# Diversinet Corp.

## Diversinet Java Crypto Module

Software Version: 2.0

## FIPS 140-2 Non-Proprietary Security Policy

FIPS Security Level: 1
Document Version: 0.10

Prepared for:


**Diversinet Corp.**
2235 Sheppard Avenue East, Suite 1700
Toronto, Ontario M2J5B5
Canada

Phone: +1 (416) 756-2324
Email: sales@diversinet.com
http://www.diversinet.com


Prepared by:


**Corsec Security, Inc.**
13135 Lee Jackson Memorial Hwy., Suite 220
Fairfax, VA 22033
United States of America

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

# Table of Contents



# Table of Figures



# List of Tables





© 2011 Diversinet Corp.
This document may be freely reproduced and distributed whole and intact including this copyright notice.

# 1 Introduction

## 1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the Diversinet Java Crypto Module from Diversinet Corp. This Security Policy describes how the Diversinet Java Crypto Module 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 Diversinet Java Crypto Module is referred to in this document as the Diversinet Java Crypto Module, the crypto-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 Diversinet website (www.diversinet.com) contains information on the full line of products from Diversinet.
- 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 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 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 to Diversinet. With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 Submission Package is proprietary to Diversinet and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact Diversinet.



© 2012 Diversinet Corp.
This document may be freely reproduced and distributed whole and intact including this copyright notice.

# 2 Diversinet Java Crypto Module

## 2.1 Overview

The Diversinet MobiSecure® Platform enables enterprises to rapidly deploy HIPAA[1]-compliant mobile health applications to anyone, anywhere, on any mobile device.

MobiSecure® Platform offers convenient and secure management of critical data and personal information, from a mobile phone, using SMS[2], or a tablet or PC[3], directly over the Internet and wireless networks, utilizing SOAP[4]. Furthermore, it provides organizations the necessary tools to deploy customized mobile applications that meet their business and brand needs. MobiSecure® Platform uses two-factor authentication technology for user and device authentication and state-of-the-art encryption techniques for data protection. Its uniqueness and strengths lies in the ability for organizations to securely publish content to a wide range of different mobile phones. In addition, it enables the customization of both the application configuration as well as the user interface over-the-air without the need of re-deployment. Figure 1, below, illustrates the MobiSecure® Platform.

The security of MobiSecure® Platform is formed around in-depth security design principles that provide controls at multiple levels of data storage, access and transfer. Although Diversinet's focus and expertise is in designing and implementing high security standards at the software architecture, application and cryptographic levels, Diversinet works with it customers to define the overall security strategy, which includes operations security; communication and network security; access controls; business continuity and disaster recovery; and compliance with privacy regulations.

[1] HIPAA – Health Insurance Portability and Accountability Act
[2] SMS – Short Message Service
[3] PC – Personal Computer
[4] SOAP – Simple Object Access Protocol



© 2012 Diversinet Corp.
This document may be freely reproduced and distributed whole and intact including this copyright notice.

