

Websense, Inc.

Websense Crypto Module Java

Software Version: 1.0

FIPS 140-2 Non-Proprietary Security Policy

FIPS Security Level: 1
Document Version: 0.9



Prepared for:



Websense, Inc.
10240 Sorrento Valley Road
San Diego, California 92121
United States of America

Phone: +1 800-723-1166
Email: info@websense.com
<http://www.websense.com>

Prepared by:



Corsec Security, Inc.
13135 Lee Jackson Memorial Highway, Suite 220
Fairfax, Virginia 22033
United States of America

Phone: +1 703-267-6050
Email: info@corsec.com
<http://www.corsec.com>

Table of Contents

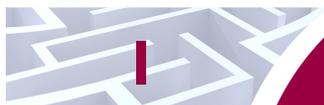
I	INTRODUCTION	3
1.1	PURPOSE	3
1.2	REFERENCES	3
1.3	SUBMISSION PACKAGE ORGANIZATION.....	3
2	WEBSense CRYPTO MODULE JAVA	4
2.1	OVERVIEW.....	4
2.2	MODULE SPECIFICATION.....	5
2.2.1	<i>Physical Cryptographic Boundary</i>	5
2.2.2	<i>Logical Cryptographic Boundary</i>	6
2.3	MODULE INTERFACES	7
2.4	ROLES AND SERVICES.....	8
2.4.1	<i>Crypto Officer Role</i>	8
2.4.2	<i>User Role</i>	9
2.4.3	<i>Authentication</i>	10
2.5	PHYSICAL SECURITY	10
2.6	OPERATIONAL ENVIRONMENT.....	10
2.7	CRYPTOGRAPHIC KEY MANAGEMENT	10
2.7.1	<i>Key Generation</i>	13
2.7.2	<i>Key Entry and Output</i>	13
2.7.3	<i>Key/CSP Storage and Zeroization</i>	13
2.8	EMI/EMC	13
2.9	SELF-TESTS	13
2.9.1	<i>Power-Up Self-Tests</i>	13
2.9.2	<i>Conditional Self-Tests</i>	14
2.10	MITIGATION OF OTHER ATTACKS	14
3	SECURE OPERATION	15
3.1	SECURE MANAGEMENT	15
3.1.1	<i>Initialization</i>	15
3.1.2	<i>Management</i>	15
3.1.3	<i>Zeroization</i>	15
3.2	USER GUIDANCE	15
4	ACRONYMS	17

Table of Figures

FIGURE 1 – V-SERIES APPLIANCE HARDWARE BLOCK DIAGRAM.....	6
FIGURE 2 – LOGICAL BLOCK DIAGRAM AND CRYPTOGRAPHIC BOUNDARY	7

List of Tables

TABLE 1 – SECURITY LEVEL PER FIPS 140-2 SECTION	5
TABLE 2 – FIPS INTERFACE MAPPINGS.....	8
TABLE 3 – CRYPTO OFFICER SERVICES.....	9
TABLE 4 – USER SERVICES	9
TABLE 5 – FIPS-APPROVED ALGORITHM IMPLEMENTATIONS	11
TABLE 6 – CRYPTOGRAPHIC KEYS, CRYPTOGRAPHIC KEY COMPONENTS, AND CSPs.....	12
TABLE 7 – ACRONYMS	17



Introduction

1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the Websense Crypto Module Java (Software Version: 1.0) from Websense, Inc. This Security Policy describes how the Websense Crypto Module Java meets the security requirements of Federal Information Processing Standards (FIPS) Publication 140-2, which 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) and the Communications Security Establishment Canada (CSEC) Cryptographic Module Validation Program (CMVP) website at <http://csrc.nist.gov/groups/STM/cmvp>.

This document also describes how to run the module in a secure FIPS-Approved mode of operation. This policy was prepared as part of the Level 1 FIPS 140-2 validation of the module. The Websense Crypto Module Java 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 Websense website (<http://www.websense.com>) contains information on the full line of products from Websense.
- The CMVP website (<http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm>) contains contact information for individuals to answer technical or sales-related questions for the module.

