapping and unwrapping KAT
m. SP 800-56A KAS KATs per IG 9.6 (P 256/384 curves)

Core PHP Library

n. HMAC-SHA-1 KAT
o. SHA-1 KAT

Utils Binary

p. AES Key Wrap KAT
q. AES Key Unwrap KAT

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

B. Conditional Self-Tests:

1. NDRNG Continuous Test
2. DRBG Continuous Test
3. DRBG Health Checks (SP 800-90A Section 11.3)
4. ECDSA Pairwise Consistency Test
5. SP 800-56A KAS Conditional Tests per IG 9.6

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 [test type]. Going to 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. If the ECDSA Pairwise Consistency Test fails, the module provides the following error: "ECDA pairwise consistency test failed, going to fips error state". If the SP 800-56A KAS Conditional Test fails, the module provides the following error: "EC conditional test of key assurance failed, going to fips error state". In either case, the module is then uninstalled and the GPC is shutdown.

8. Status information shall not contain CSPs or sensitive data that if misused could lead to a compromise of the module.
9. Third party clients and applications using this module shall restrict their TLS ciphers suites to those listed in Section 3 of this document.
10. Third party clients using this module shall restrict their TLS RSA modulus to be 2048 or 3072 bits.
11. Third party clients using this module shall restrict their EC key sizes to NIST defined P 256/384 curves
12. Third party clients using this module shall restrict their HMAC keys to be >= 112 bits.
13. Customer TLS Keys used to replace the Accellion TLS Key shall be restricted to RSA with a modulus 2048 or 3072 bits.
14. The GCM IV is generated internally to the module as specified by SP 800-38D Section 8.2.1. The fixed field allows for 2^32 different names. In the case of a power loss, the GCM encryption/decryption session is terminated and the GCM key/IV would need to be regenerated. This meets IG A.5, #B.3.

9. Physical Security Policy

The Accellion Cryptographic Module is a software module intended for use with Red Hat Enterprise Linux 5; 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
ECDSA	Elliptic Curve Digital Signature Algorithm
ECDH	Elliptic Curve Diffie-Hellman
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FIPS	Federal Information Processing Standard
HMAC	Keyed-Hash Message Authentication Code
KAS	Key Agreement Scheme
KDF	Key Derivation Function
MD5	Message-Digest Algorithm 5

RNG	Random Number Generator
RSA	Rivest, Shamir and Adleman Algorithm
SFTA	Secure File Transfer Appliance
SHA	Secure Hash Algorithm
SSL	Secure Sockets Layer
Triple-DES	Triple-Data Encryption Standard (Algorithm)
TLS	Transport Layer Security