Cloakware, Inc.

Cloakware Security Kernel

Software Version: 1.0

FIPS 140-2 Non-Proprietary Security Policy

FIPS Security Level: 1
Document Version: 1.0

Prepared for:

Cloakware, Inc.
8219 Leesburg Pike, Suite 350
Vienna, VA 22182-2656

Phone: (703) 752-4830

www.cloakware.com

Prepared by:

Corsec Security, Inc.
10340 Democracy Lane, Suite 201
Fairfax, VA 22030

Phone: (703) 267-6050
Email: info@corsec.com

http://www.corsec.com

Table of Contents

1 INTRODUCTION 4
1.1 PURPOSE 4
1.2 REFERENCES 4
1.3 DOCUMENT ORGANIZATION 4

2 CLOAKWARE SECURITY KERNEL 5
2.1 OVERVIEW 5
2.2 CRYPTOGRAPHIC BOUNDARY 6
2.2.1 Physical Cryptographic Boundary 6
2.2.2 Logical Cryptographic Boundary 7
2.3 MODULE INTERFACES 8
2.4 ROLES AND SERVICES 9
2.4.1 Crypto-Officer Role 9
2.4.2 User Role 10
2.5 PHYSICAL SECURITY 11
2.6 OPERATIONAL ENVIRONMENT 11
2.7 CRYPTOGRAPHIC KEY MANAGEMENT 11
2.7.1 Key Generation 15
2.7.2 Key Entry and Output 15
2.7.3 CSP Storage and Zeroization 15
2.8 EMI/EMC 15
2.9 SELF-TESTS 15
2.10 DESIGN ASSURANCE 16
2.11 MITIGATION OF OTHER ATTACKS 16

3 SECURE OPERATION 17
3.1 INITIAL SETUP 17
3.2 CRYPTO-OFFICER GUIDANCE 17
3.2.1 Installation 17
3.2.2 Management 17
3.3 USER GUIDANCE 17

4 ACRONYMS 18

APPENDIX A 20

Table of Figures

FIGURE 1 – TYPICAL DEPLOYMENT OF CLOAKWARE PASSWORD AUTHORITY 5
FIGURE 2 – STANDARD SERVER BLOCK DIAGRAM 7
FIGURE 3 – LOGICAL BLOCK DIAGRAM AND CRYPTOGRAPHIC BOUNDARY 8

List of Tables

TABLE 1 – FIPS 140-2 SECURITY LEVELS 6
TABLE 2 – FIPS INTERFACE MAPPINGS 9
TABLE 3 – MAPPING OF CRYPTO OFFICER ROLE'S SERVICES TO TYPE OF ACCESS 10
TABLE 4 – MAPPING OF USER ROLE'S SERVICES TO TYPE OF ACCESS 10
TABLE 5 – FIPS-APPROVED ALGORITHM IMPLEMENTATIONS 12

TABLE 6 – LIST OF CRYPTOGRAPHIC KEYS, CRYPTOGRAPHIC KEY COMPONENTS, AND CSPS 13
TABLE 7 – ACRONYMS 18
TABLE 8 – POWER-UP SELF-TESTS ERROR CODES 20
TABLE 9 – CONDITIONAL SELF-TESTS ERROR CODES 20

1 Introduction

1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the Cloakware Security Kernel from Cloakware, Inc. This Security Policy describes how the Cloakware 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 Cryptographic Module Validation Program (CMVP) website, which is maintained by National Institute of Standards and Technology (NIST) and Communication Security Establishment Canada (CSEC): http://csrc.nist.gov/groups/STM/index.html.

The Cloakware Security Kernel is referred to in this document as the cryptographic module, or the module.

1.2 References

This document deals only with 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 Cloakware website (http://www.cloakware.com) contains information on the full line of products from Cloakware.
- The CMVP Vendor List (http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401vend.htm) 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
- Submission Summary
- Other supporting documentation and additional references

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

2 Cloakware Security Kernel

2.1 Overview

Cloakware, Inc. a subsidiary of Irdeto Access B.V is the leading provider of software technology solutions for securing business applications and digital assets in enterprise, consumer, and government markets. The company offers two main business lines of product: Datacenter solutions and Consumer Product solutions. The Datacenter solutions enable organizations to meet governance, risk management, and compliance objectives for password management; and Cloakware Consumer Product solutions to protect software and content on personal computers, set-top boxes, mobile phones, and media players. Its Datacenter solutions comprise Cloakware Password Authority (CPA), a password management solution to automate the life-cycle management of passwords. CPA manages passwords on "target" systems. Targets are defined as remote systems with privileged accounts (such as operating system root accounts on servers, administrator accounts for routers, and accounts on Storage Area Networks (SANs), backup systems, databases, etc.).
Cloakware uses White-box Cryptography to store password securely. Cloakware's White-box Cryptography implements cryptographic algorithms in such a way that it hides the critical data and keys in environments where hackers can observe everything in cryptographic operations. Figure 1 below shows the typical deployment of a Cloakware Password Authority Server.

