Axway Inc.

Axway Security Kernel

(Software Version 3.0.2)

FIPS 140-2
Non-Proprietary Security Policy

Level 1 Validation

Document Version 2.7

Prepared By:

Axway Inc.
6811 East Mayo Blvd, Suite 400
Phoenix, Arizona, 85054
Phone: (480) 627-1800
http://www.axway.com

Revision History

Version	Modification Date	Modified By	Description of Changes
1.0	2013-01-10	Anubhav Soni	Initial draft.
1.0.1	2013-02-15	Anubhav Soni Prabhakar Mangam	Updated with compliance for OSes and Axway logos etc.
1.1	2013-03-08	Marc Ireland Prabhakar Mangam	Reviewed and includes comments from InfoGard (Marc Ireland).
1.2	2013-03-28	Marc Ireland Prabhakar Mangam Anubhav Soni and Hristo Todorov	Reviewed and updated document based on broader review.
1.3	2013-04-16	Anubhav Soni	Updated known answer test section
1.4	2013-04-29	Anubhav Soni	Updated to include non-approved services
1.5	2013-05-01	Anubhav Soni, Marc Ireland, Prabhakar Mangam	Reviewed and updated with comments from InfoGard (Marc Ireland)
1.6	2013-05-02	Mark S, Anubhav S, Enamul H, Prabhakar M, Hristo T	Axway internal reviewed
1.7	2013-05-06	Marc Ireland Anubhav Soni Prabhakar Mangam	Reviewed & updated Services and non-approved services tables
1.8	2013-05-23	Luis Garcia, Prabhakar Mangam, Anubhav Soni	Updated with comments from InfoGard review.
1.9	2014-03-11	Prabhakar Mangam	Updated with comments from InfoGard review
2.0	2014-08-07	Marc Ireland	Updated with comments from InfoGard review
2.1	2015-03-09	Enamul Haque	Updated kernel version
2.2	2016-05-16	Bill Shine	Updated with comments from InfoGard review. Updated kernel version
2.3	2016-08-30	Bill Shine	Updated with comments from InfoGard review.
2.4	2017-04-19	Bill Shine	Updated after common criteria changes.
2.5	2017-09-21	Bill Shine	Updated with comments from UL review.
2.6	2017-12-20	Bill Shine	Updated after CMVP Comments
2.7	2018-01-11	Bill Shine	Updated after CMVP Comments

Table of Contents

Table of Figures

Table of Tables

INTRODUCTION

1.1 Purpose

This document is a non-proprietary Cryptographic Module Security Policy for the Axway Security Kernel from Axway Inc.(Axway). This Security Policy describes how the Axway Security Kernel meets the security requirements of FIPS 140-2 and how to run the module in a secure FIPS 140-2 mode. This policy was prepared as part of the Level 1 FIPS 140-2 validation of the module.

FIPS 140-2 (Federal Information Processing Standards Publication 140-2 – *Security Requirements for Cryptographic Modules*) details the U.S. and Canadian Government requirements for cryptographic modules. More information about the FIPS 140-2 standard and validation program is available on the National Institute of Standards and Technology (NIST) Cryptographic Module Validation Program (CMVP) website at: http://csrc.nist.gov/groups/STM/cmvp/index.html.

In this document, the Axway Security Kernel is referred to as the kernel or the module. The client application represents the software program linked with the cryptographic libraries provided by the Axway Security Kernel. The Validation Authority Suite and MailGate are currently the only applications making use of the Axway Security Kernel. However, it is expected that a range of products developed by Axway Inc., will be supported by the Axway Security Kernel in the future.

1.2 References

This document deals only with the operations and capabilities of the module in the technical terms of a FIPS 140-2 cryptographic module security policy. More information is available on the module from the following sources:

- The Axway website (http://www.axway.com) contains information on the full line of products from Axway.
- The CMVP website (http://csrc.nist.gov/groups/STM/cmvp/index.html) contains contact information for answers to technical or sales-related questions for the module.

1.3 Document Organization

The Security Policy document is one document in a FIPS 140-2 submission package. In addition to this document, the Submission Package contains:

- Vendor Evidence document
- Finite State Machine
- Other supporting documentation as additional references

This Security Policy and the other validation submission documentation were reviewed by UL Verification Services Inc. under contract to Axway. With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 Validation Documentation is proprietary to Axway and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact Axway.

2 Axway Security Kernel

2.1 Overview

The Axway Security Kernel (version 3.0.2) is a multi-chip standalone software cryptographic module implemented as two dynamic link libraries (DLLs) on Windows or two Shared Objects (SOs) on Linux and SunOS. The Axway Security Kernel is a user space shared library. It does not modify or become part of the Operating System (OS) kernel. Table 1 gives the operating systems and corresponding file names of the kernel.

The module has been tested and validated on Microsoft Windows 2012 (64 bit) on a 64 bit Intel Xeon E5-2620 processor, RHEL 6.3 (64 bit) on a 64 bit Intel Xeon E5-2620 processor, and Solaris 10 on a 64 bit Sun UltraSparc T1 processor (each configured for single user mode). Compliance is maintained on following platforms including (but not limited to):

- Windows 7 32 and 64 bit
- Windows 8 32 and 64 bit
- Windows 10 64 bit.
- Windows 2008 32 and 64 bit
- Windows 2008 R2 64 bit
- Windows 2012 64bit
- RHEL 6.X 32 and 64 bit
- RHEL 7.X 64 bit
- Solaris 10/Zones Solaris 10 32 and 64 bit
- Solaris 11/Zones Solaris 11 32 and 64 bit

The platforms supported by the module are binary compatible with the platforms used in the FIPS validation. The CMVP makes no statement as to the correct operation of the module on the operational environments for which testing was not performed.

