BeCrypt Ltd.

BeCrypt Cryptographic Library v 2.0 FIPS 140-2 Non-Proprietary Security Policy

Level 1 Validation

July 2009

Table of Contents

1 INTRODUCTION 3
1.1 Purpose 3
1.2 References 3
1.3 Document History 3
2 PRODUCT DESCRIPTION 4
2.1 High Level Block Diagram 5
2.2 Finite State Machine 6
3 MODULE PORTS AND INTERFACES 6
4 ROLES, SERVICES AND AUTHENTICATION 7
4.1 Identification and Authentication 7
4.2 Roles and Services 7
5 PHYSICAL SECURITY 9
6 CRYPTOGRAPHIC KEY MANAGEMENT 9
7 SELF-TEST 11
8 Crypto-Officer and User Guidance 12
8.1 Secure Setup and Initialization 12
8.2 Module Security Policy Rules 12
8.3 Zeroization 12

1 INTRODUCTION

1.1 Purpose

This is a non-proprietary FIPS 140-2 Security Policy for the "BeCrypt Cryptographic Library v2.0" cryptographic module. It describes how this module meets all the requirements as specified in the FIPS 140-2 Level 1 requirements. This Policy forms a part of the submission package to the validating lab.

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 **csrc.nist.gov/groups/STM/cmvp/index.html**.

1.2 References

This Security Policy describes how this module complies with the eleven sections of the Standard:

- For more information on the FIPS 140-2 standard and validation program please refer to the NIST website at **csrc.nist.gov/groups/STM/cmvp/index.html**.
- For more information about BeCrypt Ltd. please visit **www.becrypt.com**.

1.3 Document History

Authors	Date	Version	Comment
PS	21/2/08	0a1	First draft
PS	23/5/08	01	Issued
PS	12/6/08	01	Re-issued after minor amendments
PS	10/9/08	02	Re-issued after review comments
PS	12/9/08	02	Minor correction
PS	3/6/09	03	Updated for v2.0
PS	15/9/09	04	Updated after further review comments
PS	13/10/09	05	Updated after further review comments
PS	14/10/09	06	Updated after further review comments

2 PRODUCT DESCRIPTION

BeCrypt Cryptographic Library provides core cryptographic functionality for software applications. It supports AES (ECB,CBC and OFB), RSA (sign/verify, key generation, key wrapping), SHA-1, SHA-256, Random Number Generation (ANSI X9.31) and HMAC-SHA-256 algorithms in the approved mode and, MD5, DES, Triple DES and RC2 in the non-approved mode.

The cryptographic module is comprised of two sub components, viz., a 16-bit sub component object that is designed to operate in a pre-OS or DOS environment and a 32/64-bit sub component object designed to operate in 32/64-bit operating environments. It is implemented as a set of cryptographic primitives that dependent upon the underlying platform CPU hardware to provide processing and memory storage services.

The module is a multi-chip standalone cryptographic module consisting of software that executes on a general-purpose Intel® X86 General PC computing platform with an Intel® Core™ 2 CPU, configured in single-user mode. The module was tested on Windows XP Professional 32-bit SP3, Windows XP Professional 64-bit SP2, Linux Ubuntu 8.10 and MacOSX.

In this document, the BeCrypt Cryptographic Library is also referred to as “the module”.

The product meets the overall requirements applicable to Level 1 security for FIPS 140-2.

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	1
Self-Tests	1
Design Assurance	1
Mitigation of Other Attacks	N/A
Cryptographic Module Security Policy	1
Overall Level of Validation	1

Table 1 Module Compliance Table

EMI/EMC properties of the BeCrypt Cryptographic Library are not meaningful for the library itself. Systems utilizing the BeCrypt Cryptographic Library services have their overall EMI/EMC ratings determined by the host system. The validation environment used for Functional Testing had FCC Class B ratings.