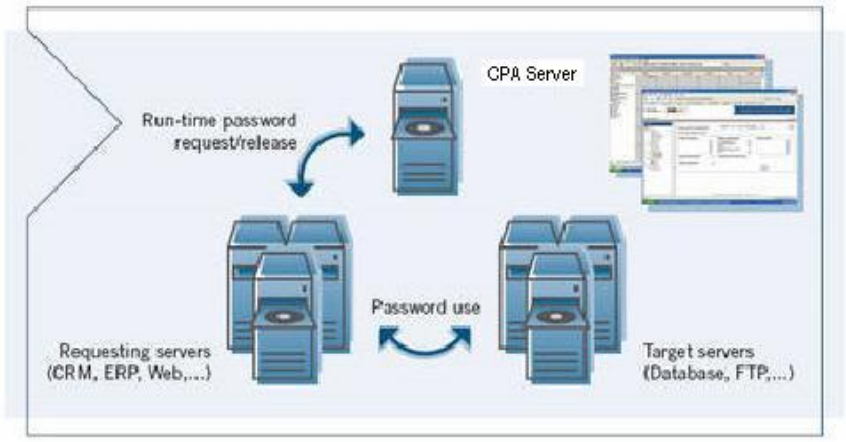

Figure 1 – Typical deployment of Cloakware Password Authority

The CPA product uses a three-tiered architecture, a data storage layer, application layer, and interface layer. The data storage layer provides a centralized database for storage of managed password information (the credentials repository), and can use any vendor's database such as Oracle, Sybase, DB2, MySQL, Microsoft SQL Server, to store CPA information. The application layer implements the business logic in the CPA Server, and regulates access by human administrators and automated requestors (e.g. applications, pearl scripts, and database front-ends). The interface layer allows Cloakware to offer redundancy and load-balancing in accessing the CPA Server.

A pluggable authentication module architecture enables methods for authenticating CPA administrators, including identity (ID) and password, and with the extensibility to add other proprietary methods. Only authenticated and authorized Servers and applications are able to request application IDs and passwords. CPA goes far beyond OS-level authentication by including checks for the executing application's ID and location, tamper detection, and unique keying material per Server. Unattended Servers no longer need hard-coded credentials to access other Servers.

The Cloakware Password Authority implements several management interfaces: a web-based graphical user interface (GUI), command-line interface (CLI) and Java Native Interface (C/C++ Application Programming Interface calls). The client and server communicate securely over by using Transport Layer Security (TLS). The CPA stores administrator and application IDs and passwords in a White Box AES-encrypted repository.

All of the products' cryptographic functionality is provided by the Cloakware Security Kernel, which is composed of functionality contained in a single library. The module is a multi-chip standalone cryptographic module evaluated for use on a standard server platform running one of the following operating systems (OSs):

- Red Hat Enterprise Linux (RHEL) AS 5.0
- Windows Server 2008
- Solaris 10

The Cloakware Security Kernel is validated at the following FIPS 140-2 Section levels:

Table 1 – FIPS 140-2 Security Levels

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[1]	1
9	Self-tests	1
10	Design Assurance	1
11	Mitigation of Other Attacks	N/A
14	Cryptographic Module Security Policy	1

2.2 Cryptographic Boundary

The following sections will define the physical and logical boundary of the Cloakware Security Kernel.

2.2.1 Physical Cryptographic Boundary

As a software cryptographic module, there are no physical protection mechanisms implemented; the module must rely on the physical characteristics of the host server. The physical cryptographic boundary of the Cloakware Security Kernel is defined by the hard metal enclosure around the computer on which it runs. The module supports the physical interfaces of a standard server. The physical interfaces include the mouse and keyboard ports, optical drives, floppy disk, serial ports, parallel ports, networks ports, monitor port, and power plug. See Figure 2 for a standard server block diagram.