1.3 Submission Package 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 Model document
- Other supporting documentation as additional references

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

2

Websense Crypto Module Java

2.1 Overview

Websense has integrated its portfolio of Web security, email security, and data loss prevention (DLP) solutions into a single, unified, and consolidated offering called TRITON™. The TRITON solution is a highly-flexible and scalable architecture that unifies content analysis, platforms, and security management. Unified content analysis includes real-time threat analysis provided by the Websense Advanced Classification Engine (ACE). Powered by the Websense ThreatSeeker® Network, ACE combines multiple analytic methods to dynamically classify both incoming and outgoing content. The TRITON solution also combines the management and reporting capabilities for Websense Web Security, email security, and DLP technologies into a single interface, providing greater visibility, control, and management capabilities.

- Web Security – Websense Web Security Gateway enables businesses to take full advantage of today's social, interactive Web while lowering costs across the enterprise. Based on the Websense TRITON architecture, Web Security Gateway consolidates real-time Web 2.0 security, enterprise-class data loss prevention (DLP), and email security, both in the cloud and on-premises.

Websense Web Security Gateway solutions offer:

- Zero-day and Web 2.0 malware protection
 - Social networking security
 - Enterprise-class DLP
 - Advanced application and protocol controls
 - Visibility into encrypted SSL traffic
 - Reduced bandwidth consumption
 - Lower total cost of ownership
- Data Loss Prevention – As a part of Websense TRITON solution, data loss prevention technologies provide a market-leading DLP capabilities designed to secure sensitive information and intellectual property. DLP has the ability to detect and discover sensitive data, stored or in transmission, throughout the network and help prevents loss of data through multiple channels such as email, Web, USB¹ drives, LAN² storage, and printing.

Websense DLP solutions help simplify the task of achieving regulatory compliance by providing capabilities to manage and enforce regulatory requirements. The solutions also unlock the full potential of Web 2.0 applications without the concern for inadvertent or deliberate posting of inappropriate or sensitive data.

- Email Security – Websense Email Security Gateway Anywhere provides the next generation of email security with fully integrated enterprise-class DLP, superior malware protection with world-class Web security analytics, and centralized management of email, data, and Web security across platforms in a single, unified console. When integrated with DLP for email traffic, the solution provides accurate detection of email traffic without complex tuning, maximum performance and resiliency, and superior malware protection using multiple advanced analytics to defend against blended attacks and harmful malware.

The Websense Crypto Module Java provides cryptographic and secure communication services for the Websense-developed family of solutions described above. These solutions are deployed on two forms:

- Pre-installed on high-performance, pre-configured, security-hardened hardware
- A fully-customizable “ready-to-install” software package

¹ USB – Universal Serial Bus

² LAN – Local Area Network

The Websense Crypto Module Java is validated at the FIPS 140-2 Section levels listed in **Error! Reference source not found. Error! Reference source not found.**

Table 1 – Security Level Per FIPS 140-2 Section

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

2.2 Module Specification

The Websense Crypto Module Java is a software module with a multi-chip standalone embodiment. The overall security level of the module is 1. The following sections will define the physical and logical boundary of the module.

2.2.1 Physical Cryptographic Boundary

As a software cryptographic module, there are no physical security mechanisms implemented; the module must rely on the physical characteristics of the host platform. The physical cryptographic boundary of the Websense Crypto Module Java is defined by the hard enclosure around the host platform on which it executes. Figure 1 below shows the physical block diagram of the module which runs on a Windows Server 2008 R2 (64-bit) OS⁵ with JRE v1.6.0 and on the commercial off-the-shelf (COTS) V-Series appliance.