2.1 High Level Block Diagram

Figure 1 shows a block diagram of the cryptographic module that illustrates the physical boundary of the module and shows the module physical interfaces. The physical cryptographic boundary is the physical boundary of the PC case.

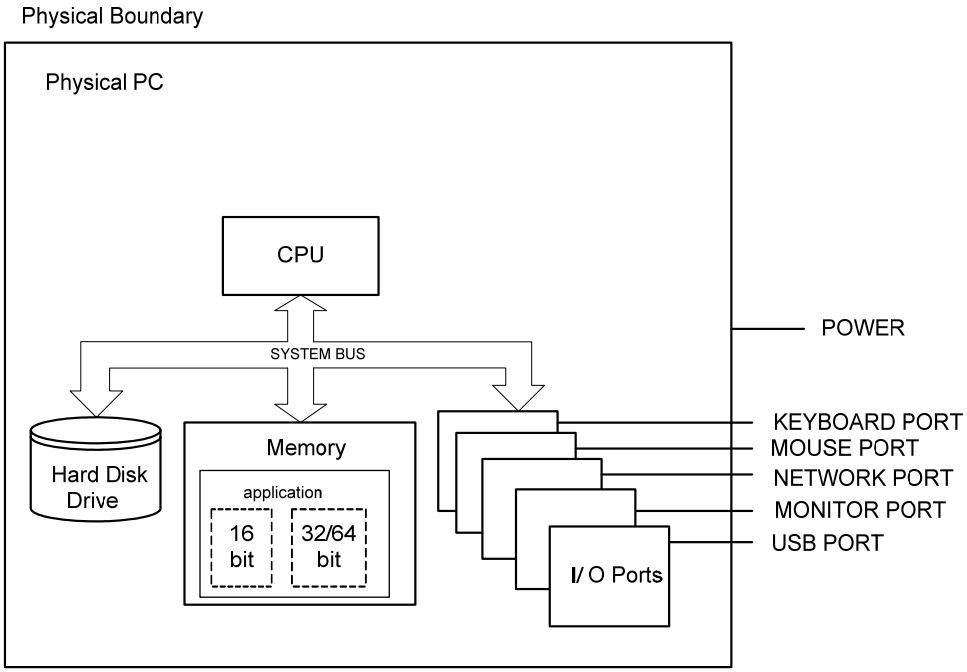

Figure 1. High Level Block Diagram Showing Physical Boundaries

Figure 2 shows a logical block diagram of the cryptographic module illustrating the components related to, and included within, the cryptographic boundary of the module. The 16 and 32/64-bit subcomponent objects are illustrated as logical components of a user application.

The cryptographic boundary includes all application files as listed below:

- fipslib – 32/64-bit cryptographic object
- becrypt – 16-bit cryptographic object

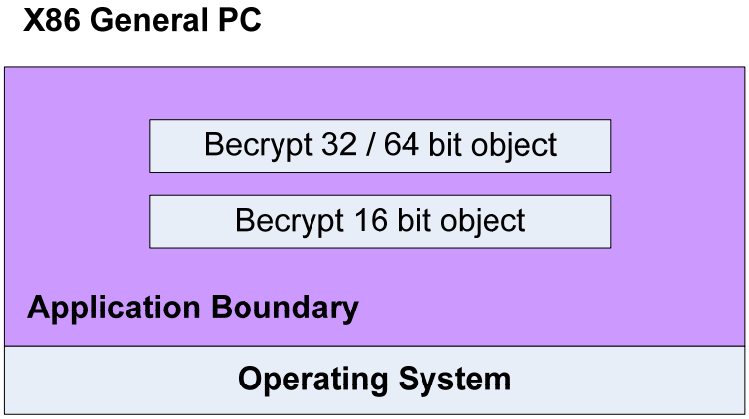

Figure 2. High Level Block Diagram Showing Logical Cryptographic Boundaries.

The application boundary is defined as the boundary of the user application that incorporates the 16-bit and 32/64-bit sub component objects.