Table 1 – Binary Forms of the Kernel

Operating Systems	Binary File Names
Windows 2008 32 and 64bit Windows 7 32 and 64 bit Windows 2008 R2 64 bit Windows 10 64 bit. Windows 2012 64bit Windows 8 32 and 64 bit	libeay32-TMWD.dll ssleay32-TMWD.dll
Linux kernel 2.6.13 and later versions RHEL 6.X 32 and 64 bit RHEL 7.X 64 bit	libcrypto-TMWD.so.1.0.2 libssl-TMWD.so. 1.0.2
Solaris 10/ Zones Solaris 10 32 and 64 bit Solaris 11/Zones Solaris 11 32 and 64 bit	libcrypto-TMWD.so.1.0.2 libssl-TMWD.so.1.0.2

The kernel is built upon a custom version of OpenSSL 1.0.2k. As a cryptographic module, the Axway Security Kernel presents an identical application programming interface (API) to several products by Axway Inc., including Axway Validation Authority Suite and MailGate.

The cryptographic capabilities of Validation Authority and MailGate are provided by the Axway Security Kernel. Validation Authority offers a comprehensive, scalable, and reliable framework for real-time validation of digital certifications for the Public Key Infrastructure (PKI). Governments and businesses worldwide rely on PKI and digital certificates issued by certificate authorities (CAs) to secure information transmissions on the internet. Not all certificates are valid. Some may be fake, expired, or revoked. Therefore, it is of vital importance to make sure that only valid certificates are trusted. Validation Authority provides a variety of PKI and certificate management functionalities such as real-time validation of digital certificates issued by any CA. The MailGate platform is

capable of performing tasks such as email encryption, secure file collaboration, network defense, content filtering, and data protection.

The Axway Security Kernel supports the following FIPS-approved algorithms:

Table 2 – FIPS Approved Algorithms

Algorithm Certs	Description
AES (Cert. #4466)	Encrypt/Decrypt - ECB, CBC, CFB128, OFB and CTR modes; 128,192 and 256 bits
CKG	Vendor Affirmed
CVL (Cert. #1177)	KDF135, TLS SP800-135 TLS 1.0/1.1 TLS1.2 (SHA 256, 384 and 512) No parts of the TLS protocol, other than the KDF, have been tested by the CAVP and CMVP.
CVL (Cert. #1178)	ECDSA SigGen Component Curves tested: P-224, P-256 and P-384
CVL (Cert. #1179)	ECC CDH Primitive Curves tested: P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409 and B-571; ECC SCHEMES: EphemUnified: Curves tested: P-256, K-283, B-283
DRBG (Cert. #1449)	Hash_Based DRBG: [Prediction Resistance Tested: Enabled and Not Enabled (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512) (SHS Val#3678)] HMAC_Based DRBG: [Prediction Resistance Tested: Enabled and Not Enabled (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512) (HMAC Val#2964)] CTR_DRBG: [Prediction Resistance Tested: Enabled and Not Enabled; BlockCipher_Use_df: (AES-128, AES-192, AES-256) (AES Val#4466)] BlockCipher_No_df: (AES-128, AES-192, AES-256) (AES Val#4466)]
ECDSA (Cert. #1089)	FIPS 186-4 ECDSA using SHA-1 or SHA-2: PKG: CURVES (P-224 P-256 P-384) PKV: CURVES (P-192 P-224 P-256 P-384 B-163 B-233) SigGen: CURVES (P-224: (SHA-1, 224, 256, 384, 512) K-233: (SHA-1, 224, 256, 384, 512) B-233: (SHA-1, 224, 256, 384, 512) SIG (gen) with SHA-1 affirmed for use with protocols only. SigVer: CURVES (P-192: (SHA-1, 224, 256, 384, 512), P-224: (SHA-1, 224, 256, 384, 512), P-256: (SHA-1, 224, 256, 384, 512), P-384: (SHA-1, 224, 256, 384, 512), K-163: (SHA-1, 224, 256, 384, 512), K-233: (SHA-1, 224, 256, 384, 512), B-163: (SHA-1, 224, 256, 384, 512), B-233: (SHA-1, 224, 256, 384, 512))
HMAC (Cert. #2964)	HMAC-SHA1 and HMAC-SHA2 with: 20,28,32,48 and 64 byte MACs with KS=BS,KS<BS, KS>BS

Algorithm Certs	Description
RSA (Cert. #2442)	FIPS 186-2 RSA: Signature Generation PKCS1.5: Modulus lengths: 4096 (bits) SHAs: SHA-224, SHA-256, SHA-384, SHA-512 Signature Verification PKCS1.5: Modulus lengths: 1024, 2048, 4096 (bits) SHAs: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 FIPS 186-4 RSA Key Generation: Public Key Exponent: Random Probable Random Primes: Mod lengths: 2048, 3072 (bits) Primality Tests: C.2, C.3 Signature Generation PKCS1.5: Mod 2048 SHA: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 Mod 3072 SHA: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 Signature Verification PKCS1.5: Mod 1024 SHA: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 Mod 2048 SHA: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 Mod 3072 SHA: SHA-1, SHA-224, SHA-256, SHA-384, SHA-512
SHA (Cert. # 3678)	SHA-1 SHA-2: SHA-224, SHA-256, SHA-384 and SHA-512
Triple-DES (Cert. #2397)	Triple-DES Encrypt/Decrypt: 168 bits (for 3-key), in TECB, TCBC, TCFB64, and TOFB modes The user is responsible for ensuring the module is not used to perform more than 2^28 encryptions with the same Triple-DES key.

