*CommVault Systems, Inc.*

# CommVault Crypto Library, v1.0 FIPS 140-2 Non-Proprietary Security Policy

## Level 1 Validation

**February 2009**

## +Table of Contents



# 1 INTRODUCTION

## *1.1 Overview*

CommVault Crypto Library (CVCL) is a cryptographic software module used in various products by CommVault Systems, Inc. The module provides a collection of FIPS Approved and Non-FIPS Approved cryptographic services for key generation, symmetric and asymmetric encryption, hash, HMAC and signature generation/verification.

## *1.2 Purpose*

This document is a non-proprietary FIPS 140-2 Level 1 Security Policy for the CVCL Module version 1.0 serving the following purposes:

1. It describes how the module conforms to the eleven sections of the FIPS 140-2 Level 1 Standard.
2. It provides user with instructions on how to install and operate the module in order to comply with FIPS 140-2.
3. It is required for the FIPS 140-2 validation.

FIPS 140-2 (Federal Information Processing Standards Publication 140-2) specifies the security requirements for a cryptographic module protecting sensitive information. Based on four security levels for cryptographic modules this standard identifies requirements in eleven sections. For more information about the standard please visit http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf.

## 1.3 References

- NIST Security Requirements for Cryptographic Modules, FIPS PUB 140-2, December 03, 2002
- NIST Security Requirements for Cryptographic Modules, Annex A: Approved Security Functions for FIPS PUB 140-2, May 19, 2007
- NIST Security Requirements for Cryptographic Modules, Annex B: Approved Protection Profiles for FIPS PUB 140-2, June 14, 2007
- NIST Security Requirements for Cryptographic Modules, Annex C: Approved Random Number Generators for FIPS PUB 140-2, March 19, 2007
- NIST Security Requirements for Cryptographic Modules, Annex D: Approved Key Establishment Techniques for FIPS PUB 140-2, June 26, 2007
- Advanced Encryption Standard, FIPS PUB 197, November 11, 2001
- The Keyed-Hash Message Authentication Code (HMAC), FIPS PUB 198, March 6, 2002
- Additional publications can be found on NIST website at http://csrc.nist.gov/publications/PubsFIPS.html
- For more information about CommVault Systems, Inc. visit http://www.commvault.com.

## 1.4 Document History

| Authors | Date | Version | Comment |
|---|---|---|---|
| Andrei Erofeev | 11/01/07 | 1.0 | Initial revision |
| Andrei Erofeev | 09/30/08 | 1.1 | Feedback from DOMUS |
| Andrei Erofeev | 12/30/08 | 1.2 | Feedback from CMVP |
| Andrei Erofeev | 02/23/09 | 1.3 | More comments from CMVP |
| Andrei Erofeev | 04/27/09 | 1.4 | Clarified PRNG retries |

# 2 CRYPTOGRAPHIC MODULE SPECIFICATION

CommVault Crypto Library (CVCL) is a software library providing cryptographic services to all CommVault products, particularly to Simpana, the enterprise level backup solution. Simpana is composed of several program modules allowing one to perform data backup, hierarchical storage management (data archiving) and continuous data replication.

CVCL implements Triple-DES, AES (data encryption, key protection), RSA (signing, verification), SHA-1, SHA-256, SHA-512, HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA512 and ANSI X9.31PRNG algorithms in the approved FIPS mode.

In addition to the FIPS Approved algorithms CVCL implements Blowfish, Serpent, Twofish, DES, RSA (encrypt/decrypt), MD5, and HMAC-MD5 algorithms when operated in the non-approved mode.

CVCL is packaged as a dynamic (shared) software module exporting cryptographic API to any software that supports C calling conventions.

The product meets the overall requirements applicable to Level 1 security for FIPS 140-2 as outlined in the following table.

| Security Requirements Section | Level |
|---|---|
| Cryptographic Module Specification | 1 |
| Cryptographic Module Ports and Interfaces | 1 |
| Roles and Services and Authentication | 1 |
| Finite State Machine Model | 1 |
| Physical Security | N/A |
| Operational Environment | 1 |
| Cryptographic Key Management | 1 |
| EMI/EMC | 3 |
| Self-Tests | 1 |
| Design Assurance | 3 |
| Mitigation of Other Attacks | N/A |
| Cryptographic Module Security Policy | 1 |
| Overall Level of Certification | 1 |

**Table 1: Module Compliance Table**

The module was tested and validated on a machine running Microsoft Windows 2003.
The module was tested and validated on a machine running Red Hat Advanced Server 5.0.
The module was tested and validated on a machine running Sun Solaris 10.