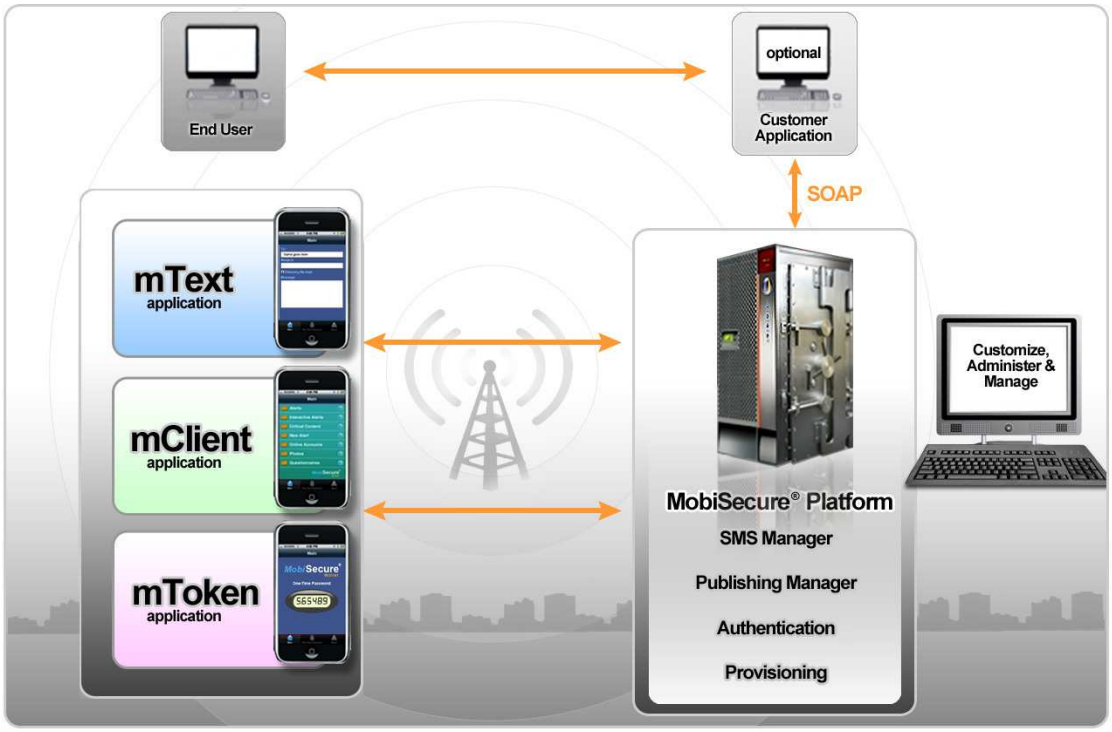


**Figure 1 – Diversinet MobiSecure® Platform**

The Diversinet Java Crypto Module is the FIPS validated module that provides cryptographic functionality for the MobiSecure® Platform. The Diversinet Java Crypto Module is validated at the following FIPS 140-2 Section levels:

**Table 1 – Security Level per FIPS 140-2 Section**

| 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[5] | 1 |
| 9 | Self-tests | 1 |
| 10 | Design Assurance | 1 |
| 11 | Mitigation of Other Attacks | N/A |
| 14 | Cryptographic Module Security Policy | 1 |

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



© 2012 Diversinet Corp.
This document may be freely reproduced and distributed whole and intact including this copyright notice.

# 2.2 Module Specification

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

The cryptographic module is a single Java Archive (JAR) binary, named DvSecurityAPI.jar, that provides cryptographic and secure communication services for other applications developed by Diversinet. In this document, those applications will be referred to as the calling application. The module is a standalone cryptographic library that can be called to provide encryption, decryption, hash generation and verification, certificate generation and verification, random bit generation, and message authentication functions.

## 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 Diversinet Java Crypto Module is defined by the hard enclosure around the host platform on which it executes. Figure 2 below shows the physical block diagram of the module which runs on a Windows Server 2008 R2 (64-bit) OS[6] with JDK[7] 1.6 and on a GPC[8] using an Intel Xeon processor.

[6] OS – Operating System
[7] JDK – Java Development Kit
[8] GPC – General-Purpose Computer



© 2012 Diversinet Corp.
This document may be freely reproduced and distributed whole and intact including this copyright notice.