³ N/A – Not Applicable

⁴ EMI/EMC – Electromagnetic Interference / Electromagnetic Compatibility

⁵ OS – Operating System

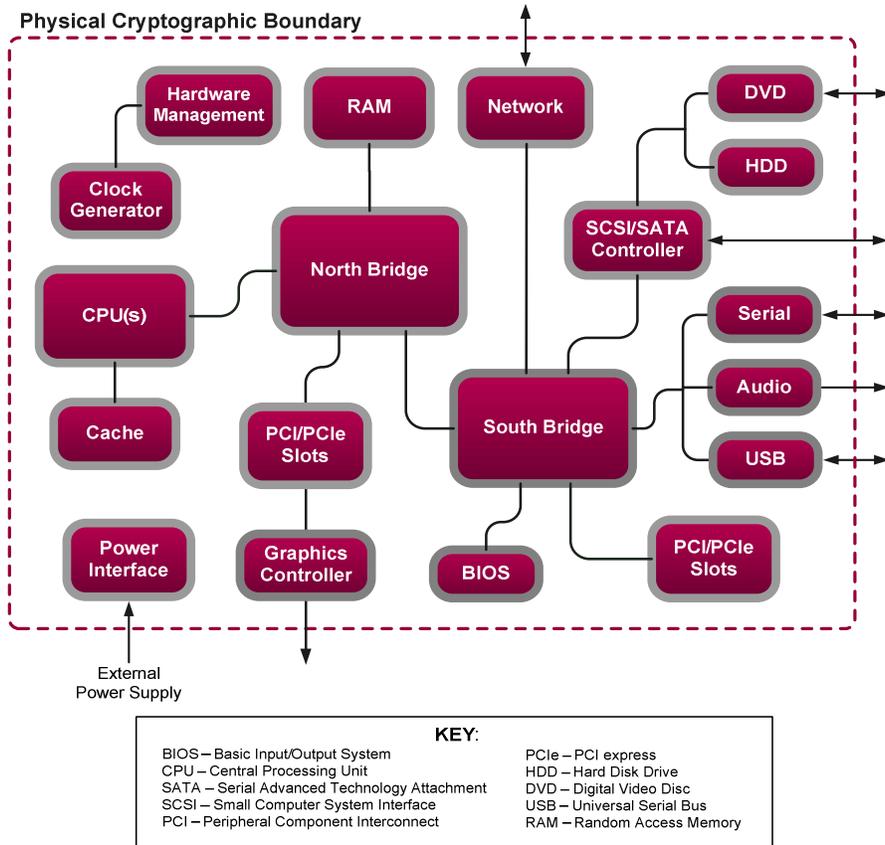


Figure 1 – V-Series Appliance Hardware Block Diagram

2.2.2 Logical Cryptographic Boundary

Figure 2 below shows a logical block diagram of the module, where “Calling Application” represents any other software/firmware component loaded on the appliance that employs the module’s services. The module’s logical cryptographic boundary (also illustrated in Figure 2) encompasses all functionality provided by the module as described in this document. The module takes the form of a single shared library that can be called by calling applications to provide cryptographic services.