2.2 Finite State Machine

The 'Finite State Machine Mode' description is located in the Vendor Evidence (VE) document.

3 MODULE PORTS AND INTERFACES

The physical ports map to logical interfaces as described in Table 2.

All parameters passed to the module routines are through Data Input interfaces. These interfaces accept keying material for processing that is stored by the User application in a local database.

All control flags passed to the module routines are through Control Input interfaces.

All parameters returned by the module routines are through Data Output interfaces. These interfaces pass ciphertext keying material and data.

All error codes returned by the module routines are through Status Output interfaces.

The table below maps elements of the API to the four required components of the logical interface.

FIPS 140-2 Interface	Logical Interface	Physical Port
Data Input	API functions that accept input data arguments	PC USB port, PCMCIA port, network port, Keyboard port, Mouse port, optical drive, floppy drive
Data Output	API functions that produce output in arguments	PC USB port, PCMCIA port, network port, optical drive, floppy drive
Control Input	API functions to initialize and shutdown the module and to run self tests	Mouse port, Keyboard port, PC Power Button
Status Output	API functions return values which return information regarding module status	PC monitor
Power	N/A	Supplied by device

Table 2 – Mapping Physical Ports and Logical Interfaces

The logical interfaces are described further in Table 3.

Logical Interface	Description
Data input interfaces	The data input is all plaintext data entering the API functions for the purpose of being processed by the modules services.
Data output interfaces	The data output is all plaintext data exiting the API functions after being processed by the modules services.
Control input interfaces	The control input interface is used to configure the module. This is implemented by the API functions for key generation and module initialization.
Status output interface	The status output consists of all messages either logged by the module or returned by the API functions.

Table 3. Logical Interfaces

4 ROLES, SERVICES AND AUTHENTICATION

The module supports a crypto officer (CO) role and a user role. The crypto officer and user may be different operators or they may be the same operator performing role-specific module operations. Both the crypto officer and user roles are implicitly assumed. The crypto officer role is implicitly assumed by the operator configuring the module for use at installation time. Crypto officer operations consist of configuring the PC in single user mode and installing and building applications that use the cryptographic module.

4.1 Identification and Authentication

Multiple concurrent operators are not allowed as the module is restricted to single user mode. Operators may change roles while operating the module. Access to the authorized roles is not restricted as explained in Table 4.

Role	Type of Authentication	Authentication Data
Crypto Officer	None	None
User	None	None

Table 4. Roles and Required Identification and Authentication

4.2 Roles and Services

The Becrypt Cryptographic Library supports the services listed in the following table. The table groups the authorized services by the operator roles and identifies the Cryptographic Keys and CSPs associated with the services. The modes of access are also identified per the explanation.

- **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.)

Tables 5 and 6 below show the services available to each role. Table 7 shows non-authorized services.

Role	*Authorized Services*	*Cryptographic Keys and CSPs*	*Access Type*
32/64-bit sub component			
CO	Key Generation	Seed and Seed Key	R,W
CO	AES Encrypt / Decrypt	AES data encryption key	R,E
CO	RSA Sign / Verify	RSA private and public key	R,E
CO	RSA Key Generation	RSA private and public key	W
CO	RSA Key wrapping	RSA private and public key	R.E
CO	SHA-1	None	R,E
CO	SHA-256 Hash	None	N/A
CO	HMAC-SHA-256	HMAC SHA-256 key	R,E
CO	Show Status	None	N/A
CO	Perform Self Tests	AES data encryption key HMAC integrity check key RSA public and private key RNG seed and seed key	R,E R,E R,E R,E
CO	Initialization	None	N/A
CO	Uninstall	None	N/A
16-bit sub component			
CO	Import Key	AES data encryption key	R,E
CO	AES Encrypt / Decrypt	AES data encryption key	R,E
CO	SHA-256 Hash	None	N/A
CO	HMAC-SHA-256	HMAC SHA-256 key	R/E
CO	Show Status	None	N/A
CO	Perform Self Tests	AES data encryption key HMAC integrity check key	R,E R,E