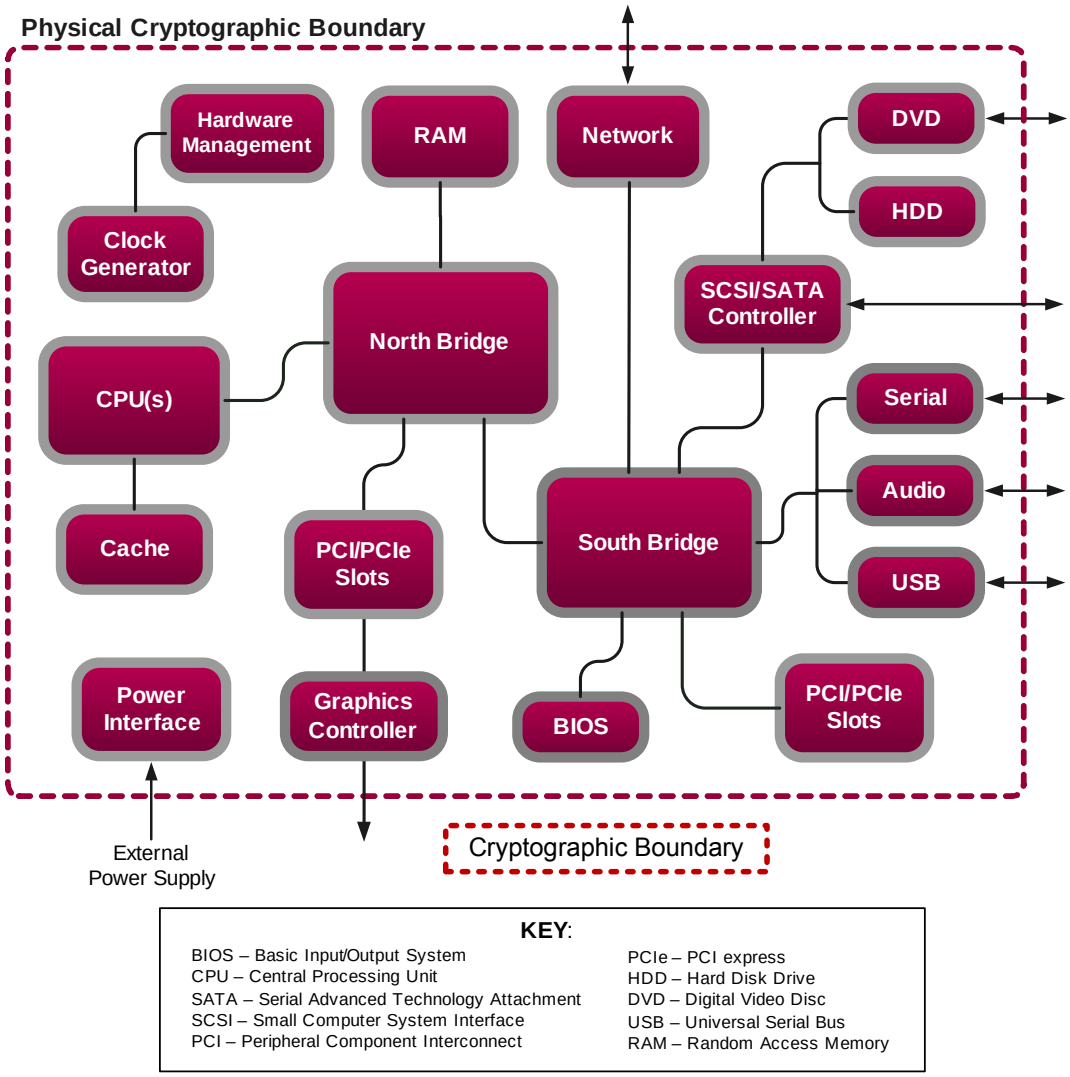


**Figure 2 – GPC Server Block Diagram**

## 2.2.2 Logical Cryptographic Boundary

Figure 3, below, shows a logical block diagram of the module. The module's logical cryptographic boundary (also illustrated in Figure 3) 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, executing in the JVM[9], to provide cryptographic services.

[9] JVM – Java Virtual Machine



© 2012 Diversinet Corp.
This document may be freely reproduced and distributed whole and intact including this copyright notice.