[1] EMI/EMC – Electromagnetic Interference / Electromagnetic Compatibility

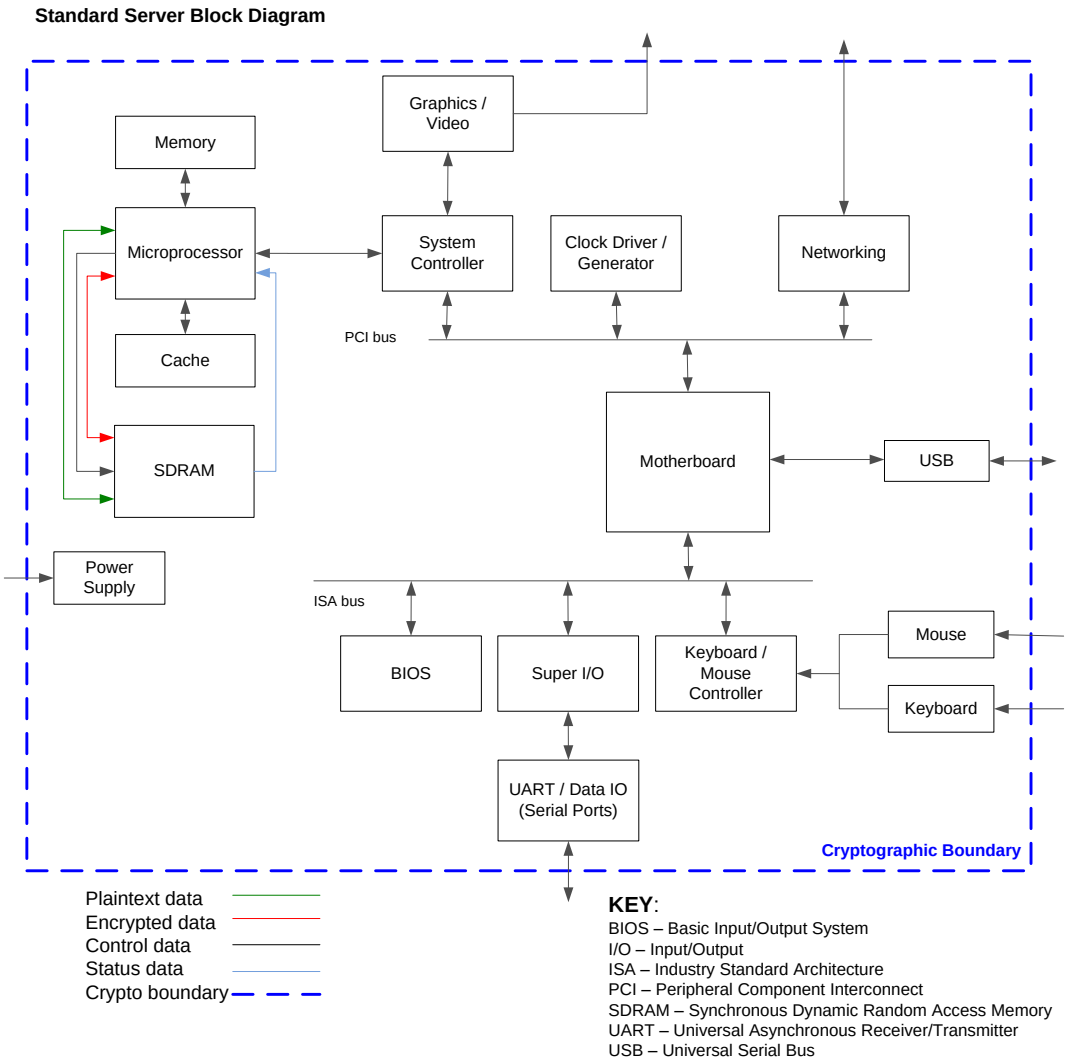

Figure 2 – Standard Server Block Diagram

2.2.2 Logical Cryptographic Boundary

Figure 3 below shows a logical block diagram of the module. The module is a software cryptographic module running on Red Hat Enterprise Linux (RHEL) AS 5.0 (libcwcrypto.a), Windows Server 2008 (cwcrypto.lib), or Solaris 10 (libcwcrypto.a). The module's logical cryptographic boundary encompasses all functionality contained within a single library. This single library is linked at run-time to a C/C++ library, which can be called by host applications to provide cryptographic services.