The Axway Security Kernel supports the following non-Approved but allowed cryptographic algorithms:

- EC Diffie-Hellman using P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, and B-571 curves (key agreement; key establishment methodology provides between 112 and 256 bits of encryption strength)
- MD5 for use within TLS only.
- RSA 2048-bit - for key transport in TLS sessions (key wrapping; key establishment methodology provides 112 bits of encryption strength)

Apart from the tested algorithms, the Axway Security Kernel also provides the following non-approved cryptographic algorithms:

- Algorithms Disallowed as of January 1, 2014 per the NIST SP 800-131A algorithm transitions:
    - FIPS 186-2 ECDSA using SHA-1: Signature generation (P-192, P-224, K-163, K-233, B-163, and B-233 curves), Key generation (P-192, K-163, and B-163 curves)
    - FIPS 186-2 RSA: PKCS #1 V1.5 Signature generation (1024-bit using SHA-1 and SHA-2;
    - EC Diffie-Hellman using P-192, K-163 or B-163 curves (key agreement; key establishment methodology provides less than 112 bits of encryption strength; non-compliant)
    - RSA Encrypt/Decrypt 1024-bit (key wrapping; key establishment methodology provides 80 bits of encryption strength; non-compliant)

- AES-IGE
- Diffie-Hellman
- Two-Key Triple-DES
- Blowfish
- CMAC
- CMS
- Camellia
- Cast
- DES
- DSA
- des_old
- DTLS1
- IDEA
- KRB5_asn
- KSSL
- MD4
- MD5
- MDC2
- RC2
- RC4
- PKCS12
- PKCS7
- RIPEMD
- Seed
- SRP
- Whirlpool

The Axway Security Kernel supports a FIPS-Approved mode of operation. Please see Section 2.3 below for a list of Approved and non-Approved services.

The Axway Security Kernel is validated at FIPS 140-2 section levels shown in Table 3. Note that in Table 3, N/A indicates “Not Applicable”.

Table 3 – Security Level Per FIPS 140-2 Section

Section	Section Title	Level
1	Cryptographic Module Specification	1
2	Cryptographic Module Ports and Interfaces	1
3	Roles, Services, and Authentication	1
4	Finite State Model	1
5	Physical Security	N/A
6	Operational Environment	1
7	Cryptographic Key Management	1
8	EMI/EMC (Electromagnetic Interference/Compatibility)	1
9	Self-Tests	1
10	Design Assurance	1
11	Mitigation of Other Attacks	N/A

2.2 Module Interfaces

The Axway Security Kernel is a software module that meets overall level 1 of FIPS 140-2 requirements. The logical cryptographic boundary of the module consists of the Axway Security Kernel running on different OSs. The module is composed of two binary files cross-compiled on the OS. Table 1 summarizes the platforms and the binary files.

Figure 1 shows the logical cryptographic boundary of the kernel. The module provides a set of cryptographic services (API calls) in areas such as Transport Layer Security (TLS) and SP800-90A DRBG.

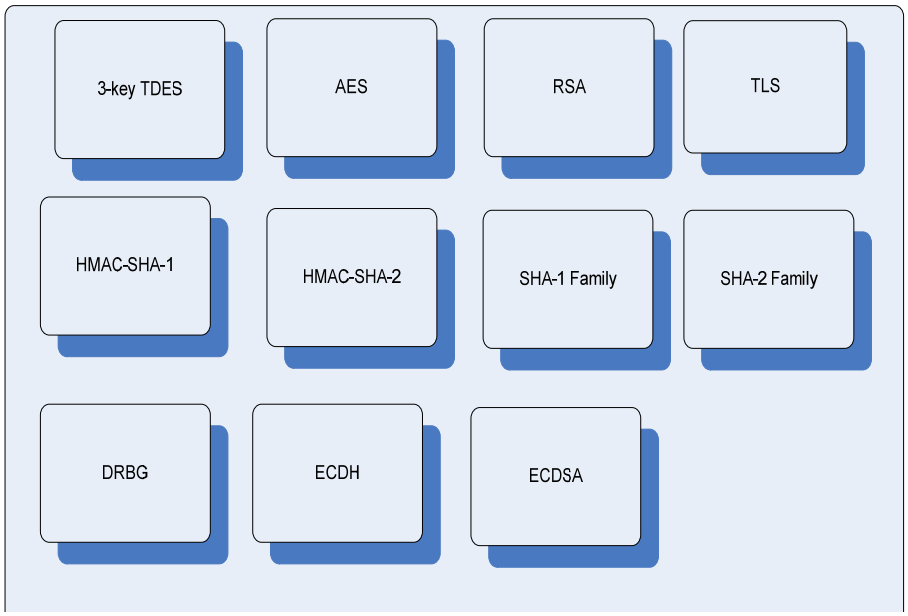


Figure 1 – Logical Cryptographic Boundary

The kernel's interactions with surrounding components, including the Central Processing Unit (CPU), hard-disk, memory, client application, and the OS are demonstrated in Figure 2.