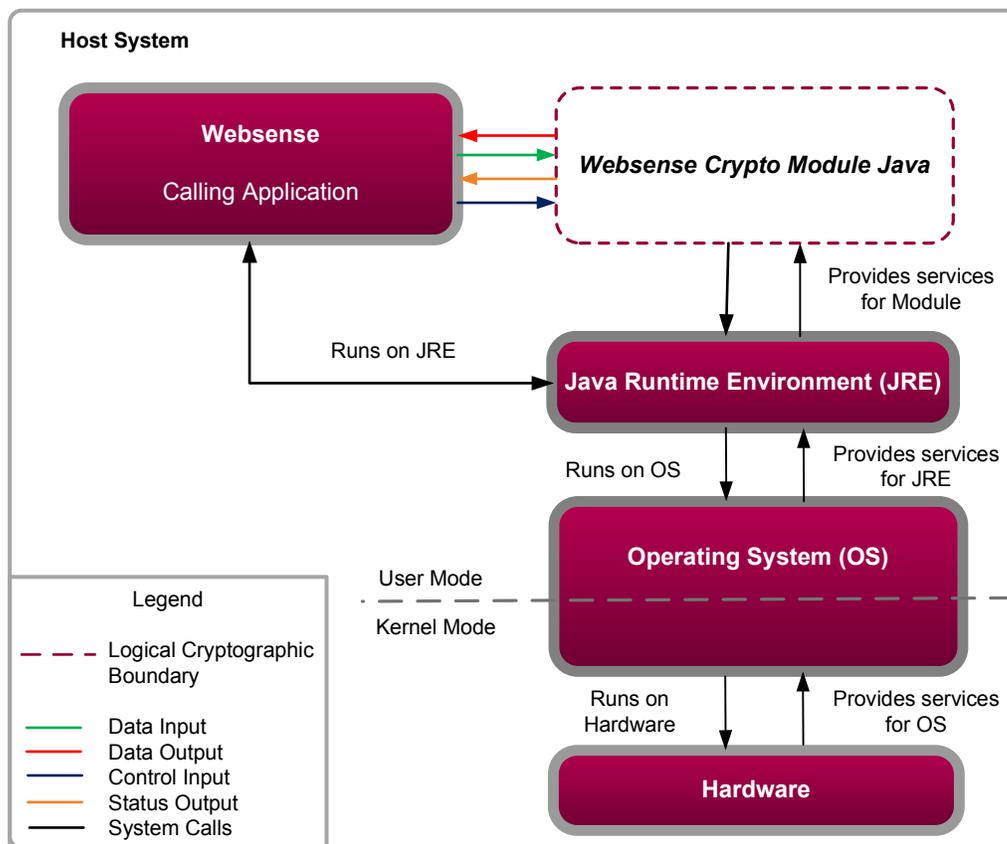


Figure 2 – Logical Block Diagram and Cryptographic Boundary

The cryptographic module is a single Java Archive (JAR) binary that provides cryptographic and secure communication services for other applications developed by Websense. In this document, those applications will be referred to as the calling application. The module is used by the calling application to provide encryption/decryption, hash verification, hashing, cryptographic key generation, random bit generation, and message authentication functions.

2.3 Module Interfaces

The module's logical interfaces exist at a low level in the software as an Application Programming Interface (API). The module's logical and physical interfaces can be categorized into the following interfaces defined by FIPS 140-2:

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

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 platform.

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

Table 2 – FIPS Interface Mappings

FIPS 140-2 Interface	Physical Interface	Module Interface (API)
Data Input	Network/Serial/USB ⁶ port, DVD ⁷ , PCIe ⁸ slot	Function calls that accept, as their arguments, data to be used or processed by the module
Data Output	Network/Serial/USB port, DVD, Graphics/Video port, PCIe slot	Arguments for a function that specify where the result of the function is stored
Control Input	Network/Serial/USB port, LCD ⁹ button, PCIe slot, System identification button, Power button	Function calls utilized to initiate the module and the function calls used to control the operation of the module.
Status Output	Network/Serial/USB port, Graphics/Video, LED ¹⁰ indicators, LCD, PCIe slot	Return status for function calls
Power Input	Power plug/adaptor, Power Switch	Not Applicable

2.4 Roles and Services

There are two roles in the module that operators may assume: a Crypto Officer role and User role. The Crypto Officer is responsible for managing the module and monitoring the module's status, while the User accesses the services implemented by the module. The available functions are utilized to provide or perform the cryptographic services.

The various services offered by the module are described in Table 3 and Table 4. The Critical Security Parameters (CSPs) used by each service are also listed. Please note that the keys and CSPs listed in the tables use the following notation to indicate the type of access required:

- R – Read: The keys and CSPs are read.
- W – Write: The keys and CSPs are established, generated, modified, or zeroized.
- X – Execute: The keys and CSPs are used within an Approved or Allowed security function or authentication mechanism.