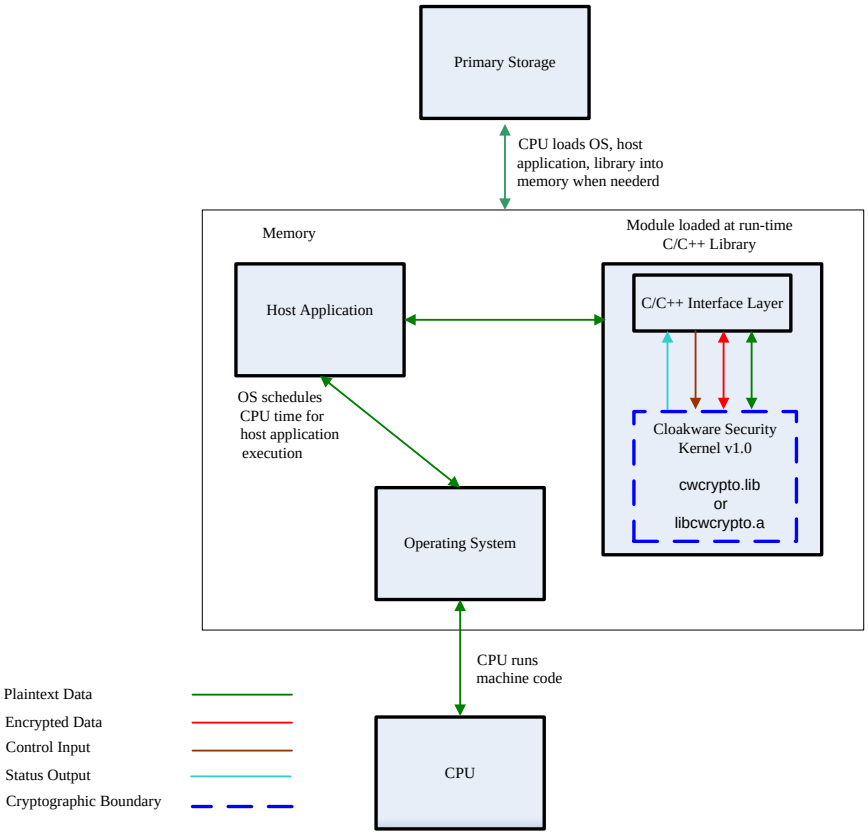

Figure 3 – Logical Block Diagram and Cryptographic Boundary

2.3 Module Interfaces

The module's logical interfaces exist at a low level in the software as an Application Programming Interface (API). The API interface is mapped to the following four logical interfaces:

- Data input
- Data output
- Control input
- Status output

The module features the physical ports of the host server, as depicted in Figure 2. The following is a list of physical interfaces implemented on a host server:

- Keyboard port
- Network ports
- Mouse port
- Display monitor port
- CD-ROM[2] drive
- LED[3] indicators

[2] CD-ROM – Compact Disc – Read-Only Memory

[3] LED – Light-Emitting Diode

- Floppy disk
- Power plug/adapter
- Serial ports
- Power switch
- USB ports
- Parallel ports

As a software module, the module has no physical characteristics. Thus, the module's manual controls, physical indicators, and physical and electrical characteristics are those of the host server.

The FIPS-defined interfaces map to their physical and logical counterparts as described in Table 2 below.

Table 2 – FIPS Interface Mappings

Logical Interface	Physical Interface Mapping (Standard Server)	Module Mapping
Data Input Interface	Keyboard, mouse, CD-ROM, floppy disk, and serial/USB/parallel/network ports	The API calls that accept input data for processing through their arguments.
Data Output Interface	Floppy disk, monitor, and serial/USB/parallel/network ports	The API calls that return by means of their return codes or arguments generated or processed data back to the caller.
Control Input Interface	Keyboard, CD-ROM, floppy disk, mouse, and serial/USB/parallel/network port	The API calls that are used to initialize and control the operation of the module.
Status Output Interface	Floppy disk, monitor, and serial/USB/parallel/network ports	Return values for API calls.
Power Interface	Power Switch	Not Applicable

2.4 Roles and Services

While the module itself provides no mechanism for the authentication of operators, it supports the following authorized roles: the Crypto-Officer (CO) role and the User role. All operators assume roles implicitly through the execution of APIs.

Note: The following definitions are used in the "CSP and Type of Access" column in Table 3 and Table 4.

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

2.4.1 Crypto-Officer Role

The Crypto-Officer role is used for accessing the module's symmetric encryption/decryption, signature generation/verification, hashing, cryptographic key generation, random number generation, and message authentication functions.

Descriptions of the services available to the Crypto-Officer role are provided in Table 3 below.