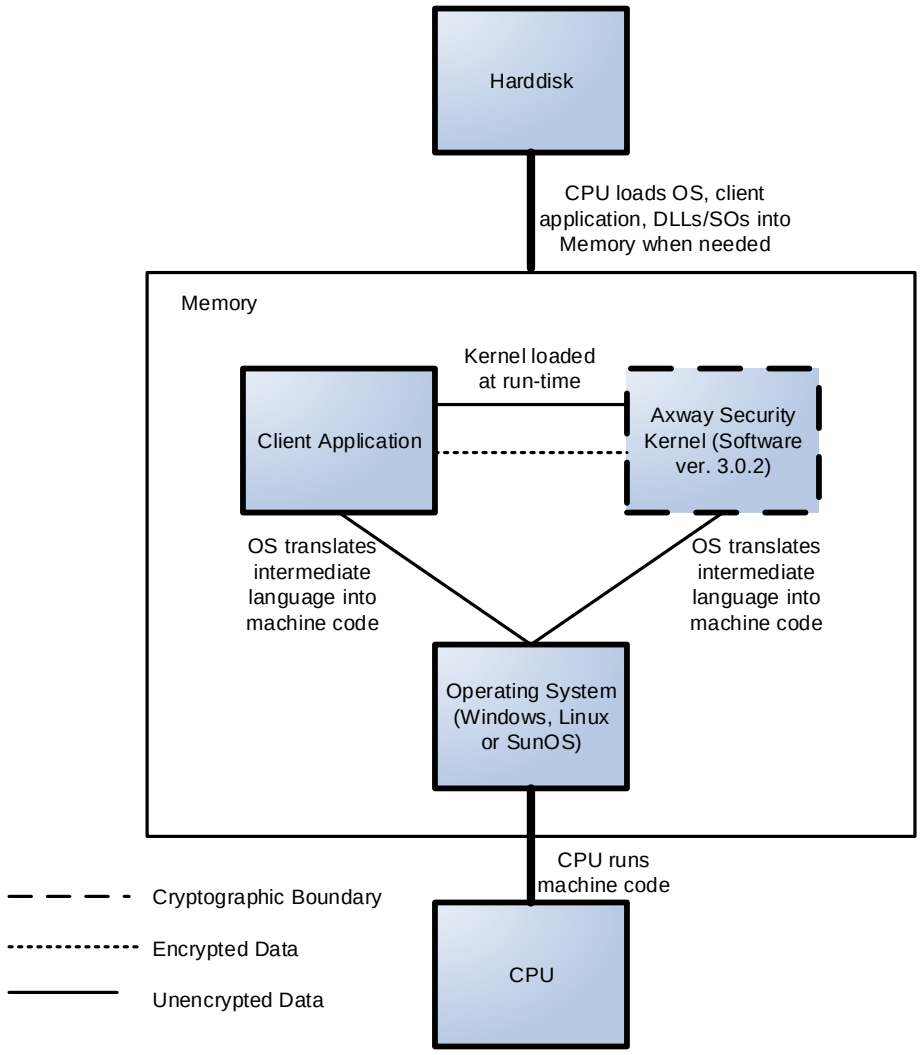
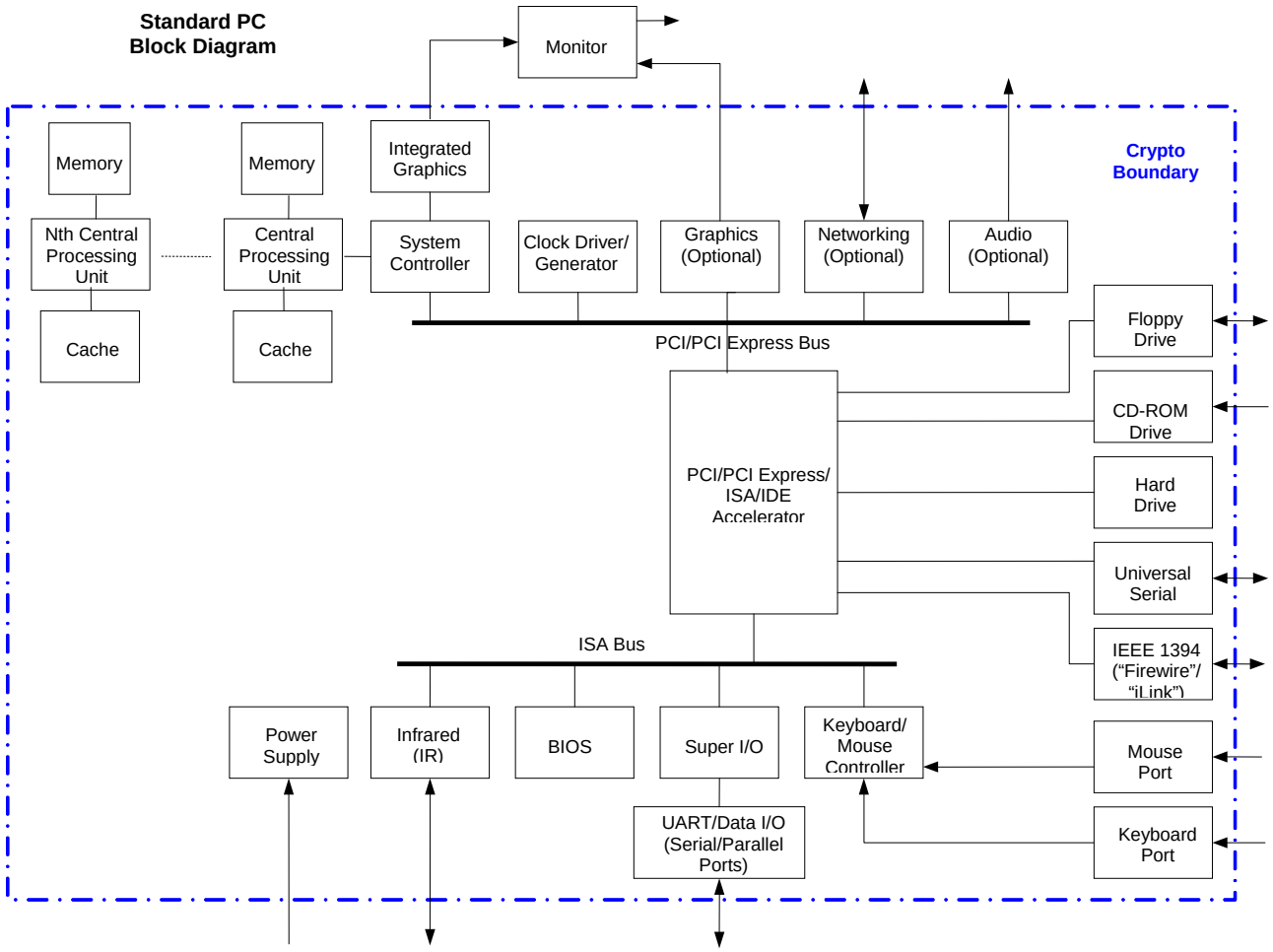