# 3 CRYPTOGRAPHIC MODULE PORTS AND INTERFACES

The module is classified as a multi chip standalone module designed to meet FIPS 140-2 Level 1 requirements. The physical interfaces of the module are the same as the computer system on which it is executing. The logical interfaces are the C API calls through which the module is providing its services. In particular, **Data Input Interface** is a collection of C API calls that accept through their arguments data to be used in cryptographic operations, **Data Output Interface** is a collection of C API calls that return processed or generated data back to the controlling program through their arguments or return codes, **Control Input Interface** is a collection of C API calls that are used to initialize and control the operation of the module. **Status Output Interface** is a C API call that is used to query the status of the module.

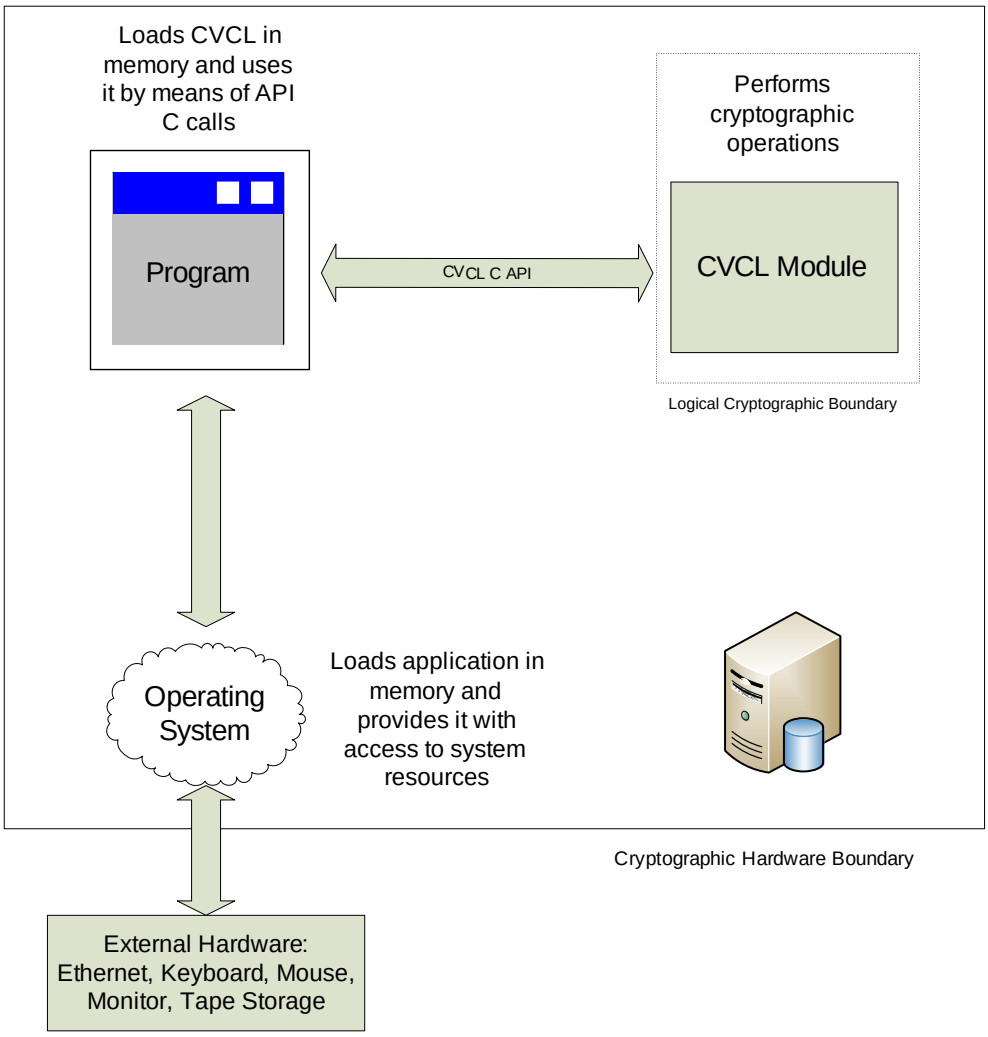


**Figure 1: Software Block Diagram**

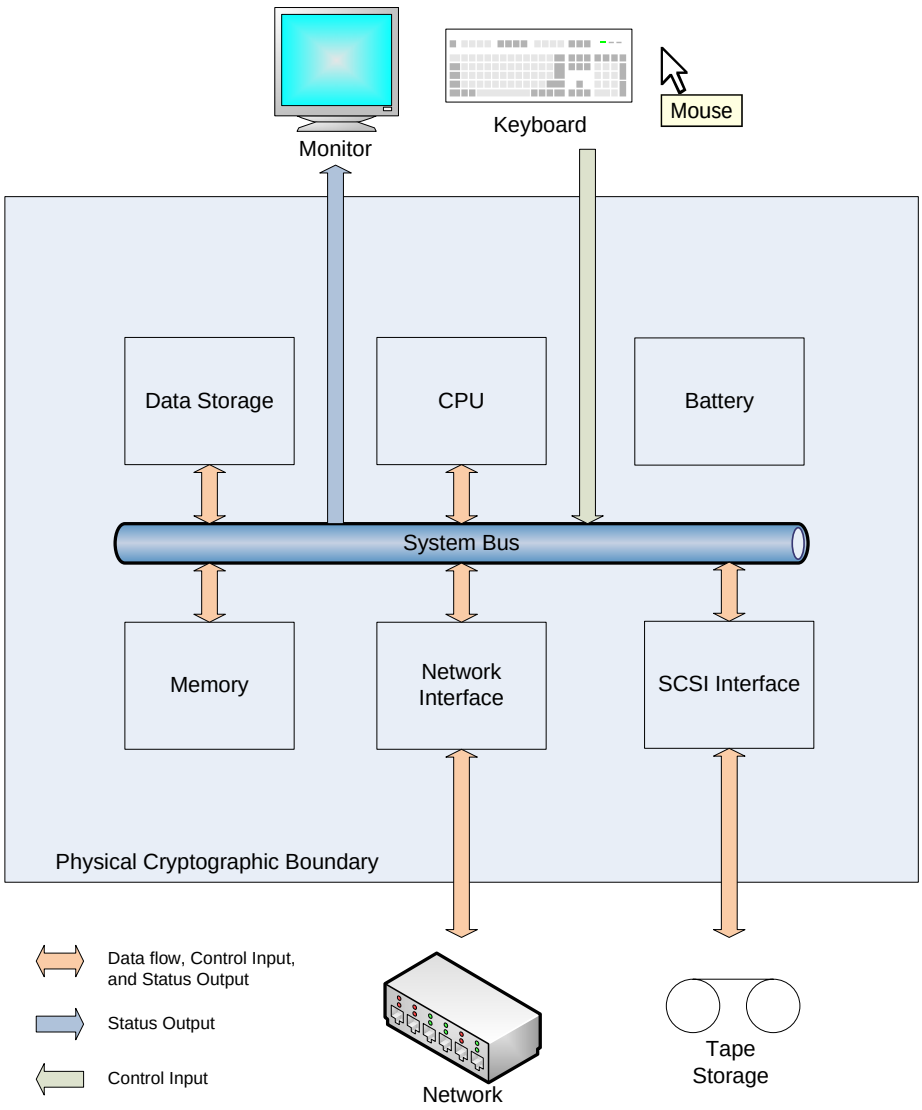


**Figure 2: Hardware Block Diagram**

The table below describes supported logical and physical interfaces as well as the relationships between them.