Table 3 – Mapping of Crypto Officer Role's Services to Type of Access

Service	Description	CSP and Type of Access
White Box AES (WBAES) encryption/decryption	Encryption/decryption operation	Symmetric key AES – Read, Write, Execute
Hashing (SHS)	Hashing operation	None
Message authentication (HMAC[4])	Message authentication services	HMAC SHA1 key – Read, Write, Execute
Symmetric encryption/decryption	Encryption/decryption operation	Symmetric key AES, TDES – Read, Write, Execute
Symmetric key generation	Generating symmetric keys	Symmetric key AES, TDES – Read, Write, Execute
Digital Signature	Sign and verify operation	Asymmetric private key RSA, DSA – Read, Write, Execute
Key transport	Key transport mechanism	Asymmetric private key RSA – Read, Write, Execute
Asymmetric key generation	Generating asymmetric keys	Asymmetric private key RSA, DSA – Read, Write, Execute
Pseudo-random Number Generation	Generates random numbers	Seed key Seed AES – Read, Write, Execute
Show status	Show status of the module	None Execute
Perform Self-Tests	Perform self-tests on demand by rebooting the device	None Execute
Zeroization	Zeroize all CSPs	Symmetric keys, asymmetric keys, HMAC-SHA-1 key, seed key – Write

2.4.2 User Role

Like the CO role, the User role is used to access symmetric encryption/decryption, signature generation/verification, hashing, cryptographic key generation, random number generation, and message authentication functions.

Descriptions of the services available to the User role are provided in Table 4.

Table 4 – Mapping of User Role's Services to Type of Access

Service	Description	CSP and Type of Access
White Box AES (WBAES) encryption/decryption	Encryption/decryption operation	Symmetric key AES – Read, Write, Execute
Hashing (SHS)	Hashing operation	None

[4] HMAC – Hash Message Authentication Code

Service	Description	CSP and Type of Access
Message authentication (HMAC)	Message authentication services	HMAC SHA1key – Read, Write, Execute
Symmetric encryption/decryption	Encryption/decryption operation	Symmetric key AES, TDES – Read, Write, Execute
Symmetric key generation	Generating symmetric keys	Symmetric key AES, TDES – Read, Write, Execute
Digital Signature	Sign and verify operation	Asymmetric private key RSA, DSA – Read, Write, Execute
Key transport	Key transport mechanism	Asymmetric private key RSA – Read, Write, Execute
Asymmetric key generation	Generating asymmetric keys	Asymmetric private key RSA, DSA – Read, Write, Execute
Pseudo-random Number Generation	Generates random numbers	Seed key Seed AES – Read, Write, Execute
Show status	Show status of the module	None Execute
Perform Self-Tests	Perform self-tests on demand by rebooting the device	None Execute
Zeroization	Zeroize all CSPs	Symmetric keys, asymmetric keys, HMAC-SHA-1 key, seed key – Write

2.5 Physical Security

The Cloakware Security Kernel is purely a software module. As such, it depends on the physical characteristics of the host server and its protection mechanisms. Thus, physical security requirements do not apply.

2.6 Operational Environment

The module was tested for FIPS 140-2 validation on Red Hat Enterprise Linux (RHEL) AS 5.0, Windows Server 2008 and Solaris 10 operating system. For FIPS 140-2 compliance, this is considered to be single user operating system. As such, all keys, intermediate values, and other CSPs remain only in the process space of the operator using the module. The operating system uses its native memory management mechanisms to ensure that outside processes cannot access the process space used by the module.

2.7 Cryptographic Key Management

The module implements the FIPS-Approved algorithms listed in Table 5 below.

Table 5 – FIPS-Approved Algorithm Implementations

Approved Security Function	Certificate Number
Symmetric Key Algorithm	
AES - 128-,192-, 256-bit in ECB[5], CBC[6], CFB[7]8, CFB128 and OFB[8] modes	1309
White Box AES – 128, 256-bit in CBC and ECB modes	1306
Triple-DES - 168-bit in ECB, CBC, CFB8 and OFB modes	914
Secure Hashing Algorithm (SHA)	
SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	1197
Message Authentication Code (MAC) Function	
HMAC using SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	761
Pseudo Random Number Generator (PRNG)	
ANSI[9] X9.31 Appendix A.2.4 PRNG[10]	731
Asymmetric Key Algorithm	
RSA (X9.31, PKCS #1.5, PSS) sign/verify: 1024-, 1536-, 2048-, 3072-, 4096-bit	663
DSA sign/verify: 1024- bit	441

Additionally, the module utilizes the following non-FIPS-approved algorithm implementations (these algorithms are allowed for use in a FIPS-approved mode of operation):

- Diffie-Hellman support for key agreement: 1024-, 2048-, 3072-, 7680-, 15360-bits for key establishment (Caveat: provides between 80 and 256 bits of encryption strength)

  NOTE: The module does not provide full Diffie-Hellman key agreement, only the Diffie-Hellman algorithm/functionality and primitives.