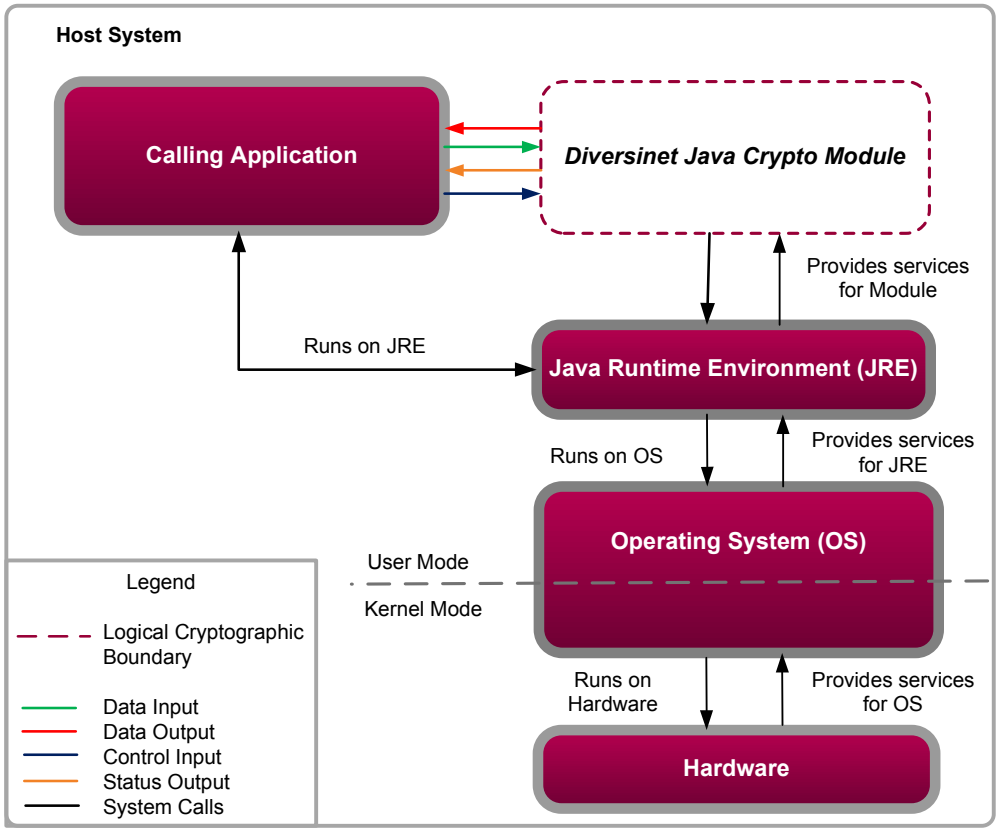


**Figure 3 – Diversinet Java Crypto Module Logical Block Diagram**

# 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.

All of these physical interfaces of the host platform are separated into logical interfaces defined by FIPS 140-2, as described in the following table:



© 2012 Diversinet Corp.
This document may be freely reproduced and distributed whole and intact including this copyright notice.

**Table 2 – FIPS 140-2 Logical Interface Mappings**

| FIPS 140-2 Interface | Physical Interface | Module Interface (API) |
|---|---|---|
| Data Input | Network port, Serial port, USB[10] port, DVD[11], SCSI[12]/SATA[13] Controller | Method calls that accept, as their arguments, data to be used or processed by the module |
| Data Output | Network port, Serial port, USB port, SCSI/SATA Controller | Arguments for a method that specify where the result of the method is stored |
| Control Input | Network port, Serial port, USB port, Power button | Method calls utilized to initiate the module and the method calls used to control the operation of the module. |
| Status Output | Network port, Serial port, USB port, Graphics controller, Audio port | Return status for method calls |
| Power Input | Power Switch | Not Applicable |

# 2.4 Roles and Services

There are two roles in the module (as required by FIPS 140-2) that operators may assume: a Crypto Officer role and a User role.