2.4.1 Crypto Officer Role

The Crypto Officer (CO) role is responsible for zeroizing keys and CSPs, executing self-tests, and monitoring status. Descriptions of the services available to the Crypto Officer role are provided in Table 3.

⁶ USB – Universal Serial Bus

⁷ DVD – Digital Versatile Disc

⁸ PCIe – Peripheral Component Interconnect Express

⁹ LCD – Liquid Crystal Display

¹⁰ LED – Light-Emitting Diode

Table 3 – Crypto Officer Services

Service	Description	Input	Output	CSP and Type of Access
Initialize module	Performs integrity check and power-up self-tests	API call parameters	Status	Integrity check HMAC key – X
Show status	Returns the current mode of the module	None	Status	None
Run self-tests on demand	Performs power-up self-tests	None	Status	Integrity check HMAC key – X
Zeroize key	Zeroizes and de-allocates memory containing sensitive data	None	None	AES key – W TDES key – W HMAC key – W RSA private/public key – W DSA private/public key – W DH ¹¹ components – W

2.4.2 User Role

The User role can utilize the module's cryptographic functionalities. Descriptions of the services available to the User role are provided in Table 4.

Table 4 – User Services

Service	Description	Input	Output	CSP and Type of Access
Generate random number (ANSI X9.31)	Returns the specified number of random bits to calling application	API call parameters	Status, random bits	ANSI X9.31 RNG seed – RWX ANSI X9.31 seed key – RX
Generate message digest (SHA ¹²)	Compute and return a message digest using SHA algorithms	API call parameters, message	Status, hash	None
Generate keyed hash (HMAC ¹³)	Compute and return a message authentication code using HMAC SHA	API call parameters, key, message	Status, hash	HMAC key – RX
Symmetric encryption	Encrypt plaintext using supplied key and algorithm specification (Triple DES ¹⁴ or AES ¹⁵)	API call parameters, key, plaintext	Status, ciphertext	AES key – RX TDES key – RX
Symmetric decryption	Decrypt ciphertext using supplied key and algorithm specification (TDES or AES)	API call parameters, key, ciphertext	Status, plaintext	AES key – RX TDES key – RX
Generate asymmetric key pair	Generate and return the specified type of asymmetric key pair (RSA ¹⁶ or DSA ¹⁷)	API call parameters	Status, key pair	RSA private/public key – W DSA private/public key – W

¹¹ DH – Diffie-Hellman

¹² SHA – Secure Hash Algorithm

¹³ HMAC – (Keyed-) Hash Message Authentication Code

¹⁴ DES – Data Encryption Standard

¹⁵ AES – Advanced Encryption Standard

¹⁶ RSA – Rivest, Shamir, Adleman

¹⁷ DSA – Digital Signature Algorithm

Service	Description	Input	Output	CSP and Type of Access
DH key agreement	Perform key agreement using Diffie-Hellman algorithm	API call parameter	Status, key components	DH components – W
Signature Generation	Generate a signature for the supplied message using the specified key and algorithm (RSA or DSA)	API call parameters, key, message	Status, signature	RSA private key – RX DSA private key – RX
Signature Verification	Verify the signature on the supplied message using the specified key and algorithm (RSA or DSA)	API call parameters, key, signature, message	Status	RSA public key – RX DSA public key – RX

2.4.3 Authentication

The module does not support any authentication mechanism. Operators of the module implicitly assume a role based on the service of the module being invoked. Since all services offered by the module can only be used by either the Crypto Officer or the User, the roles are mutually exclusive. Thus, when the operator invokes a Crypto Officer role service, he implicitly assumes the Crypto Officer role. When the operator invokes a User role service, he implicitly assumes the User role.

2.5 Physical Security

Since this is a software module, the module relies on the target platform (a purpose-built Websense appliance) to provide the mechanisms necessary to meet FIPS 140-2 physical security requirements. All components of the target platform are made of production-grade materials, and all integrated circuits are coated with commercial standard passivation.