Table 5. Cryptographic Officer – Roles and Services

Role	*Authorized Services*	*Cryptographic Keys and CSPs*	*Access Type*
32/64-bit sub component			
User	AES Encrypt / Decrypt	AES data encryption key	R,E
User	RSA Sign / Verify	RSA private and public key	R,E
User	RSA Key Generation	RSA private and public key	W
User	RSA Key Wrapping	RSA private and public key	R,E
User	SHA-1	None	R,E
User	SHA-256 Hash	None	N/A
User	HMAC-SHA-256	HMAC SHA-256 key	R/E
User	Show Status	None	N/A
User	Perform Self Tests	AES data encryption key HMAC integrity check key RSA public and private key RNG seed and seed key	R,E R,E R,E R,E

Role	*Authorized Services*	*Cryptographic Keys and CSPs*	*Access Type*
16-bit sub component			
User	AES Encrypt / Decrypt	AES data encryption key	R,E
User	SHA-256 Hash	None	R,E
User	HMAC-SHA-256	HMAC SHA-256 key	R,E
User	Show Status	None	N/A
User	Perform Self Tests	AES data encryption key HMAC integrity check key	R,E R,E

Table 6. User – Roles and Services

Role	*Non-authorized Services*	*Cryptographic Keys and CSPs*	*Access Type*
32/64-bit sub component			
User, CO	RSA Encrypt / Decrypt	N/A	N/A
User, CO	MD5 Hash	N/A	N/A
User, CO	RC2 Encrypt / Decrypt	N/A	N/A
User, CO	Triple-DES Encrypt / Decrypt	N/A	N/A
User, CO	DES Encrypt / Decrypt	N/A	N/A

Table 7. CO and User – Non-Authorised Roles and Services

5 PHYSICAL SECURITY

This section is not applicable as this is a software module.

6 CRYPTOGRAPHIC KEY MANAGEMENT

The following table identifies the Cryptographic Keys and Critical Security Parameters (CSPs) employed within the module.

Key	Generation	Storage	Use	Role
32/64-bit sub component				
Data Encryption Key AES	Generated internally using a RNG compliant to ANSI X9.31.	Key not stored	Encryption Key used to encrypt data	User CO
RNG seed	Generated externally drawn from an entropy pool.	Key not stored	Used to generate random numbers during disk key creation in FIPS mode.	User CO

RNG seed key	Pre-loaded during the manufacturing process.	Compiled in the binary	Used to generate random numbers during disk key creation in FIPS mode.	User CO
Integrity Check Key HMACSHA256	Pre-loaded during the manufacturing process	Compiled in the binary	Private key for approved integrity technique for checking the integrity of cryptographic module binaries.	User CO
RSA private and public key	Generated internally using the approved RSA key generation method	Key not stored	Used for approved key transport or calculation and verification of digital signatures	User CO
16-bit sub component				
Data Encryption Key AES	Generated internally using a RNG compliant to ANSI X9.31.	Key not stored	Encryption Key used to encrypt data	User CO
Integrity Check Key HMACSHA256	Pre-loaded during the manufacturing process	Compiled in the binary	Private key for approved integrity technique for checking the integrity of cryptographic module binaries.	User CO

Table 8. Cryptographic keys and CSPs

The module keys map to the following algorithms certificates:

Approved or allowed Security Functions	***Certificate***
32/64-bit implementation	
Symmetric Key Encryption	
AES **ECB, CBC and OFB (FIPS PUB 197) : 128, 256**	#1088
HASHING	
SHA-1 and SHA-256 byte-oriented	#1021
HMAC-SHA-256	#611
Asymmetric Keys	
RSA **ANSI X9.31 (Sig Gen / Sig Ver) : 1024, 2048** **ANSI X9.31 (Key Generation) : 1024, 2048**	#513
RSA Key Wrapping	(allowed in FIPS mode, see caveat below)
Random Number Generation (ANSI X9.31)	#610
16-bit implementation	
Symmetric Key Encryption	
AES **ECB (FIPS PUB 197) : 128, 256**	#1087