Figure 2 – Logical Cryptographic Boundary and Interactions with Surrounding Components

In addition to the binaries, the physical device consists of the integrated circuits of the motherboard, the CPU, Random Access Memory (RAM), Read-Only Memory (ROM), computer case, keyboard, mouse, video interfaces, expansion cards, and other hardware components included in the PC such as hard disk, floppy disk, Compact Disc ROM (CD-ROM) drive, power supply, and fans. The physical cryptographic boundary of the module is the hard opaque metal and plastic enclosure of the PC, server, or mainframe running the module. The block diagram for a standard PC is shown in Figure 3. The physical block diagram for a server or a mainframe is similar to Figure 3. Note that in this figure, I/O means Input/Output, BIOS stands for Basic Input/Output System, PCI stands for Peripheral Component Interconnect, ISA stands for Instruction Set Architecture, and IDE represents Integrated Drive Electronics.

Figure 3 – Standard PC Physical Block Diagram



All of these physical interfaces are separated into logical interfaces defined by FIPS 140-2, as described in Table 4.

Table 4 – FIPS 140-2 Logical Interfaces

Logical Interface	Axway Security Kernel Port/Interface	Module Mapping
Data Input Interface	Keyboard, mouse, CD-ROM, floppy disk, and serial/Universal Serial Bus (USB)/parallel/network ports	Arguments for API calls that contain data to be used or processed by the kernel
Data Output Interface	Hard Disk, floppy disk, monitor, and serial/USB/parallel/network ports	Arguments for API calls that contain kernel response data to be used or processed by the caller
Control Input Interface	Keyboard, CD-ROM, floppy disk, mouse, and serial/USB/parallel/network port	API function calls
Status Output Interface	Hard Disk, floppy disk, monitor, and serial/USB/parallel/network ports	Arguments for API calls, function return value, error message
Power Interface	Power Interface	Not Applicable

2.3 Roles and Services

The operators of the module can assume two roles as required by FIPS 140-2: a Crypto Officer role and a User role. The operator of the module assumes either of the roles based on the operations performed without any authentication. Table 5 gives a brief description of the roles and their privileges.

Table 5 – Module Roles and Privileges

Role Name	Role Privileges
Crypto Officer	1. Installing/uninstalling the kernel on the specific platform. 2. Initiating power-up self-tests on demand.
User	Performing cryptographic services by making API calls to the kernel.

The following subsections detail both of the roles and their responsibilities.

2.3.1 Crypto Officer Role

The Crypto Officer role has the ability to install and uninstall the module and run power-up self-tests on demand. Descriptions of the services available to the Crypto Officer role are provided in Table 6, where CSP refers to Critical Security Parameter.

Table 6 – Crypto Officer Services

Service	Description	Input	Output	CSP and Type of Access
Install kernel	Installs and configures the kernel	Command	Success or failure	None
Uninstall kernel	Remove the kernel from the OS	Command	Success or failure	None
Initiate power-up self-tests	Power-up self-tests as described in Section 2.7	Command	Success or failure	None

2.3.2 User Role

The User role accesses the module's services that include encryption, decryption, and authentication functionality. Descriptions of the services available to the User role are provided in Table 7 – Approved User Services

.

Table 7 – Approved User Services

Service	Role	Description	CSPs and Access
Random Number Generation	User	Used for random number and symmetric key generation.	V, C, Key, Entropy input Access:read/write/execute
Asymmetric Key Generation	User	Used to generate ECDSA, ECDH, and RSA keys.	RSA and ECDSA private key Access:read/write/execute
Symmetric Encrypt/Decrypt	User	Used to encrypt or decrypt data	AES and TDES key Access:read/write/execute
Message Digest	User	Used to generate a sha-1 or sha-2 message digest	None
Keyed Hash	User	Used to generate or verify data integrity with HMAC.	HMAC key Access:read/write/execute
Key Agreement	User	Used to perform key agreement (i.e., EC-DH, RSA Key Transport, TLS) on behalf of the calling process.	AES, TDES key, RSA, private key, HMAC key, TLS Premaster Secret, TLS Master Secret, EC Diffie-Hellman Private Components Access:read/write/execute
Digital Signature	User	Used to generate or verify RSA or ECDSA digital signatures.	RSA and ECDSA private key Access:read/write/execute
Show Status	User	Show status of a service (function call)	None

2.3.3 Non-Approved Services

The following are the non-Approved services available for the User role. These provide non-Approved services that include encryption, decryption, and authentication functionality. Descriptions of the Approved services available to the User role are provided above in Table 7 – Approved User Services

Table 8 – Non-Approved Services

Service	Role	Algorithms used	Description
AES-IGE	User	AES-IGE	Used to provide AES-IGE encryption/decryption services to callers.
blowfish	User	blowfish	Used to provide blowfish encryption/decryption services to callers.
camellia	User	camelia	Used to provide camellia encryption/decryption services to callers.
cast	User	cast	Used to provide cast encryption/decryption services to callers.
cmac	User	cmac	Used to generate or verify data integrity used by Triple-DES.
cms	User	cms	Used to provide certificate management services to callers.
des	User	DES_OLD, DES, two key triple DES	Used to provide DES encryption/decryption services to callers.
Diffie-Hellman	User	Diffie-Hellman	Used to provide Diffie Hellman key exchange.
dsa	User	Dsa	Used to provide DSA digital signature signing and signature verification services to callers.
DTLS1	User	DTLS1	Used to provide DTLS1 transport
ECDH using P-192, K-163, or B-163 curves	User	ecdh	Used to provide key key agreement, key establishment, using P-192, K-163 or B-163 curves .