2.6 Operational Environment

The Websense Crypto Module Java was tested and found compliant with the FIPS 140-2 requirements on the following operating system:

- 64-bit Windows Server 2008 R2 with Java Runtime Environment (JRE) v1.6.0.

All cryptographic keys and CSPs are under the control of the OS, which protects the keys and CSPs against unauthorized disclosure, modification, and substitution. The module only allows access to keys and CSPs through its well-defined APIs. The module performs a Software Integrity Test using a FIPS-Approved message authentication code (HMAC SHA-512).

The Websense Crypto Module Java, when compiled from the same unmodified source code, is vendor-affirmed to be FIPS 140-2 compliant when running on the following supported operating systems:

- Xen 3.2.0 running on CentOS 5.1
- 32-bit Windows 2008 R2 SP1

2.7 Cryptographic Key Management

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

Table 5 – FIPS-Approved Algorithm Implementations

Algorithm	Certificate Number
Symmetric Key Algorithm	
AES - 128-, 192-, 256-bit in ECB ¹⁸ , CBC ¹⁹ , CFB ²⁰ 128 and OFB ²¹ modes	1936
Triple-DES - 112-, 168-bit in ECB, CBC, CFB8, CFB64, and OFB modes	1262
Asymmetric Key Algorithm	
RSA (PKCS ²² #1.5, PSS ²³) sign/verify: 1024-, 1536-, 2048-, 3072-, 4096-bit	1002
RSA (ANSI X9.31) key generation with 1024, 1536, 2048, 3072, 4096 bit keys	1002
DSA sign/verify: 1024-bit	618
Secure Hashing Algorithm (SHA)	
SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	1701
Message Authentication Code (MAC)	
HMAC using SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	1169
Pseudo Random Number Generation (PRNG)	
ANSI X9.31 Appendix A.2.4 PRNG	1020

NOTE: The following security functions have been deemed “deprecated” or “restricted” by NIST. Please refer to NIST Special Publication 800-131A for further details.

- two-key Triple DES for encryption
- ANSI X9.31 Appendix A.2.4 PRNG
- key lengths providing 80 bits of security strength for digital signature generation

Additionally, the module utilizes the following non-FIPS-Approved and non-compliant algorithm implementations:

- 1024-bit Diffie-Hellman (non-compliant) – key agreement: key establishment methodology provides 80 bits of encryption strength
- MD2²⁴
- MD4
- MD5
- Elliptic Curve Diffie-Hellman (non-compliant)
- Elliptic Curve DSA (non-compliant)

The module supports the critical security parameters listed below in Table 6. Please note that the “Input” and “Output” columns in Table 6 are in reference to the module’s logical boundary. Keys that enter and exit the module via an API call parameter are in plaintext.

¹⁸ ECB – Electronic Codebook

¹⁹ CBC – Cipher-Block Chaining

²⁰ CFB – Cipher Feedback

²¹ OFB – Output Feedback

²² PKCS – Public-Key Cryptography Standards

²³ PSS – Probabilistic Signature Scheme

²⁴ MD – Message Digest

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

CSP/Key	CSP/Key Type	Input	Output	Storage	Zeroization	Use
HMAC key (Integrity check)	HMAC key	Never	Never	In module binary	N/A	Software integrity check
AES key	AES128, 192, 256 bit key	Input electronically in plaintext	Never	Plaintext in volatile memory	By power cycle or host reboot	Encryption, decryption
TDES key	TDES 112, 168 bit key	Input electronically in plaintext	Never	Plaintext in volatile memory	By power cycle or host reboot	Encryption, decryption
HMAC key	HMAC key	Input electronically in plaintext	Never	Plaintext in volatile memory	By power cycle or host reboot	Message Authentication with SHS
RSA private key	RSA 1024, 1536, 2048, 3072, 4096 bit key	Input electronically in plaintext	Never	Plaintext in volatile memory	By power cycle or host reboot	Signature generation, decryption
		Internally generated	Output electronically in plaintext			Used by calling application
RSA public key	RSA 1024, 1536, 2048, 3072, 4096 bit key	Input electronically in plaintext	Never	Plaintext in volatile memory	By power cycle or host reboot	Signature verification, encryption
		Internally generated	Output electronically in plaintext			Used by calling application
DSA private key	DSA 1024-bit key	Input electronically in plaintext	Never	Plaintext in volatile memory	By power cycle or host reboot	Signature generation
		Internally generated	Output electronically in plaintext			Used by calling application
DSA public key	DSA 1024-bit key	Input electronically in plaintext	Never	Plaintext in volatile memory	By power cycle or host reboot	Signature verification
		Internally generated	Output electronically in plaintext			Used by calling application
DH public components	Public components of DH protocol	Internally generated	Output electronically in plaintext	Plaintext in volatile memory	By power cycle or host reboot	Used by calling application
DH private component	Private exponent of DH protocol	Internally generated	Never	Plaintext in volatile memory	By power cycle or host reboot	Used by calling application

