Quantum Corporation

Scalar Key Manager

Software Version 2.0.1

FIPS 140-2 Non-Proprietary Security Policy

Document Version 1.4

Last Update: 2010-11-03 8:43:00 AM

Table of Contents

Document History

Version	Date of Change	Author	Changes to Previous Version
1.0	2010-02-19	Quantum	Release Version
1.1	2010-03-17	Quantum	Updated version information
1.2	2010-06-15	atsec	Updated certificates for OpenSSL and OE
1.3	2010-08-12	atsec	Addressed NIST comments
1.4	2010-10-28	atsec	Updated certificates and OpenSSL information

Acronyms

AES	Advanced Encryption Standard
AK	Authentication Key
ANSI	American National Standards Institute
CO	Cryptographic Officer
DEK	Data Encryption Key
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FIPS	Federal Information Processing Standard
KEK	Key Encryption Key
HMAC	Hash-based Message Authentication Code
MD5	Message Digest 5
NIST	National Institute of Standards and Technology
RAM	Random Access Memory
RHEL	Red Hat Enterprise Linux
RNG	Random Number Generator
RSA	Rivest, Shamir, Adleman
SHA	Secure Hash Algorithm
SKM	Scalar Key Manager
TLS	Transport Layer Security

1.Cryptographic SKM Specification

Scalar Key Manager (SKM), Software Version 2.0.1, is a software solution that provides symmetric encryption key management, key generation, secure key retrieval, key database replication, and compliance audit logging of all key access and configuration functions. The software runs on a hard, opaque, commercial grade general purpose computer and was tested on rPath Linux V. 2.6.29. The cryptographic SKM provides logical interfaces for data input, data output, status output, and command input through its command interface.

Security Component	Security Level
Cryptographic SKM Specification	1
Cryptographic SKM 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	1
Mitigation of Other Attacks	N/A

1.1.Description of Approved Mode

The SKM only supports an Approved mode of operation. Once the module has successfully completed its power-on self-tests, it is in the Approved mode, which is indicated by the following status messages on the administrative console:

StartupTests successful
self test failures <0>
OpenSSL FIPS mode <1>

The following algorithms are supported by the SKM:

- AES (Cert. #1255)
- SHA-256 (Cert. #1151)
- HMAC-SHA-256 (Cert. #734)
- NIST-Recommended ANSI X9.31 RNG (Cert. #698)

In addition to the algorithms above, the SKM also makes use of the Red Hat Enterprise Linux 5 (RHEL) OpenSSL Cryptographic Module (Cert. #1320) by means of the portability allowance specified in FIPS 140-2 Implementation Guidance G.5, for RSA and cryptographic functionality to support TLS, which includes the following algorithms that have been re-tested for added assurance:

- RSA Sign/Verify, provided by OpenSSL (Cert. #736)
- RSA (key wrapping; key establishment methodology provides 80 bits or 112 bits of encryption strength), provided by RHEL OpenSSL
- AES, provided by OpenSSL (Cert. #1499)
- HMAC-SHA-1, provided by OpenSSL (Cert. #882)
- SHA-1, provided by OpenSSL (Cert. #1350)
- ANSI X9.31 RNG, provided by OpenSSL (Cert. #816)