- RSA key transport: 1024-, 2048-, 3072-, 7680-, 15360-bits (key wrapping; key establishment methodology provides between 80 and 256 bits of encryption strength)

[5] ECB – Electronic Codebook
[6] CBC – Cipher-Block Chaining
[7] CFB – Cipher Feedback
[8] OFB – Output Feedback
[9] ANSI – American National Standards Institute
[10] PRNG – Pseudo Random Number Generator

The module supports the following critical security parameters in Table 6 below. It should be noted that the module key paths are being interpreted being input and output "INT" paths as defined in IG 7.7. Therefore, key establishment mechanism is not applicable for this particular module since all input and output are occurring within the physical boundary of the host server.

Table 6 – List of Cryptographic Keys, Cryptographic Key Components, and CSPs

Key	Key Type	Key Strength	Generation / Input	FIPS Approved establishment mechanism	Output	Storage	Zeroization	Use
White Box (WBAES) keys (16 keys)	AES - CBC, ECB - 128-, 256-bit Standard Fixed Standard Dynamic	128, 256-bit	Generated Internally	Not Applicable	None	Resides in volatile memory using the code transformation technique	By power cycle or API service termination	Encrypt/decrypt stored password, TLS session keys
Symmetric key	AES - 128-, 192-, 256-bit Triple DES - 168-bit	128, 192, 256-bit	Generated Internally using the FIPS Approved PRNG	Not Applicable	None	Resides in plaintext on volatile memory	By power cycle or Zeroization Service	Encrypt/decrypt
Asymmetric public key	RSA 1024-,1536-, 2048-, 3072- and 4096-bit public key	Between 80 to 150	Generated Internally using the FIPS Approved PRNG	Not Applicable	None	Resides in plaintext on volatile memory	By power cycle or Zeroization Service	Key wrapping, Signature Verification
Asymmetric private key	RSA 1024-,1536-, 2048-, 3072- and 4096-bit private key	Between 80 to 150	Generated Internally using the FIPS Approved PRNG	Not Applicable	None	Resides in plaintext on volatile memory	By power cycle or Zeroization Service	Key Unwrapping, Signature Generation
DSA public key	DSA 1024-bit	80	Generated Internally using the FIPS Approved PRNG	Not Applicable	None	Resides in plaintext on volatile memory	By power cycle or Zeroization Service	Signature Generation

Key	Key Type	Key Strength	Generation / Input	FIPS Approved establishment mechanism	Output	Storage	Zeroization	Use
DSA private key	DSA 1024-bit	80	Generated Internally using the FIPS Approved PRNG	Not Applicable	None	Resides in plaintext on volatile memory	By power cycle or Zeroization Service	Signature Verification
HMAC-SHA-1 key	HMAC SHA-1	80	Generated Internally using the FIPS Approved PRNG	Not Applicable	None	Resides in plaintext on volatile memory	By power cycle or Zeroization Service	Generate Message Authentication Code (MAC)
ANSI X9.31 PRNG seed	16 bytes	Not Applicable	None	Not Applicable	None	Resides in plaintext on volatile memory	By power cycle or Zeroization Service	Generate random number
ANSI X9.31 PRNG seed key	16-, 24-, or 32-byte AES key	128, 192 and 256	None	Not Applicable	None	Resides in plaintext on volatile memory	By power cycle or Zeroization Service	Generate random number

2.7.1 Key Generation

The module uses an ANSI X9.31 Appendix A.2.4 PRNG implementation to generate cryptographic keys. This PRNG is FIPS-Approved as shown in Annex C to FIPS PUB 140-2.

2.7.2 Key Entry and Output

The cryptographic module itself does not support key entry or key output from its physical boundary. Naturally however, keys are passed to the module as parameters from applications resident on the host platform via the exposed APIs. The host application using the module is responsible for ensuring that the input or output of secret and private keys is accomplished in encrypted form.

2.7.3 CSP Storage and Zeroization

The module does not persistently store any CSPs. The White-box AES keys in Table 6 above reside only on the volatile memory using the code transformation technique and can be zeroized by termination of the API or power cycling the module. All of the other keys and CSPs in Table 6 reside only on the volatile memory in plaintext and can be zeroized using the destruction method included in the API. This method is capable of overwriting the key in memory with random bytes.

2.8 EMI/EMC

The module is a software module, and depends on the host server for its physical characteristics. However, the host server have been tested for, and meet, applicable Federal Communications Commission (FCC) EMI and EMC requirements for business use as defined in Subpart B, Class A of FCC 47 Code of Federal Regulations Part 15. All systems sold in the United States must meet the applicable FCC requirements.