| FIPS 140-2 Interface | Logical Interface | Physical Interface |
|---|---|---|
| Data Input interface | The API C calls that accept input data for processing through their arguments. | Ethernet, mouse, keyboard. |
| Data Output interface | The API C calls that return by means of their return codes or arguments generated or processed data back to the caller. | Ethernet, mouse, keyboard, monitor. |
| Control Input interface | The API C calls that are used to initialize and control the operation of the CVCL module. | Ethernet, mouse, keyboard. |
| Status Output interface | The API C calls that are used to query the status of the | Ethernet, mouse, keyboard, monitor, hard disk. |

| | | |
|---|---|---|
| | CVCL module. Cvcl.log file where the status is being output to after completion of initialization and POST. | |
| Power Interface | N/A | 120V Power Supply |

**Table 2: Mapping Physical and Logical Interfaces**

# 4 ROLES, SERVICES AND AUTHENTICATION

## *4.1 Identification and Authentication*

The CVCL module implements the following two roles: Crypto-Officer and User role. Maintenance role is not supported. The roles are implicitly assumed and the module does not provide an authentication mechanism that would allow to explicitly distinguish users between the two supported roles. The module relies on the Operating System to implement fine-grained access control, thus if only certain user should be permitted to use the module's services, the OS authentication mechanisms should be enabled by the OS administrator.

| *Role* | *Type of Authentication* | *Authentication Data* |
|---|---|---|
| **User** | N/A | N/A |
| **Crypto Officer** | N/A | N/A |

**Table 3: Authentication Type Table**

## *4.2 Roles and Services*

The table below lists supported FIPS Approved and Non-FIPS Approved authorized services. For each service it specifies which role the service can be used in, what cryptographic keys and critical security parameters (CSP) the service can access and how.

**R** - The item is **read** or referenced by the service.
**W** -The item is **written** or updated by the service.
**E -** The item is **executed** by the service. (The item is used as part of a cryptographic function.)

| Role | | Approved or Allowed Services | Cryptographic Keys and CSPs | Access Type |
|---|---|---|---|---|
| Crypto Officer | User | | | |
| **FIPS Approved** | | | | |
| X | | Installation of the Module | None | N/A |
| X | X | Initialization of the Module | None | N/A |
| X | X | Power-on self-test | Module RSA Public Signature Key | RE |
| X | X | Key Generation | Symmetric Keys, Asymmetric Key Pairs | RE |
| X | X | Key Zeroization | Symmetric Keys, Asymmetric Key Pairs | WE |
| X | X | Querying the state of the module | None | N/A |
| X | X | Symmetric Data Encryption/Decryption<br>• Triple DES – ECB, CBC modes with 192-bit keys.<br>• AES – ECB, CBC modes with 128-bit and 256-bit keys. | Triple DES, AES keys | RWE |
| X | X | Digest Algorithms | None | N/A |

| | | | | |
|---|---|---|---|---|
| | | • SHA-1<br>• SHA-256<br>• SHA-512 | | |
| X | X | Message Authentication<br>• HMAC-SHA-1<br>• HMAC-SHA-256<br>• HMAC-SHA-512<br>• RSA signature generation and verification | RSA key pairs,<br>HMAC keys | RWE |
| X | X | Pseudo-Random Number Generation<br>• ANSI X9.31 PRNG | Seed value, PRNG key | RWE |
| **Non-FIPS Approved Mode** | | | | |
| X | X | Symmetric Data Encryption/Decryption<br>• DES – ECB, CBC modes<br>• Blowfish with 128-bit and 256-bit keys – ECB, CBC modes<br>• Serpent with 128-bit and 256-bit keys – ECB, CBC modes<br>• Twofish with 128-bit and 256-bit keys – ECB, CBC modes | DES, Blowfish, Serpent, and Twofish keys | RWE |
| X | X | Asymmetric Data Encryption/Decryption<br>• RSA with 1024- , 2048- and 4096-bit keys | RSA key pairs | RWE |
| X | X | Digest Algorithms<br>• MD5 | None | RWE |
| X | X | Data Authentication:<br>• HMAC-MD5 | HMAC-MD5 key | RWE |

**Table 4: Roles and Services**

# 5 PHYSICAL SECURITY

Physical security is not applicable to this software module at Security Level 1.