FIPS 186-2 ECDSA	User	ecdsa	ECDSA using SHA-1: Signature generation (P-192, P-224, K-163, K-233, B-163, and B-233 curves), Key generation (P-192, K-163, and B-163 curves)
idea	User	idea	Provide data encryption / decryption using IDEA to the caller.
KRB5_asn	User	KRB5_asn	Provide asn.1 parsing for Kerebos SSL
KSSL	User	kssl	Provide Kerebos SSL service
Md4	User	Md4	Provide md4 digest generation to the caller
Md5	User	Md5	Provide md5 digest generation to the caller
Mdc2	User	Mdc2	Provide mdc2 digest generation to the caller
ocsp	User	N/A	Provide OCSP protocol methods for handling OCSP requests and responses.
PKCS12	User	N/A	Provide PKCS12 certificate handling to the caller.
PKCS7	User	N/A	Provide PKCS7 certificate handling to the caller.
Rc2	User	Rc2	Provide rc2 encryption/decryption to the caller.
Rc4	User	Rc4	Provide rc4 encryption/decryption to the caller.
ripemd	User	ripemd	Provide ripemd digest generation to the caller.
RSA Encrypt/Decrypt using 1024 bit keys.	User	rsa	Provide rsa encryption/decryption with 1024 bit keys.
FIPS 186-2 RSA	User	rsa	PKCS #1 V1.5 Signature generation (1024-bit using SHA-1 and SHA-2)
seed	User	seed	Provide seed encryption/decryption to the caller

srp	User	srp	Provide srp password authentication to the caller
ts	User	N/A	Provide time service authority functions to the caller
ui	User	N/A	Provide terminal I/O functions to the caller
whirlpool	User	whirlpool	Provide a whirlpool digest to the caller
X509	User	N/A	Provide miscellaneous X509 functions to the caller
X509v3	User	N/A	Provide miscellaneous X509v3 functions to the caller.

2.4 Physical Security

The Axway Security Kernel is a multi-chip standalone module. The physical security requirements do not apply to this module, since it is purely a software module and does not implement any physical security mechanisms.

Although the module consists entirely of software, the FIPS 140-2 evaluated platform is a standard PC, a server, or a mainframe, which has been tested for and meets applicable Federal Communication Commission (FCC) EMI and EMC requirements for business use as defined in Subpart B of FCC Part 15.

2.5 Operational Environment

The module runs on the general purpose Microsoft Windows, Linux and SunOS operating systems. See Column 1 of Table 1 for the OSs that are supported by the module. The OS being used must be configured for single user mode per NIST CMVP guidance. The module was tested and validated on Windows 2012 (64 bit), RHEL 6.3 (64 bit), and Solaris 10 (64 bit). Single user mode configuration instructions for various OSs can be found in Section 3.1.1 of this document.

2.6 Cryptographic Key Management

The module supports the following CSPs:

Table 9 – List of Cryptographic Keys, Cryptographic Key Components, and CSPs

Key	Key Type	Generation	Input/Output	Storage	Zeroization	Use
TDES Keys	Symmetric key 128 or 192 bits	1. Generated internally using a NIST [SP 800-90A] DRBG. 2. Established via key agreement	N/A	Plaintext in volatile memory only	Zeroized after use	Encrypt plaintext/ Decrypt ciphertext
AES Key	Symmetric key 128, 192, or 256 bits.	1. Generated using a NIST [SP 800-90A] DRBG. 2. Established via key agreement	N/A	Plaintext in volatile memory only	Zeroized after use	Encrypt plaintext/ Decrypt ciphertext
RSA Private Key	Private key 1024, 2048, or 3072 bits.	Generated internally using a NIST [SP 800-90A] DRBG.	N/A	Plaintext in volatile memory only	Zeroized upon use	Decrypt ciphertext/ Sign messages (usually hash values)
RSA Public Key	Public key 1024, 2048, or 3072 bits.	Generated internally using a NIST [SP 800-90A] DRBG.	N/A	Plaintext in volatile memory only	Zeroized upon use	Encrypt plaintext/ Verify signatures
ECDSA Private Key	Private key 160, 224, 256, 384, or 512 bits	Generated internally using a NIST [SP800-90A] DRBG.	N/A	Plaintext in volatile memory only	Zeroized upon use	Sign messages (usually hash values)
ECDSA Public Key	Public key 160, 224, 256, 384, or 512 bits	Generated internally using a NIST [SP 800-90A] DRBG.	N/A	Plaintext in volatile memory only	Zeroized upon use	Verify signatures
EC Diffie-Hellman Public Keys for P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, and B-571 curves	Public key 160, 224, 256, 384, or 512 bits	Generated internally using a NIST [SP 800-90A] DRBG.	N/A	Plaintext in volatile memory	Zeroized upon use	Establish symmetric keys

Key	Key Type	Generation	Input/Output	Storage	Zeroization	Use
EC Diffie-Hellman Private Keys for P-224, P-256, P-384, P-521, K-233, K-283, K-409, K-571, B-233, B-283, B-409, and B-571 curves	Private key 160, 224, 256, 384, or 512 bits	Generated internally using a NIST [SP 800-90A] DRBG.	N/A	Plaintext in volatile memory	Zeroized upon use	Establish symmetric keys
NIST SP800-90A DRBG State	DRBG V, C, and Key Key: 256 bits V: 128 bits C: 128 bits	Updated as part of DRBG operation.	N/A	Plaintext in volatile memory only	Zeroized upon use	Generate random numbers
Entropy Input	Entropy 384 bits	Provided by external process	N/A	Plaintext in volatile memory only	Zeroized upon use	Initialize DRBG
TLS Master Secret	TLS Master Secret 384 bits.	Derived from the TLS pre-master secret using the TLS KDF	N/A	Plaintext in volatile memory only	Zeroized when TLS session is over	Derive keys in TLS sessions
TLS Pre-Master Secret	TLS Pre-Master Secret Size depends on cipher	Generated internally using a NIST [SP 800-90A] DRBG.	N/A	Plaintext in volatile memory only	Zeroized when TLS session is over	Derive keys in TLS sessions
HMAC Software Integrity Test Key	Software integrity test key 160 bits.	Hard coded	N/A	Plaintext in hard disk and in volatile memory	Zeroized when the library integrity test is completed	Software integrity test
HMAC Key	Integrity key 160, 192, 256, or 512 bits.	1. Generated internally using a NIST [SP 800-90A] DRBG. 2. Established via key agreement	N/A	Plaintext in volatile memory only	Zeroized upon use	Log file signing and TLS session integrity