Approved or allowed Security Functions	*Certificate*
HASHING	
SHA-256 byte-oriented	#1020
HMAC-SHA-256	#610
Non-Approved Security Function	
32/64-bit implementation	
MD5	N/A
RC2	N/A
DES	N/A
Triple-DES	non-compliant
RSA encrypt / decrypt	N/A

Table 9. FIPS Approved Algorithms Table

Please Note:

- Only authorized services shall be called when operating in FIPS mode.
- RNGCreateKey() is the only function implemented by the module that can be used for random number generation in FIPS approved mode.
- All keys generated by the approved Random Number Generation algorithm must be outputted from the module in encrypted format using AES
- RSA (key wrapping; key establishment methodology provides between 80 and 112 bits of encryption strength)

7 SELF-TEST

The cryptographic module will perform the following power up tests that can be either initiated by re-starting the cryptographic module, or initiated on demand by calling the appropriate API initialization function as described in section 8.1.

1. Known answer tests for cryptographic algorithms
2. Software Integrity (using an Approved Algorithm) tests on all components in the cryptographic boundary

Cryptographic Algorithm KATs:
The module will perform Known Answer Tests (KAT) on the cryptographic algorithms implemented. The known answer tests will be performed before the algorithms are utilized.

Known Answer Tests (KATs) are run at power-up for:

- 16-bit sub component : AES KAT encryption and decryption (ECB mode), SHA256 KAT
- 32/64-bit sub component : AES KAT encryption and decryption (ECB mode), RSA signature generation and verification KAT, HMAC KAT, RNG KAT, SHA1 KAT, SHA256 is tested as part of the HMAC software integrity test on the module

Conditional RNG Test: The test is run each time an Approved RNG generates a random number. The test involves comparing the generated value with the previously generated value.

The following conditional test is carried out:

- 32/64-bit sub component : RNG test failure to a constant value

Conditional Pairwise Consistency Test: The test is run each time key pairs are generated using the Approved RSA Key Generation service. Different tests are run depending on whether the keys will be used to perform an approved key transport method or if they are used to perform the calculation and verification of a digital signature.

- if the keys are to be used for an approved key transport method. The keys are tested by encrypting a plaintext value and comparing the result with the original plaintext value.
- If the keys are used to perform the calculation and verification of digital signatures. The consistency of the keys is tested by the calculation and verification of a digital signature.

Software/Firmware Integrity Tests: The module checks the integrity of its various components (16-bit sub component binaries and 32/64-bit sub component binaries) using HMAC-SHA-256. The software/firmware integrity tests are completed individually over the 16-bit and 32/64-bit sub-components. i.e., each sub component implements its own integrity test.

8 Crypto-Officer and User Guidance

This section describes the configuration, maintenance, and administration of the cryptographic module.

8.1 Secure Setup and Initialization

The module must be validated and initialized by successful completion of all self tests as documented in section 7 above. The self tests must be performed as part of the module initialization and can also be performed on demand by either COs or Users.

The steps to securely initialize the module are as follows:

- install the module on the target platform
- initialize the module using fipsSelfTests(). Operators must confirm that the status returned is '0x0' and there are no errors reported.
- after successful initialization, users may operate the module and access the FIPS Approved cryptographic services implemented by the module. In order to run in FIPS Approved mode the module must use FIPS approved cryptography.

8.2 Module Security Policy Rules

COs must observe the following practice:

- when performing key generation services, COs are required to use a data source with high entropy for generating the seed value to be provided to the Approved RNG algorithm.
- the module is operating in FIPS mode when using FIPS Approved cryptography.
- keys that are entered or output from the cryptographic boundary must be encrypted using AES .

8.3 Zeroization

All keys in the module are zeroized when the module is uninstalled by formatting the hard drive on the PC.