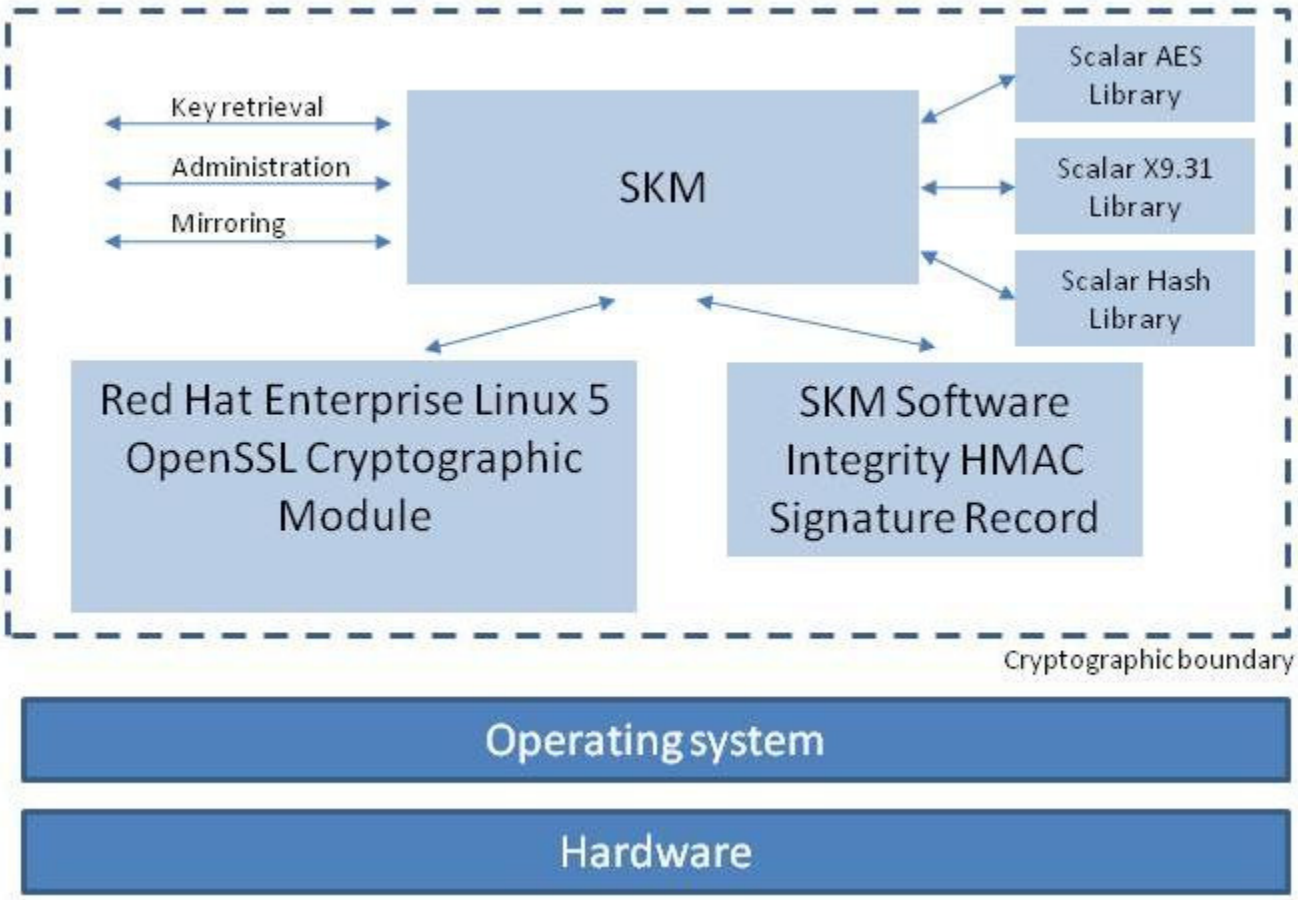
- MD5 within TLS only, provided by OpenSSL

The Red Hat Enterprise Linux 5 OpenSSL Cryptographic Module was built in accordance to its Security Policy and User Guide and has not been modified in any way.

1.2.Cryptographic SKM Boundary

The SKM is a software only product, but for the purposes of the FIPS 140-2 validation, it is considered a multiple-chip standalone module. The logical cryptographic boundary is defined by the executable application file (V. 2.0.1), the Scalar AES Library (V. 2.0.3), the Scalar X9.31 Library (V. 1.1.0), the Scalar Hash Library (V. 2.0.3), the Red Hat Enterprise Linux 5 OpenSSL Cryptographic Module (Cert. #1320[1]), and the HMAC signature value contained within the SQLite database, where as the physical embodiment is the general purpose computer or hardware appliance on which the SKM operates. The physical cryptographic boundary contains the general purpose computing hardware of the system executing the application. This system hardware includes the central processing unit(s), cache and main memory (RAM), system bus, and peripherals including disk drives and other permanent mass storage devices, network interface cards, keyboard and console and any terminal devices.