CSP/Key	CSP/Key Type	Input	Output	Storage	Zeroization	Use
ANSI X9.31 PRNG seed	128-bit random value	Internally generated	Never	Plaintext in volatile memory	Power cycle or host reboot	Generate random number
ANSI X9.31 PRNG seed key	AES 128, 192, 256-bit key	Internally generated	Never	Plaintext in volatile memory	Power cycle or host reboot	Generate random number

2.7.1 Key Generation

The module supports the generation of the DSA, RSA, and Diffie-Hellman (DH) public and private keys. The module also 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. However, keys are passed to the module as parameters from the applications resident on the host platform via the exposed APIs. Similarly, keys and CSPs exit the module in plaintext via the well-defined exported APIs.

2.7.3 Key/CSP Storage and Zeroization

As a software module, the module does not provide for the persistent storage of keys and CSPs. Keys and CSPs stored in RAM can be zeroized by a power cycle or a host platform reboot. Additionally, symmetric, asymmetric, and HMAC keys are either provided by or delivered to the calling process, and are subsequently destroyed by the module at the completion of the API call. The ANSI X9.31 PRNG seed and seed key are initialized by the module at power-up and remain stored in RAM until the module is uninitialized by a host platform reboot or power cycle.

The key zeroization techniques used for clearing volatile memory, once invoked, take effect immediately, and do not allow sufficient time to compromise any plaintext secret and private keys and CSPs stored by the module.

2.8 EMI/EMC

Websense Crypto Module Java is a software module. Therefore, the only electromagnetic interference produced is that of the host platform on which the module resides and executes. FIPS 140-2 requires that the host systems on which FIPS 140-2 testing is performed meet the 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. The V-Series Appliance hardware meets these FCC requirements.

2.9 Self-Tests

The Websense Crypto Module Java performs a set of self-tests upon power-up and conditionally during operation as required in FIPS 140-2.

2.9.1 Power-Up Self-Tests

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

- Software integrity check using HMAC SHA-512

- Known Answer Tests (KATs) for:
 - AES
 - Triple-DES
 - SHA-1
 - HMAC SHA1
 - HMAC SHA-224
 - HMAC SHA-256
 - HMAC SHA-384
 - HMAC SHA-512
 - RSA (sign/verify)
 - PRNG
- Pair-wise Consistency Test for:
 - DSA

2.9.2 Conditional Self-Tests

The module performs the following conditional self-tests:

- Continuous RNG test
- RSA Pair-wise Consistency test
- DSA Pair-wise Consistency test

If any self-test fails, the module will enter a critical error state, during which cryptographic functionality and all data output is inhibited. To clear the error state, the CO must reboot the host system, reload the module, or restart the calling application.

2.10 Mitigation of Other Attacks

This section is not applicable. The modules do not claim to mitigate any attacks beyond the FIPS 140-2 Level 1 requirements for this validation.



Secure Operation

The Websense Crypto Module Java 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. Section 3.1 below provides guidance to the Crypto Officer for managing the module.