2.6.1 Key Generation

The module uses a NIST [SP 800-90A] DRBG in CTR_DRBG(AES) mode by default to generate cryptographic keys (Hash_DRBG and HMAC_DRBG are also options). The DRBG is seeded differently depending on the platform that it is running on. The entropy input is 384 bits and is assumed to provide full entropy. **The module generates cryptographic keys whose strengths are modified by available entropy. There is no assurance of the minimum strength of generated keys.**

2.6.2 Key Input/Output

RSA and ECDSA public keys are output from and input into the kernel in plaintext form. Symmetric keys are input into and output from the kernel in encrypted form.

2.6.3 Key Storage

Session keys are stored in volatile memory in plaintext. RSA and ECDSA key pairs are stored in hard disk in plaintext.

2.6.4 Key Zeroization

Keys are zeroized when they are no longer used; RSA and ECDSA key pairs are zeroized when new ones are generated.

The zeroization of the keys is carried out by overwriting the storage or memory with zeros.

Any role may perform key zeroization since it is performed immediately after use of each CSP and optionally performed via power cycle.

2.7 Self-Tests

The Axway Security Kernel performs a self-test immediately after being loaded into memory. On Microsoft Windows systems, the self-tests are executed by the “DllMain” method, which runs immediately after a dll is loaded into memory. On linux, the “tmwd_libcrypto_init” method is given an attribute of (constructor), making gcc generate code that executes this method when the dll loads. On Solaris, the “so_init” method is marked with a “#pragma init”, which causes it to execute when the dll is loaded. The following self-tests are run at power-up:

- Software integrity tests using HMAC-SHA-1.
- SP800-90A DRBG KATs
- Triple-DES Encrypt/Decrypt KATs with 3 independent keys (56 bits each) in ECB mode, and a decrypt only test in CBC mode.
- AES Encrypt/Decrypt KATs with a 128-bit key in ECB mode.
- SHA-1 KAT.
- SHA-224 KAT.
- SHA-256 KAT.
- SHA-384 KAT.
- SHA-512 KAT.
- RSA Encrypt/Decrypt Pairwise Consistency Test with 2048-bit keys
- RSA Sign/Verify KATs with 2048-bit keys
- ECDSA Sign/Verify KATs using SECP 224R1 and SECT 233K1 curves.
- ECDH KAT using SECP 224R1 curve.
- TLS KDF.

The conditional self-test performed by the module include the following four tests.

- Pair-wise consistency test for RSA keys.
- Pair-wise consistency test for ECDSA keys.
- Continuous DRBG Test.
- SP800-90A Health Checks

If the self-tests fail, the library will fail to load.

2.8 Design Assurance

Axway uses the Subversion for configuration management of source code and documentation including Axway Security Kernel's FIPS documentation. See the SVN project website http://subversion.apache.org for more information. This software provides access control, versioning, and logging.

2.9 Mitigation of Other Attacks

This section is not applicable. The module does not claim to mitigate any attacks beyond the FIPS 140-2 level 1 requirements for this validation.

3 Secure Operation

The Axway Security Kernel meets Level 1 requirements for FIPS 140-2. The sections below describe how to place and keep the module in FIPS-approved mode of operation.

3.1 Crypto Officer Guidance

The Crypto Officer is responsible for installing, uninstalling, configuring, and managing the module and running the power-up self-tests on demand. Before installing the module, the Crypto Officer should make sure that the specific OS is in single user mode.

3.1.1 Operation System Configuration

The Crypto Officer must maintain control of the installation media.

FIPS 140-2 mandates that a cryptographic module be limited to a single user at a time. Before the module can be installed, the Crypto Officer must have a standard PC or mainframe computer running on one of the OS listed in Column 1 of Table 1. The OS being used must be configured for single user mode and disallow remote login.

To configure Windows for single user mode, the Crypto Officer must ensure that all remote guest accounts are disabled in order to ensure that only one human operator can log into the Windows OS at a time. The services that need to be turned off for Windows are

- Fast-user switching (irrelevant if PC or server is a domain member)
- Terminal services
- Remote registry service
- Secondary logon service
- Telnet service
- Remote desktop and remote assistance service

Once the Windows OS has been properly configured, the Crypto Officer can use the system “Administrator” account to install software, uninstall software, and administrate the module.

The specific procedure to configure RHEL for single user mode is described below.

1. Login as the “root” user.

2. Edit the system files /etc/passwd and /etc/shadow and remove all the users except “root” and the pseudo-users. Make sure the password fields in /etc/shadow for the pseudo-users are either a star (*) or double exclamation mark (!!). This prevents login as the pseudo-users.

3. Edit the system file /etc/nsswitch.conf and make “files” the only option for “passwd”, “group”, and “shadow”. This disables Network Information Service and other name services for users and groups.

4. In the /etc/xinetd.d directory, edit the files “rexec”, “rlogin”, “rsh”, “rsync”, “telnet”, and “wu-ftpd”, and set the value of “disable” to “yes”.