## 2.4.1 Crypto Officer Role

The Crypto Officer role has the ability to initialize the module, determine module status, and zeroize all module keys and CSPs[14]. Descriptions of the services available to the Crypto Officer role are provided in Table 3, below. Please note that the keys and CSPs listed in the table indicate the type of access required using the following notation:

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

[10] USB – Universal Serial Bus
[11] DVD – Digital Versatile Disc
[12] SCSI – Small Computer Systems Interface
[13] SATA – Serial Advanced Technology Attachment
[14] CSP – Critical Security Parameter



© 2012 Diversinet Corp.
This document may be freely reproduced and distributed whole and intact including this copyright notice.

**Table 3 – Crypto Officer Services**

| Service | Description | CSP and Type of Access |
|---|---|---|
| Initialize module | Loads module and calls power-up self-tests | Integrity check HMAC[15] key – X |
| ModuleIntegrity.isModuleOperative() | Returns the current mode of the module | N/A |
| Zeroize keys | Format hard-drive of host GPC | AES key – W<br>Triple DES key – W<br>HMAC key – W<br>RSA private key – W<br>RSA public key – W |

Key zeroization by the Crypto-Officer will take the form of a full format of the host GPC's hard-drive. This command is not a part of the module's API. The host GPC's OS provides this service.

### 2.4.2 User Role

The User role has the ability to generate all CSPs supported by the module and perform encryption, decryption, and verification operations with the generated CSPs. Descriptions of the services available to the User role are provided in Table 4, below.

**Table 4 – User Services**

| Service | Description | CSP and Type of Access |
|---|---|---|
| SHA1Engine.update(), SHA1Engine.complete() | Compute and return a message digest using the SHA[16]-1 algorithm | None |
| SHA256Engine.update(). SHA256Engine.complete() | Compute and return a message digest using the SHA-256 algorithm | None |
| SHA512Engine.update(), SHA512Engine.complete() | Compute and return a message digest using the SHA-512 algorithm | None |
| ModuleIntegrity.algorithmsTest(), ModuleIntegrity.integrityCheck() | Performs power-up self-tests | Integrity check HMAC key – RX<br>AES key – X<br>Triple DES key – X<br>RSA private key – X<br>RSA public key – X<br>HMAC keys – X<br>DRBG seed/key – X |
| CTRDRBGEngine.engineNextBytes () | Generates 128-bit blocks of random bits, up to the requested bit size. | DRBG seed/key – X |

[15] HMAC – (Keyed-)Hash Message Authentication Code
[16] SHA – Secure Hash Algorithm



© 2012 Diversinet Corp.
This document may be freely reproduced and distributed whole and intact including this copyright notice.

| Service | Description | CSP and Type of Access |
|---|---|---|
| HMacEngine.update(), HMacEngine.complete() | Compute and return a message authentication code using HMAC SHA-1, HMAC SHA-256, or HMAC SHA-512 | HMAC key – RX |
| DESEngine.process() | Encrypt/decrypt plaintext/ciphertext using Triple-DES algorithm specification | Triple DES key – RX |
| AESEngine.process() | Encrypt/decrypt plaintext/ciphertext using AES algorithm specification | AES key – RX |
| RSAPSSEngine.sign() | Generate a signature for the supplied message using the specified key and algorithm | RSA private key – RX |
| RSAPSSEngine.verify() | Verify the signature on the supplied message using the specified key and algorithm | RSA public key – RX |
| RSAEngine.process() | Encrypt/Decrypt a block of data using the specified key | RSA private key – RX<br>RSA 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

The Diversinet Java Crypto Module is a software module, which FIPS defines as a multi-chip standalone cryptographic module. As such, it does not include physical security mechanisms. Thus, the FIPS 140-2 requirements for physical security are not applicable.

## 2.6 Operational Environment

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

- Windows Server 2008 R2 (64-bit) and Java Development Kit (JDK) v1.6, running on an Intel Xeon E5530 processor

All cryptographic keys and CSPs are under the control of the host OS, which protects the keys and CSPs against unauthorized disclosure, modification, and substitution. The module only allows access to keys and CSPs through its APIs. The module performs a Software Integrity Test using a FIPS-Approved message authentication code (HMAC SHA-1).