2.9 Self-Tests

The module performs the following self-tests at power-up:

- Software integrity test using HMAC SHA-1
- Cryptographic Algorithm tests
    - White Box AES (WBAES) KAT
    - AES Known Answer Test (KAT)
    - Triple-DES KAT
    - HMAC KATs
        - HMAC SHA-1
        - HMAC SHA-224
        - HMAC SHA-256
        - HMAC SHA-384
        - HMAC SHA-512
    - SHA-1 KAT
        - SHA-1 (performed as part of the HMAC-SHA-1 known answer test)
        - SHA-224 (performed as part of the HMAC-SHA-224 known answer test)
        - SHA-256 (performed as part of the HMAC-SHA-256 known answer test)
        - SHA-384 (performed as part of the HMAC-SHA-384 known answer test)
        - SHA-512 (performed as part of the HMAC-SHA-512 known answer test)
    - RSA encrypt/decrypt KAT
    - RSA signature generation/verification KAT
    - DSA pairwise consistency test for sign/verify
    - ANSI X9.31 PRNG Appendix A.2.4 KAT

The module performs the following conditional self-tests:

- Continuous Random Number Generator test (CRNGT)

- DSA pairwise consistency test for sign/verify
- RSA pairwise consistency test for sign/verify and encrypt/decrypt

If any of the power-up self-tests fails, the module returns an error code (FIPS_R_SELFTEST_FAILED). An internal global error flag is also set and subsequently tested to prevent invocation of any cryptographic function calls by entering into a critical error state. No data output or cryptographic operations are possible when the module enters the critical error state. A power-up self-test error can only be cleared by restarting the module.

Failure of a conditional self-test transitions the module to a soft error state. After providing error status, the module will terminate the API call and will return the control back to the host application. No data output or cryptographic operations are possible when the module enters the soft error state.

2.10 Design Assurance

Cloakware uses Concurrent Versions System (CVS) as the configuration management system. It records the history of the source files. It stores all the versions of a file in a single file in a clever way that only stores the differences between versions. It also helps as a part of a group of people working on the same project by insulating everyone from each other. Every person works in his own directory, and CVS merges the work when each person is done.

Additionally, Microsoft Visual SourceSafe (VSS) version 6.0 is used to provide configuration management for the Cloakware Security Kernel's FIPS documentation. This software provides access control, versioning, and logging.

2.11 Mitigation of Other Attacks

The module does not claim to mitigate any additional attacks in an approved FIPS mode of operation.

3 Secure Operation

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

3.1 Initial Setup

It is the Crypto-Officer's responsibility to configure the module according to FIPS-Approved mode. The Crypto-Officer should make sure that the Operating System (OS) is configured to a Single User mode of operation. An operator can access the application linked to the module through a web-based graphical user interface (GUI) or a command-line interface (CLI).

The sections below describe how to install and manage the module in FIPS-Approved mode of operation and how to make secure calls.

3.2 Crypto-Officer Guidance

The following two sections contain the necessary guidance to securely install and administer the cryptographic module.

3.2.1 Installation

The software module will be provided as a binary to the Crypto-Officer by Cloakware, Inc. The module is installed during the process of installing the host application. With the delivered software, the Crypto-Officer also receives detailed documentation on installing, uninstalling, configuring, managing and upgrading the host application.

3.2.2 Management

The Crypto-Officer is responsible for making sure the module is running properly in FIPS-Approved mode of operation. The Crypto-Officer is able to monitor and configure the module through a web-based graphical user interface (GUI) or a command-line interface (CLI). Detailed instructions to manage and troubleshoot the host application are provided in the Administrator's Guide. The Crypto-Officer should monitor the module status regularly for normal operation.

3.3 User Guidance

The cryptographic functionality of the module (i.e. the collection of User role services) is listed in Table 4 above.

4 Acronyms

This section describes the acronyms used throughout the document.