1.3.Block Diagram



[1] The Red Hat Enterprise Linux 5 OpenSSL Cryptographic Module is separately version controlled within Quantum as the “Scalar OpenSSL FIPS Lib“, Version 3.0.0, as identified on the associated algorithm certificates.

2.Cryptographic SKM Ports and Interfaces

The SKM provides a logical interface via its service input and output parameters.

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

- Data input – Data input to all functions that accept input from Crypto Officer or User
- Data output – Data output from all functions that return data as arguments or return values from Crypto Officer or User
- Control input – All control input into functions by the Crypto Officer and User
- Status output – Status information returned to Crypto Officer or User as return/exit codes and log entries

The API function specifications are included in the Scalar Key Manager Administrative Interface Guide and Scalar Key Manager Key Retrieval Guide, which cover all inputs and outputs for each service.

3.Roles, Services and Authentication

The cryptographic module supports two distinct roles: Cryptographic Officer (CO) and User. The CO is the individual(s) responsible for creating and managing cryptographic keys through the key life cycle, and related key access policies, while the User is the individual(s) who retrieves data encryption keys for use. These roles are explicitly assumed by selecting the appropriate TLS certificate for use when requesting a service from the SKM.

Role	Type of authentication	Authentication data
Cryptographic Officer	N/A for FIPS 140-2 Level 1	N/A for FIPS 140-2 Level 1
User	N/A for FIPS 140-2 Level 1	N/A for FIPS 140-2 Level 1

The following table specifies the service classes that are supported and the specific commands associated with each.

Role	Service Type	CSP	Algorithm	Service Name	Access
CO	Key creation	KEK, DEK, RNG, AK	AES, RNG, HMAC	Automatically generate keys, Create symmetric key, Change next increment,	Read, Write
User	Key retrieval	RNG, KEK, AK	AES, RNG, HMAC	Get next key, Get symmetric key	Read
CO	Manage keys	AK	HMAC	Activate key, Activate key instance, Change activation date, Change deletable, Change expiration date, Change mirror key option,	Read, Write

				Change key rollover, Remove template record, Revoke key, Revoke key instance, Rollover key, Set key access flag, Set meta data	
CO	Manage key access	None	N/A	Add user to group, Authorize admin, Delete group from group access, Delete group from group member, Delete key from group access, Delete key from user access, Delete user from group member, Delete user from user access, Get group access list, Get group list for key, Get group list for user, Get group member list, Get key access flag, Get key access list, Get key list for group, Get key list for user, Get user access list, Get user list for group, Get user list for key, Remove group access to key, Remove key from key access, Remove user access to key, Remove user from group, Set group access to key, Set user access to key	N/A
CO	Symmetric key deletion	AK	HMAC	Delete key, Delete key instance	Read
CO	Symmetric key reports	AK	HMAC	Display key instance list, Display key name list, Display symmetric key policy, Get template depth, Get template list, Retrieve meta data	Read
CO	Manage certificates and private keys	Asymmetric	N/A	Delete certificate, Delete private key, Export certificate, Get certificate list, Get private key list, Import certificate, Import private key	Read, Write
CO	Symmetric key import and export	Asymmetric, KEK, AK	RSA, AES, HMAC	Export symmetric key, Export symmetric key batch, Import symmetric key, Import symmetric key batch, Push key to device	Read, Write
CO	Status	AK	HMAC	Crypto self test,	Read

				Get system status, Report FIPS-140 mode, Administration NOOP, Validate database	
CO	Key mirroring	DEK, KEK, AK, Asymmetric	AES, HMAC, RSA	Force key synchronization, Get mirror address, Get mirror status, Get mirrored data hash, Get queue size, List mirror names, Remove mirror address, Set mirror address, Trigger put	Read, Write
CO	Server management	None	N/A	Set log level, Stop key store	N/A

4.Physical Security

The SKM is a software only module and as such, the physical security requirements defined in FIPS 140-2 are not applicable.