# 6 OPERATIONAL ENVIRONMENT

CVCL Module is designed to work on a General Purpose Computer with one of the supported operating systems:

1. Microsoft Windows 2003
2. Red Hat Advanced Server 5.0
3. Solaris 10

The operating system segregates the computing environment into processes spaces. Though all processes share the same hardware resources, they are logically separated from each other by the operating system. Each process has an independent pool of virtual memory, and the CVCL module functions completely within the address space of the process that loads it. CVCL module does not attempt to communicate with other processes or other components of operating system, and operating system shouldn't allow other processes to interfere with the module. On this accord the module satisfies the FIPS 140-2 requirement for a single user mode of operation.

It is the administrator's responsibility to configure OS authentication mechanisms and ensure that only authorized users have access to the software that is loading the CVCL Module.

Since CVCL Module is software executing within a General Purpose Computer, and physical cryptographic boundary is drawn through all the hardware interfaces of the Computer, such as Ethernet or SCSI Bus, all major data paths within computer must be protected.

The replacement or modification of the Module by unauthorized parties is explicitly prohibited.

# 7 CRYPTOGRAPHIC KEY MANAGEMENT

## *7.1 Implemented algorithms*

CVCL Module provides low-level key generation and management routines for all algorithms listed in Table 4 of this Security Policy. Since the module implements both FIPS Approved and Non-FIPS Approved algorithms, it is the responsibility of the user to ensure that only the FIPS Approved services are being used (Please refer to **Section 10 "Crypto-Officer and User Guide"** for more details).

## *7.2 Key Generation*

The module provides services to generate pseudo-random symmetric and asymmetric keys according to the FIPS Approved ANSI X9.31 standard.

## *7.3 Key Entry and Output*

CVCL Module does not import or export keys across the physical cryptographic boundary. It is the responsibility of the application that loads the Module to protect keys when they're being exported or imported across the physical cryptographic boundary. It is also the responsibility of the application to ensure that only FIPS Approved cryptographic algorithms are being used for key protection.

CVCL Module accepts and passes keys across logical cryptographic boundary as parameters via API calls.

## *7.4 Key Storage*

The module does not provide any long-term key storage. Keys stored in the NVRAM are protected from unauthorized disclosure, access or modification by the operating system that is responsible for allocating isolated and independent virtual memory for the module and the process using it.

## *7.5 Zeroization of Keys*

The following precautions are taken to make sure that all keys and seeds are being destroyed properly:

1. CVCL Module zeroizes all intermediate security sensitive material.
2. CVCL Module zeroizes all keys passed to the authorized services when the services are no longer needed and are being destroyed.
3. CVCL Module provides API that can be used to explicitly reset any of the authorized services at any time and to zeroize keys and seeds being used by them.

## *7.6 Supported keys*

The following table summaries the module's keys and CSP's:

| Key | Generation | Use | Role |
|---|---|---|---|
| Triple DES, AES | Generated internally using a PRNG compliant to ANSI X9.31 | Used to encrypt, decrypt data and protect keys. | User, CO |
| RSA Keys | Generated internally using a PRNG compliant to ANSI X9.31 | Used to perform encrypt/ decrypt operations and signature generation/ verification. | User, CO |
| Module RSA Public Key | Hard coded within the module | Used to verify the software integrity at initialization | User, CO |
| HMAC keys | Generated internally using a PRNG compliant to ANSI X9.31 | Used to generate and verify HMAC | User, CO |
| ANSI X9.31 PRNG Seed Key | Gathered from internal OS data | Used to seed ANSI X9.31 PRNG | User, CO |

**Table 5: Cryptographic Keys and CSPs**

The module keys map to the following algorithms certificates:

| Approved Security Function | Certificate |
|---|---|
| **Encryption/Decryption** | |
| TDES (ECB, CBC) | 700 |
| AES (ECB, CBC) | 847 |
| **Hash Functions** | |
| SHA-1 | 838 |
| SHA-256 | 838 |
| SHA-512 | 838 |
| **Message Authentication** | |
| HMAC-SHA-1 | 465 |
| HMAC-SHA-256 | 465 |
| HMAC-SHA-512 | 465 |
| **Digital Signatures** | |
| RSA PKCS1-v1_5 | 405 |
| **Random Number Generation** | |
| ANSI X9.31 | 482 |

**Table 6: FIPS Approved Algorithms Table**

# 8 SELF-TESTS

## *8.1 Power-On Self Tests*

As described in **Section 2 "Cryptographic Module Specification"**, CommVault Crypto Library is a software shared library that can be loaded by one of the CommVault applications. Before the library can be used in cryptographic operations, it must be initialized. The loading application performs initialization by calling the CVCL initialization function, which internally executes the following two Power-On Self Tests, upon successful completion of which the library is switched to the Initialized State. While running the POST, data output and data input interfaces are inhibited.

