Odyssey Security Component
Version 2.0
Non-Proprietary Security Policy
Document Version 1.1

Pulse Secure, LLC.

January 9, 2015
TABLE OF CONTENTS

1. MODULE OVERVIEW .............................................................................................................. 3
2. SECURITY LEVEL .................................................................................................................. 4
3. MODES OF OPERATION ...................................................................................................... 5
4. PORTS AND INTERFACES ................................................................................................. 5
5. IDENTIFICATION AND AUTHENTICATION POLICY ...................................................... 6
6. ACCESS CONTROL POLICY ............................................................................................. 7
7. CRYPTOGRAPHIC KEY MANAGEMENT ............................................................................ 10
8. OPERATIONAL ENVIRONMENT ....................................................................................... 10
9. SECURITY RULES ............................................................................................................. 10
10. PHYSICAL SECURITY ...................................................................................................... 11
11. MITIGATION OF OTHER ATTACKS POLICY .............................................................. 11
12. DEFINITIONS AND ACRONYMS .................................................................................. 12
1. Module Overview

The Odyssey Security Component (OSC) is a software module that implements a set of cryptographic algorithms for use by a software application. This Security Policy document details the Odyssey Security Component Version 2.0.

The OSC comprises a dynamic link library, odFIPS2.dll, compiled from source code written using a combination of C, C++ and assembly language implementations on specific platforms. The binary library resides in user space only. The OSC runs on PCs under Windows XP (SP 2) and Windows 2000 (SP 3) operating systems. The module has a multi-chip standalone embodiment as defined by FIPS 140-2.

The module only implements an Approved mode of operation.

The module was operational tested on the following Common Criteria evaluated platforms:

- Dell Optiplex GX400 running Windows 2000 Professional (SP 3)
  CC EAL 4
  CCEVS Validation Report available at:
  http://www.niap-ccevs.org/cc-scheme/st/vid4002/

- Dell Optiplex GX270 running Windows XP Professional (SP 2)
  CC EAL 4
  CCEVS Validation Report available at:
  http://www.niap-ccevs.org/cc-scheme/st/vid9506/

![Figure 1: Hardware Diagram Showing PC Containing Cryptographic Module](image-url)
2. Security Level

The Odyssey Security Component cryptographic module meets the overall requirements applicable to Level 2 security of FIPS 140-2.

Table 1 - Module Security Level Specification

<table>
<thead>
<tr>
<th>Security Requirements Section</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cryptographic Module Specification</td>
<td>2</td>
</tr>
<tr>
<td>Module Ports and Interfaces</td>
<td>2</td>
</tr>
<tr>
<td>Roles, Services and Authentication</td>
<td>2</td>
</tr>
<tr>
<td>Finite State Model</td>
<td>2</td>
</tr>
<tr>
<td>Physical Security</td>
<td>N/A</td>
</tr>
<tr>
<td>Operational Environment</td>
<td>2</td>
</tr>
<tr>
<td>Cryptographic Key Management</td>
<td>2</td>
</tr>
<tr>
<td>EMI/EMC</td>
<td>2</td>
</tr>
<tr>
<td>Self-Tests</td>
<td>2</td>
</tr>
<tr>
<td>Design Assurance</td>
<td>2</td>
</tr>
<tr>
<td>Mitigation of Other Attacks</td>
<td>N/A</td>
</tr>
</tbody>
</table>
3. Modes of Operation

Approved mode of operation

In FIPS mode, the OSC supports the following FIPS Approved algorithms:

- AES 128, 192, 256 – ECB, CBC, and Counter modes (See certificate #783)
- AES-CCM – Key sizes 128, 192, and 256 (See certificate #784)
- Triple-DES – TECB and TCBC modes (See certificate #679)
- SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 (See certificate #787)
- HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512 (See certificate #430)
- DSA Sign/Verify, Key Gen, and PQG Gen/Verify (See certificate #293)
- RSA Sign/Verify (See certificate #373)
- FIPS 186-2 RNG (See certificate #451)

The module only supports an Approved mode of operation. Once loaded into memory and executed, the module is running in FIPS mode. An operator of the module can verify that the module is running in the FIPS Approved mode of operation by first executing the “EnableFIPSModule” command, followed by the “Get State” service, which shall return the following: OD_FIPS_STATE_ENABLED.

The cryptographic module provides the following allowed cryptographic algorithms:

- RSA Encrypt/Decrypt (for Key Transport only) (key wrapping; key establishment methodology provides between 80 and 128 bits of encryption strength)

The following non-Approved algorithm is also available in the Approved mode of operation:

- RSA Encrypt/Decrypt (for bulk data) - No security is claimed for data that has been encrypted using RSA.

4. Ports and Interfaces

All FIPS ports and interfaces are defined as the API of the cryptographic module. The API contains all data input, data output, control input, and status output interfaces to and from the module.
5. Identification and Authentication Policy

Assumption of roles

The OSC cryptographic module shall support two roles, User and Cryptographic Officer. The authentication mechanism is provided by the host Operating System. Proper operation of the module requires that the host Operating System be configured to enforce a password length of at least six characters. The module relies on the Operating System to distinguish between an operator assuming the User role or Crypto Officer role. An operator with Administrator privileges to the Operating System assumes the Crypto Officer role. Table 2 lists these roles along with their required identification and authentication techniques. Table 3 outlines each authentication mechanism and the associated strengths.