5.Operational Environment

The module operates on a modifiable operational environment, rPath Linux, V. 2.6.29. The operating environment must be configured for single-user mode. The operator of the module is the single-user.

The module was tested on an IBM System x3250 M2 (4190AC1) Server Appliance running rPath Linux V. 2.6.29 64-bit with an Intel Xeon 3120 64-bit processor.

6.Cryptographic Key Management

The SKM supports the following CSPs and public keys (Note: All secret and private cryptographic keys are automatically zeroized at the end of each session or program termination):

Key Encryption Key (KEK)	AES, 256-bit Protects DEKs stored in the database
KEK RSA Private Key	RSA, 2048-bit Unwraps the stored KEK for use
Authentication Key (AK)	HMAC SHA-256, 256-bit DEKs stored encrypted along with key policy data in database records. Each record has an HMAC SHA-256 hash of the policy data and encrypted key value appended to it
AK RSA Private Key	RSA, 2048-bit Unwraps the stored AK for use
Data Encryption Key (DEK)	AES, 128, 192, or 256-bit

	Data encryption keys created, stored, and managed by the SKM.
RNG Seed	Seed Value, 128-bit value SKM RNG and OpenSSL RNG
Master Key	RNG AES Seed Key, 256-bit SKM RNG and OpenSSL RNG
Import Private Key	RSA, 2048-bit Used to import DEKs
TLS SKM Private Key	RSA, 1024 or 2048-bit Used to establish the TLS Session
TLS Encryption Keys	AES, 128, 192, or 256-bit Protects the TLS session
TLS Integrity Keys	HMAC SHA-1, 256-bit Data authentication for the TLS session
Software Integrity Key	HMAC SHA-256, 256-bit Provides data authentication/integrity of the software module
KEK RSA Public Key	RSA, 2048-bit Protects the KEK, which is persistently stored on the Server
AK RSA Public Key	RSA, 2048-bit Protects the AK, which is persistently stored on the Server
TLS SKM Public Key	RSA, 1024 or 2048-bit Used to establish the TLS Session
TLS CA Certificate	X.509 Certificate Validates client TLS certificates
TLS Client/Server Public Key	RSA, 1024 or 2048-bit Used to establish the TLS session
Partner Export Public Key	RSA, 2048-bit Used to wrap exported outbound DEKs

7.Electromagnetic Interference/Electromagnetic Compatibility

Operational testing was performed on an IBM x3250 M2 server appliance, which has received appropriate FCC certification for FIPS 140-2 Level 1.

8.Self Tests

8.1.Power-Up Tests

On power up the application performs known-answer tests for the following cryptographic functions:

- AES KAT
- HMAC SHA-256 KAT (As part of the Software Integrity Test)
- SHA-256 KAT (As part of the Software Integrity Test)
- RNG KAT
- Software Integrity Test (HMAC SHA-256)

Upon successful completion of the power-up self-tests, the following is output to the log file:

StartupTests successful
self test failures <0>
OpenSSL FIPS mode <1>

If power-up self-tests do not complete successfully, the module will exit.

8.2.Conditional Tests

The SKM will perform the following conditional test:

- Continuous RNG Test on the NIST-Recommended ANSI X9.31 RNG

8.3.Critical Functions Tests

The application performs a database validation test to inspect for key corruption or substitution on the associated SQLite database. Each time the module reads a record from the database, the module calculates an HMAC SHA-256 hash over the requested record and compares the calculated result with the expected, known hash that is stored along with the record. If the regenerated HMAC does not match the original HMAC, the key record is marked as corrupt. Alternatively, the operator may decide to invoke the “Validate database” service, which will iterate through each record contained within the entire database and verify each.

9.Mitigation of Other Attacks

The SKM does not mitigate any attacks beyond the scope of FIPS 140-2.