3.1 Secure Management

The Websense Crypto Module Java is distributed only as part of Websense's system applications and is not distributed as a separate binary. Thus, module operators are not required to perform any steps to ensure that the module is running in its FIPS-Approved mode of operation.

3.1.1 Initialization

When the module is installed and initialized, the module is considered to be running in its FIPS-Approved mode of operation. After initialization, the function *performFipsSelfTest()* is called which, in turn, calls other function for performing power-up self-tests. Upon initialization of the module, the module requires no set-up and runs its power-up self-tests which includes software integrity test that checks the integrity of the module by using an HMAC SHA-512 digest. If the integrity check succeeds, then the module performs power-up self-tests. If the module passes all the self-tests, the module is in its FIPS-Approved mode of operation, indicating the success of the power-up self-tests. Conversely, upon failure, the module and the application are terminated, indicating the failure of the power-up self-test.

Power-up self-tests can also be performed on demand by cycling the power on the host platform, reloading the module, or invoking the *performFipsSelfTests()* function.

3.1.2 Management

Since the Crypto Officer cannot directly interact with the module, no specific management activities are required to ensure that the module runs securely; the module only executes in a FIPS-Approved mode of operation. If any irregular activity is noticed or the module is consistently reporting errors, then Websense, Inc. Customer Support should be contacted.

3.1.3 Zeroization

The module does not persistently store any key or CSPs. All ephemeral keys used by the module are zeroized upon session termination. All keys and CSPs can be zeroized by power cycling or rebooting the host platform..

3.2 User Guidance

As the module also allows access to cryptographic services that are not FIPS-Approved or that provide less than the minimum NIST-recommended encryption strength, it is the responsibility of the calling application developer to ensure that only appropriate algorithms, key sizes, and key establishment techniques are applied. Users are responsible to use only the services that are listed in Table 4 above. Any use of the Websense Crypto Module Java with non-FIPS-Approved cryptographic services or keys that provide less than 80 bits of encryption strength constitutes a departure from this Security Policy, and results in the module not being in its Approved mode of operation.

Although the User does not have any ability to modify the configuration of the module, they should notify the Crypto Officer if any irregular activity is noticed.

4 Acronyms

This section describes the acronyms used in this document.

Table 7 – Acronyms

Acronym	Definition
ACE	Advanced Classification Engine
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
CO	Crypto Officer
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
CSEC	Communications Security Establishment Canada
CSP	Critical Security Parameter
CTR	Counter
DES	Data Encryption Standard
DH	Diffie-Hellman
DLP	Data Loss Prevention
DSA	Digital Signature Algorithm
DVD	Digital Versatile Disc
ECB	Electronic Code Book
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FCC	Federal Communications Commission
FIPS	Federal Information Processing Standard
FTP	File Transfer Protocol
HMAC	(Keyed-) Hash Message Authentication Code
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
JAR	Java Archive
JRE	Java Runtime Environment
KAT	Known Answer Test

Acronym	Definition
MD	Message Digest
NIST	National Institute of Standards and Technology
NVLAP	National Voluntary Laboratory Accreditation Program
OS	Operating System
PCIe	Peripheral Component Interconnect Express
PKCS	Public-Key Cryptography Standards
PSS	Probabilistic Signature Scheme
RHEL	Red Hat Enterprise Linux
RNG	Random Number Generator
RSA	Rivest Shamir and Adleman
SHA	Secure Hash Algorithm
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
TDES	Triple Data Encryption Standard
URL	Uniform Resource Locator
USB	Universal Serial Bus

Prepared by:
Corsec Security, Inc.

The logo for Corsec, featuring the word "Corsec" in a bold, red, serif font. The text is centered within a white, horizontally-oriented oval shape that has a subtle 3D effect with a light blue shadow on the right side.

13135 Lee Jackson Memorial Highway, Suite 220
Fairfax, VA 22033
United States of America

Phone: +1 703-267-6050
Email: info@corsec.com
<http://www.corsec.com>