Table 7 – Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
ANSI	American National Standards Institute
API	Application Programming Interface
CBC	Cipher-Block Chaining
CFB	Cipher Feedback
CMVP	Cryptographic Module Validation Program
CRNGT	Continuous Random Number Generator Test
CSEC	Communications Security Establishment Canada
CSP	Critical Security Parameter
CVS	Concurrent Version System
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
ECB	Electronic Codebook
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FCC	Federal Communication Commission
FIPS	Federal Information Processing Standard
HMAC	(Keyed-) Hash Message Authentication Code
KAT	Known Answer Test
LED	Light Emitting Diode
MAC	Message Authentication Code
NIST	National Institute of Standards and Technology
NVLAP	National Voluntary Laboratory Accreditation Program
OFB	Output Feedback
OS	Operating System
PC	Personal Computer
PKCS	Public Key Cryptography Standard
PRNG	Pseudo Random Number Generator
PSS	Probabilistic Signature Scheme
RAM	Random Access Memory

Acronym	Definition
RSA	Rivest Shamir and Adleman
SHA	Secure Hash Algorithm
USB	Universal Serial Bus
VSS	Visual SourceSafe
WBAES	White-box Advanced Encryption Standard

Appendix A

The following two tables provide the possible error codes associated with FIPS Self-Test failures.

Table 8 – Power-Up Self-Tests Error Codes

Power-Up Self-Test	Error Code
Firmware Integrity Test	FIPS_F_FIPS_CHECK_INCORE_FINGERPRINT, FIPS_R_UNSUPPORTED_PLATFORM, FIPS_R_FINGERPRINT_DOES_NOT_MATCH, FIPS_R_FINGERPRINT_DOES_NOT_MATCH_SEGMENT_ALIASING, FIPS_R_FINGERPRINT_DOES_NOT_MATCH_NONPIC_RELOCATED
WBAES KAT	AES_Standard_FixedKey_Encrypt_ECB_16_Gamma returned AES_Standard_FixedKey_Decrypt_ECB_16_Gamma returned AES_Standard_FixedKey_Encrypt_ECB_32_Gamma returned AES_Standard_FixedKey_Decrypt_ECB_32_Gamma returned AES_Standard_FixedKey_Encrypt_CBC_16_Gamma returned AES_Standard_FixedKey_Decrypt_CBC_16_Gamma returned AES_Standard_FixedKey_Encrypt_CBC_32_Gamma returned AES_Standard_FixedKey_Decrypt_CBC_32_Gamma returned AES_Standard_DynamicKey_Encrypt_ECB_16_Gamma returned AES_Standard_DynamicKey_Decrypt_ECB_16_Gamma returned AES_Standard_DynamicKey_Encrypt_ECB_32_Gamma returned AES_Standard_DynamicKey_Decrypt_ECB_32_Gamma returned AES_Standard_DynamicKey_Encrypt_CBC_16_Gamma returned AES_Standard_DynamicKey_Decrypt_CBC_16_Gamma returned AES_Standard_DynamicKey_Encrypt_CBC_32_Gamma returned AES_Standard_DynamicKey_Decrypt_CBC_32_Gamma returned
AES Known Answer Test (KAT)	FIPS_F_FIPS_SELFTEST_AES, FIPS_R_SELFTEST_FAILED
TDES KAT	FIPS_F_FIPS_SELFTEST_DES, FIPS_R_SELFTEST_FAILED
HMAC KATs	FIPS_F_FIPS_SELFTEST_HMAC, FIPS_R_SELFTEST_FAILED
SHA-1 KATs	FIPS_F_FIPS_SELFTEST_SHA, FIPS_R_SELFTEST_FAILED
RSA KAT (encrypt/decrypt and sign/verify)	FIPS_F_FIPS_SELFTEST_RSA, FIPS_R_SELFTEST_FAILED
DSA pairwise consistency test	FIPS_F_FIPS_SELFTEST_DSA, FIPS_R_SELFTEST_FAILED
PRNG KAT	FIPS_F_FIPS_SELFTEST_RNG, FIPS_R_SELFTEST_FAILED

Table 9 – Conditional Self-Tests Error Codes

Conditional Self-Test	Error Code
DSA pairwise consistency test for sign/verify	FIPS_F_FIPS_CHECK_DSA, FIPS_R_PAIRWISE_TEST_FAILED
Continuous RNG Test	RAND_F_FIPS_RAND, RAND_R_PRNG_ERROR, RAND_F_FIPS_RAND, RAND_R_PRNG_STUCK

Conditional Self-Test	Error Code
RSA pairwise consistency test for sign/verify	FIPS_F_FIPS_CHECK_RSA, FIPS_R_PAIRWISE_TEST_FAILED
RSA pairwise consistency test for encrypt/decrypt	FIPS_F_FIPS_CHECK_RSA, FIPS_R_PAIRWISE_TEST_FAILED

Prepared by:
Corsec Security, Inc.

10340 Democracy Lane, Suite 201
Fairfax, VA 22030

Phone: (703) 267-6050
Email: info@corsec.com
http://www.corsec.com