© 2012 Diversinet Corp.
This document may be freely reproduced and distributed whole and intact including this copyright notice.

# 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** | |
| AES: ECB[17],CBC[18], and CTR[19] modes for 128, 192, and 256 bit key sizes | #1965 |
| Triple DES: CBC mode for 1 and 2 keying option | #1276 |
| **Asymmetric Key** | |
| RSA PSS[20] signature generation/verification: 1024- , 2048-bit (w/ SHA-1, SHA-256, SHA-512) | #1017 |
| **Secure Hashing Standard (SHS)** | |
| SHA-1, SHA-256, SHA-512 | #1723 |
| **Message Authentication** | |
| HMAC SHA-1, HMAC SHA-256, HMAC SHA-512 | #1185 |
| **Random Number Generator** | |
| NIST SP800-90 DRBG (Counter-based) | #175 |

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

- The use of two-key Triple DES for encryption is **restricted** after December 31, 2010.
- The use of key lengths providing 80 bits of security strength for digital signature generation is **deprecated** after December 31, 2010.

Additionally, the module utilizes the following non-FIPS-approved algorithm implementations:

- RSA encrypt/decrypt
  - RSA encrypt/decrypt functionality is not FIPS-Approved, but is Allowed in FIPS mode

[17] ECB – Electronic Codebook
[18] CBC – Cipher-Block Chaining
[19] CTR – Counter
[20] PSS – Probabilistic Signature Scheme



© 2012 Diversinet Corp.
This document may be freely reproduced and distributed whole and intact including this copyright notice.

The module supports the critical security parameters (CSPs) listed below in Table 6.

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

| CSP/Key | CSP/Key Type | Generation / Input | Output | Storage | Zeroization | Use |
|---|---|---|---|---|---|---|
| HMAC Integrity key | HMAC key | Never | Never output from the module | In module binary | N/A | Software integrity check |
| AES key | AES 128-, 192-, 256-bit key | Input electronically in plaintext | Never output from the module | Plaintext in volatile memory | N/A | Encryption, decryption |
| Triple DES key | Triple DES 168-bit key | Input electronically in plaintext | Never output from the module | Plaintext in volatile memory | N/A | Encryption, decryption |
| HMAC key | HMAC key | Input electronically in plaintext | Never output from the module | Plaintext in volatile memory | N/A | Message Authentication with SHS |
| RSA private key | RSA 1024-, 2048-bit key | Input electronically in plaintext | Never output from the module | Plaintext in volatile memory | N/A | Signature generation, decryption |
| RSA public key | RSA 1024-, 2048-bit key | Input electronically in plaintext | Never output from the module | Plaintext in volatile memory | N/A | Signature verification, encryption |
| DRBG seed | 1000-bit entropy value | Internally generated from system entropy and SHA-512 hash | Never output from the module | Plaintext in volatile memory | N/A | Used to seed the DRBG |
| DRBG nonce | 64-bit counter value | Internally generated from system entropy | Never output from the module | Plaintext in volatile memory | N/A | Used to seed the DRBG |



© 2012 Diversinet Corp.
This document may be freely reproduced and distributed whole and intact including this copyright notice.

# 2.8 Self-Tests

## 2.8.1 Power-Up Self-Tests

The Diversinet Java Crypto Module performs the following self-tests at power-up:

- Software integrity check (HMAC SHA-1)
- Known Answer Tests (KATs)
    - AES KAT
    - Triple-DES KAT
    - RSA KAT for sign/verify
    - SHA-1 HMAC KAT
    - SHA-256 HMAC KAT
    - SHA-512 HMAC KAT
    - NIST SP800-90 DRBG KAT

Note that the HMAC KATs with SHA-1, SHA-256, and SHA-512 utilize (and thus test) the full functionality of the SHA-1 and SHA-2s; thus, no independent KATs for SHA-1 and SHA-2s implementations are required.