5. Reboot the system for the changes to take effect.

More information can be found at http://csrc.nist.gov/groups/STM/cmvp/documents/CMVPFAQ.pdf.

Once the operating system has been properly configured, the Crypto Officer can use the system “root” account to install/uninstall software and administrate the module.

The specific procedure to configure Solaris 10 for single user mode is described below:

1. Login as the "root" user.

2. Edit the system files /etc/passwd and /etc/shadow and remove all the users except "root" and the pseudo-users (daemon users). Make sure the password fields in /etc/shadow for the pseudo-users are either a star (*) or double exclamation mark (!!). This prevents login as the pseudo-users. Also make sure the shell for daemon users is /dev/null, or something else that is not exploitable.

3. Edit the system file /etc/nsswitch.conf and make "files" the only option for "passwd", "group", and "shadow". This disables NIS and other name services for users and groups.

4. Edit the system file /etc/inet/inetd.conf, and comment out all unnecessary services (by prepending a hash ('#') sign to the beginning of each unnecessary service line).

   sadmind - Solstice network administration agent server
   rpc.ttdbserverd - Sun tool-talk server
   kcms_server - Kodak Color Management System server
   fs.auto - Sun font server
   cachefsd - NFS cache service
   rquotad - remote disk quota server
   rpc.metad - Disksuite remote metaset service
   rpc.metamhd - Disksuite remote multihost service
   rpc.metamedd - Disksuite component service
   ocfserv - Smartcard service
   dtspcd - Part of the CDE package
   rpc.cmsd - remote calendar server
   in.comsat - biff, mail notification server
   in.talkd - talk server
   gssd - RPC application authentication
   in.tnamed - deprecated name server
   rpc.smserverd - removable media device sensor service (disabling requires manual CD mounting)
   dcs - remote dynamic configuration server
   ftpd - ye olde FTP server
   ktkt_warnd - Kerberos warning server
   chargen - deprecated network service
   daytime - deprecated network time
   time - legacy time service
   discard - deprecated network service
   echo - network 'echo' service
   ufsd - part of RPC
   in.uucpd - unix-to-unix copy server

5. Disable service startup scripts within /etc/rc2.d. Many additional services (not bound to inetd) are started by default. To disable startup scripts, files can be renamed to make sure they do not begin with a capital ‘S’ (which denotes Startup). Disable startup scripts that are not pertinent to the setup.

   nscd - NIS-related
   snmpdx - SNMP services
   cachefs.daemon - NFS-caching
   rpc - Remote Procedure Call services
   sendmail – Sendmail
   lp - line printer daemon
   pppd - Point-to-point Protocol services

uucp - Unix-to-Unix copy daemon
ldap - LDAP services

6. Reboot the system for the changes to take effect.

Once the operating system has been properly configured, the Crypto Officer can use the system “root” account to install/uninstall software and administrating the module.

3.1.2 Initialization

The software module will be provided to the users by Axway Inc. along with the client applications, including Validation Authority and MailGate. The module is installed during installation of the client application. The installation procedure is described in the client application’s installation manual.

The module must be installed, configured, and started before operators may utilize its features.

3.1.3 Zeroization

Zeroization of keys and other CSPs is controlled and performed by client applications. Zeroization may be manually invoked by rebooting the computer on which the kernel is running. Uninstalling the client application also results in zeroization of all keys and other CSPs.

3.1.4 Management

The Crypto Officer does not perform any management of the kernel after installation and configuration. The management tasks are conducted by the client application.

3.2 User Guidance

The module’s cryptographic functionality and security services are provided via client applications. Only the Approved and non-Approved but allowed algorithms listed in Section 2.1 shall be used by the client application in the Approved mode. End-user instructions and guidance are provided in the user manual and technical support documents of the individual client application software. Although the end-users do not have any ability to modify the configuration of the module, they should check that the client application is present and enabled and thereby providing cryptographic protection. The mode of operation is procedurally controlled and is implicitly determined based on the Approved or non-Approved services/algorithms invoked. The module is only in an Approved mode of operation when utilizing Approved services and algorithms as described in this Security Policy.

4 Acronyms

Table 10 – Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
API	Application Programming Interface
BIOS	Basic Input/Output System
CA	Certificate Authority
CBC	Cipher Block Chaining
CD	Compact Disc
CFB	Cipher Feedback
CMVP	Cryptographic Module Validation Program
CPU	Central Processing Unit
CSP	Critical Security Parameter
CVS	Concurrent Versions System
DER	Distinguished Encoding Rules
DLL	Dynamic Link Library
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
ECB	Electronic Codebook
ECDSA	Elliptic Curve Digital Signature Algorithm
EDI	Electronic Data Interchange
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FCC	Federal Communication Commission
FIPS	Federal Information Processing Standard
HMAC	(Keyed-) Hash MAC
IDE	Integrated Drive Electronics
IP	Internet Protocol
ISA	Instruction Set Architecture
KAT	Known Answer Test
KDF	Key derivative function
MAC	Message Authentication Code
N/A	Not Applicable
NIST	National Institute of Standards and Technology
PEM	Privacy-enhanced Electronic Mail
OCSP	Online Certificate Status Protocol

Acronym	Definition
OFB	Output Feedback
OS	Operating System
PC	Personal Computer
PCI	Peripheral Component Interconnect
PKCS	Public Key Cryptography Standard
PKI	Public Key Infrastructure
RAM	Random Access Memory
RNG	Random Number Generator
ROM	Read Only Memory
RSA	Rivest, Shamir and Adleman
SHA	Secure Hash Algorithm
SO	Shared Object
TDES	Triple Data Encryption Standard
TLS	Transport Layer Security
USB	Universal Serial Bus
VSS	Visual Source Safe
X509	PKI infrastructure standard
X509V3	PKI infrastructure standard version 3
XML	Extensible Markup Language