In order to run the POST manually on-demand, the module must be re-instantiated.

### 8.1.1 Software Integrity Test

The module is shipped with a precomputed RSA EMSA-PKCS1-v1_5 signature. As part of the software integrity check, the entire module shared library (DLL) is verified against this signature. If the signature verification fails, Module is transitioned to the error and finally back to the non-initialized state, where no further cryptographic operations are possible. In this case the user should retry initialization, and if this doesn't fix the error, uninstall and re-install the module from the original installation media.

### 8.1.2 Known Answer Tests

During Initialization the module performs a series of Known Answer Tests for all FIPS Approved and Non-FIPS Approved algorithms according to the following table:

| Algorithm Type | Known Answer Test Executed |
|---|---|
| Symmetric Encryption, Decryption: AES and Triple-DES in ECB and CBC modes | For each of the supported symmetric encryption algorithm (Triple-DES and AES), a hard coded plaintext is encrypted with a hard coded key and the result is checked to match the pre-computed ciphertext. If the comparison is successful, the ciphertext is decrypted back, and the result is compared with the original plaintext. |
| RSA Encryption, Decryption | Hard coded plaintext is encrypted with hard coded RSA public key, and the result is compared with precomputed ciphertext. If the comparison is successful, the ciphertext is decrypted back using matching hard coded RSA private key, and the result is compared with the original plaintext. |
| RSA signature generation and verification | A hard coded signature of a hard coded plaintext is verified using a hard coded RSA public key.<br><br>A hard coded text is signed using a hard coded RSA private key, and the signature is immediately verified using the corresponding RSA public key. |
| Hash Functions: SHA-1, SHA-256 and SHA-512 | For each of the supported hash functions, a hard coded sample text of various length is hashed, and the result is compared with the precomputed hashes. |
| Message Authentication: HMAC-SHA-1, HMAC-SHA-256 and HMAC-SHA-512 | For each of the supported HMACs, a hard coded sample text of various lengths is used along with hard coded keys to generate HMAC. The results are compared with the precomputed values. |
| Random number generation | ANSI X9.31 PRNG is initialized with a hard coded seed and internal values and is requested to generate the next pseudo-random data block. The result is compared with the precomputed value. |

**Table 7: Known Answer Tests**

## *8.2 Conditional Tests*

Conditional tests are executed after the CVCL Module has been successfully initialized, during execution of one of the relevant Cryptographic Services.

### 8.2.1 ANSI X9.31 PRNG Conditional Test

This test is executed each time the next 128-bit block of pseudo-random data is generated. The module compares the new block with the previous one. If the PRNG generates two identical consecutive random numbers, the module will enter a soft error state, then immediately retry generation to get a different value. This process will continue up to four times before the module enters a hard error state and must be rebooted. If this behavior persists, the module would need to be serviced. Since the PRNG process

operates in a single threaded capacity, no cryptographic operations will be performed until this conditional test passes.

### 8.2.2 RSA Key Pair Generation Test

Each time a new RSA Key Pair is generated, it is immediately checked for correctness by generating and verifying signature of a SHA-1 hash. If the signature generation or verification fails, the Module will attempt to generate the keys 4 more times, after which it will be switched to the Error State.

## *8.3 Failure of the Self-Tests*

If one of the Power-On Self Tests fails, the Module is transitioned to the error state and then back to the non-initialized state, and the appropriate error code is returned to the program that attempted to load the Module. While the POST is being executed, both data input and data output interfaces are inhibited.

If one of the Conditional Tests fails, the Module is allowed to retry the operation 4 more times. If Conditional Tests fail for all of the following attempts, the Module is transitioned to the Error state and no further cryptographic operations can be executed until the controlling application reinitializes the Module.