## 2.8.2 Conditional Self-Tests

The Diversinet Java Crypto Module performs the following conditional self-tests:

- NIST SP800-90 DRBG Continuous test

## 2.8.3 Critical Function Tests

The Diversinet Java Crypto Module performs the following critical function tests:

- NIST SP800-90 DRBG health tests

# 2.9 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.



© 2012 Diversinet Corp.
This document may be freely reproduced and distributed whole and intact including this copyright notice.

# 3 Secure Operation

The Diversinet Java Crypto Module 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.

## 3.1 Secure Management

The Diversinet Java Crypto Module is distributed only as part of Diversinet's system applications and is not distributed as a separate binary.

### 3.1.1 Initialization

When the module is installed and initialized the module runs in a FIPS mode of operation. This is achieved by calling a single initialization method. Upon initialization of the module, the module runs its power-up self-tests which includes software integrity test that checks the integrity of the module by using an HMAC SHA-1 digest. If the integrity check succeeds, then the module performs power-up self-tests. When the module passes the integrity test, the module is in its FIPS-Approved mode of operation. If any self-test fails, the module enters a critical error state, ceasing all cryptographic functionality, and throws an exception. The module must be reinstalled to leave the critical error state.

Power-up self-tests can also be performed on demand by invoking the ModuleIntegrity.algorithmsTest() methods or by cycling the power on the host platform.

The module is loaded by applications which make calls to the module as a cryptographic library.

### 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 Diversinet Corp. Customer Support should be contacted.

### 3.1.3 Zeroization

The module does not persistently store any keys or CSPs. All ephemeral keys used by the module are zeroized upon reboot, or session termination.

## 3.2 User Guidance

In order to ensure compliance with NIST SP800-90, the Crypto-Officer must set a method, setMaxTestTimeBetweenInstantiations() to 0 (seconds). This will require that the DRBG health tests be run on every instantiation.Only the module's cryptographic functionalities are available to the User. Users are responsible to use only the services that are listed in Table 4 above. Although the User does not have any ability to modify the configuration of the module, they should report to the Crypto Officer if any irregular activity is noticed.



© 2012 Diversinet Corp.
This document may be freely reproduced and distributed whole and intact including this copyright notice.

# 4 Acronyms

This section describes the acronyms.

**Table 7 – Acronyms**

| Acronym | Definition |
|---|---|
| AES | Advanced Encryption Standard |
| API | Application Programming Interface |
| CBC | Cipher-Block Chaining |
| CMVP | Cryptographic Module Validation Program |
| CSEC | Communications Security Establishment Canada |
| CSP | Critical Security Parameter |
| CTR | Counter |
| DVD | Digital Versatile Disc |
| DRBG | Deterministic Random Bit Generator |
| ECB | Electronic Codebook |
| EMC | Electromagnetic Compatibility |
| EMI | Electromagnetic Interference |
| FIPS | Federal Information Processing Standard |
| HIPAA | Health Insurance Portability and Accountability Act |
| HMAC | (Keyed-)Hash Message Authentication Code |
| GPC | General-Purpose Computer |
| JAR | Java Archive |
| JDK | Java Development Kit |
| JVM | Java Virtual Machine |
| KAT | Known Answer Test |
| NIST | National Institute of Standards and Technology |
| OS | Operating System |
| PC | Personal Computer |
| PSS | Probabilistic Signature Scheme |
| RSA | Rivest, Shamir, and Adleman |
| SATA | Serial Advanced Technology Attachment |
| SCSI | Small Computer System Interface |
| SHA | Secure Hash Algorithm |
| SMS | Short Message Service |
| SOAP | Simple Object Access Protocol |



© 2012 Diversinet Corp.
This document may be freely reproduced and distributed whole and intact including this copyright notice.

| Acronym | Definition |
|---|---|
| **Triple DES** | Triple Data Encryption Standard |
| **USB** | Universal Serial Bus |



© 2012 Diversinet Corp.
This document may be freely reproduced and distributed whole and intact including this copyright notice.

Prepared by:
**Corsec Security, Inc.**

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

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