# 9 DESIGN ASSURANCE

This section describes design, coding and testing practices used at CommVault during the production of the CVCL Module. It also contains Crypto Officer and User Guides detailing the correct installation, configuration, management and usage procedures that are required for the secure operation of the Module.

## *9.1 Configuration Management*

A Concurrent Versioning System (CVS) is being used by the CVCL Module development team for configuration management, change control and version control. All modifications made to the Module are logged along with the comments that accompanied them, and a complete history of changes is maintained in the CVS repository. Every individual change is assigned a unique version number. In addition, for every software release a release-specific symbolic tag is assigned to all source files and documents thereby creating a snapshot of the components included in the release.

## *9.2 Development*

The module is written using high level language “C” with several time-critical pieces optimized using architecture-specific low-level Assembler instructions on some platforms.

A software defect tracking software call Silk Radar is being used to log defects discovered during testing and to keep track of their resolution by the development team.

Weekly builds are done on all of the supported platforms. As part of the build sequence the module is compiled, linked, self-tested and signed. For every build released to the system test, a series of acceptance tests is conducted to verify at a higher level that cryptographic operations are working as expected.

# 10 CRYPTO-OFFICER AND USER GUIDE

In order to use the CVCL Cryptographic Module in a FIPS Approved Mode, it must be installed and operated in accordance with the rules described in this section.

## 10.1 Secure Setup and Initialization

CVCL Cryptographic Module is distributed on a DVD bundled with one of the CommVault Systems Products. In order to install CVCL correctly, you must follow installation instructions received with your CommVault DVD. You must have sufficient amount of space on the hard disk, enough memory and possess administrative privileges on the computer where the product is being installed. The Operating System must have an authentication mechanism configured that will allow only authorized users to use the product and the module after it has been installed. In order to use the CVCL module in single user mode, the Operating System must be configured by the administrator in such a way that only administrator's account is enabled and all other user accounts including guest's are disabled.

To validate that the CVCL Module has been installed successfully and is operated in the FIPS Approved Mode, you should do the following:

- Check version of the CVCL Module from the command line:
  1. Open command line window on the machine where Simpana is installed
  2. Navigate to the software's Base folder
  3. Execute cvcl_ver.exe executable (Windows) or cvcl_ver (UNIX) and make sure that it displays version 1.0 of CVCL Module
- Check whether the CVCL module successfully loaded and switched to the FIPS mode by opening cvcl.log file on involved computers and searching for "CVCL: Running in FIPS Mode" message.
- Check version of the CVCL Module from the Java GUI
  1. Start CommCell Console Java GUI
  2. Right-click CommServe name in the list on the left, select "Properties"
  3. In the "CommCell Properties" window open the "Version" tab
  4. Make sure that there is a "Crypto Library Version: 1.0" line on top of the window.
- Make sure that one of the FIPS Approved cryptographic services is being used for data protection
  1. Start CommCell Console Java GUI
  2. Right-click the name of the client, which data you are going to back up. Select "Properties" from the appeared popup menu
  3. In the "Properties" window navigate to the "Encryption" tab.
  4. Make sure that the "Data Encryption Algorithm" displays either "AES" or "Triple-DES".

## 10.2 Module Security Policy Rules

The module will operate in FIPS Approved mode if only FIPS Approved Cryptography Services are being used. Since the CVCL Module provides both FIPS Approved and Non-FIPS Approved Services, it will be the responsibility of the application that loaded the Module to ensure that it is using only the Approved Services.

Please refer to **Table 4** for the list and classification of all Cryptographic Services.

From the perspective of the Simpana user, correct data encryption method should be selected in the properties of each of the client machines in the CommCell:

1. Start CommCell Console Java GUI
2. Right-click the name of the client, which data you are going to back up. Select "Properties" from the appeared popup menu
3. In the "Properties" window navigate to the "Encryption" tab.
4. Make sure that the "Data Encryption Algorithm" displays either "AES" or "Triple-DES".

# 11 MITIGATION OF OTHER ATTACKS

The module does not mitigate against any specific attacks.