Table 2 - Roles and Required Identification and Authentication

<table>
<thead>
<tr>
<th>Role</th>
<th>Type of Authentication</th>
<th>Authentication Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Role-based operator authentication</td>
<td>Password</td>
</tr>
<tr>
<td>Cryptographic Officer</td>
<td>Role-based operator authentication</td>
<td>Password</td>
</tr>
</tbody>
</table>

Table 3 – Strengths of Authentication Mechanisms

<table>
<thead>
<tr>
<th>Authentication Mechanism</th>
<th>Strength of Mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>Each password is at least six characters in length. Characters are chosen from a fifty-two character set. The probability of a successful random attempt is less than (1/52^6), which is less than (1/1,000,000). Assuming that no password lockout settings were configured, that no delay is configured between password attempts, and that an attacker could attempt 100 password entries per minute, the probability of successfully authenticating to the module within one minute through random attempts is (100/(52^6)), which is less than one in 100,000.</td>
</tr>
</tbody>
</table>
6. Access Control Policy

Roles and Services

Table 4 lists each role and the services authorized for each role.

<table>
<thead>
<tr>
<th>Role</th>
<th>Authorized Services</th>
</tr>
</thead>
</table>
| User and Cryptographic Officer: | • AES Encrypt/Decrypt  
|                              | • TDES Encrypt/Decrypt  
|                              | • RSA Sign/Verify  
|                              | • DSA Sign/Verify  
|                              | • Generate Random Number  
|                              | • AES CCM  
|                              | • HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512  
|                              | • RSA Encrypt/Decrypt (for key transport only) - Note: This service is also used for encrypting/decrypting bulk data. However, no security is claimed for data that has been protected by RSA.  
|                              | • RSA Key Generation  
|                              | • DSA Key Generation  
|                              | • AES Key Encryption  
|                              | • Generate Prime Number – Generates a prime number using the FIPS 186-2 RNG  
|                              | • Modular Exponentiation  
|                              | • EnableFIPSModule – Enables all authorized services.  
|                              | • DisableFIPSModule – Disables all authorized services and returns the module to a pre-operational state.  
|                              | • GetState – Returns the current state of the cryptographic module  
|                              | • GetError – Returns a specific error code when the module is in an error state  
|                              | • Run Self-tests – This service executes the suite of power up self-tests required by FIPS 140-2 by calling the API command.  |

Note: In addition to the authenticated "Run Self-Tests" service, self-tests can also be initiated by any operator without authentication by reloading the module into memory.
Definition of Critical Security Parameters (CSPs)

The Critical Security Parameters (CSPs) defined for the OSC consist of cryptographic keys and random numbers used as seeding material. The module does not persistently store CSPs within the logical boundary, and no CSPs are output from the physical bounds of the GPC.

The following secret keys, private keys, and CSPs are supported by the module:

- **AES Keys**: 128, 192 and 256 bit keys used to AES encrypt/decrypt data.
- **TDES Keys**: 3 separate 128 bit DES keys used to TDES encrypt/decrypt data.
- **AES CCM Key**: 128, 192, or 256 bit AES Key used for AES CCM operations.
- **HMAC Keys**: For use during HMAC operations.
- **DSA Signing Private Key**: Used to digitally sign data.
- **RSA Private Key**: Used to digitally sign data.
- **AES Key Encryption Key**: 128 bit AES key for use in AES key wrapping operations.
- **FIPS 186-2 PRNG Seed and Seed Key**: Used for the generation of CSPs and Keys. These values are entered into the module (not internally generated) and the strength of the keys generated depends on the strength of these parameters.
- **HMAC Integrity Key**: HMAC-SHA-512 key used during the Software Integrity Test. (Note: This key is only used for power up self-tests and is not considered a CSP per CMVP IG 7.4.)

Definition of Public Keys:

The following are the public keys contained in the module:

- **RSA Verifying Public Key**: This is the public part of the cryptographic module’s RSA Public/Private key pair used to verify RSA signatures.
- **DSA Public Key**: This is the public part of the cryptographic module’s DSA Public/Private key pair used to verify DSA signatures.
- **RSA Wrapping Key**: Used to perform RSA key transport of keys.

Definition of CSPs Modes of Access

Table 5 defines the relationship between access to CSPs and the different module services. The modes of access shown in the table are defined as follows:

- **Read**
- **Write**
- **Execute**

Each service's API indicates the type of access to CSPs defined by that API. When a CSP is used by the API call to perform particular services, read and execute access is indicated. When a CSP is generated, modified or deleted by the API call, write access is indicated.
Table 5 – Key and CSP Access Rights within Services

<table>
<thead>
<tr>
<th>Approved Services</th>
<th>Keys/CSPs</th>
<th>Authorized Roles</th>
<th>Type of Access</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Symmetric Encryption/Decryption Services</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AES Encrypt/Decrypt</td>
<td>AES Key</td>
<td>User/CO</td>
<td>read, execute</td>
</tr>
<tr>
<td>TDES Encrypt/Decrypt</td>
<td>TDES Key</td>
<td>User/CO</td>
<td>read, execute</td>
</tr>
<tr>
<td><strong>Asymmetric Encryption/Decryption for Key Wrapping Services</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSA Encrypt</td>
<td>RSA Wrapping Public Key</td>
<td>User/CO</td>
<td>read, execute</td>
</tr>
<tr>
<td>RSA Decrypt</td>
<td>RSA Private Key</td>
<td>User/CO</td>
<td>read, execute</td>
</tr>
<tr>
<td><strong>Message Authentication Services</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AES-CCM</td>
<td>AES-CCM Key</td>
<td>User/CO</td>
<td>read, execute</td>
</tr>
<tr>
<td>HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512</td>
<td>HMAC Key</td>
<td>User/CO</td>
<td>read, execute</td>
</tr>
<tr>
<td><strong>Digital Signature Generation/Verification Services</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSA Verify</td>
<td>RSA Verifying Public Key</td>
<td>User/CO</td>
<td>read, execute</td>
</tr>
<tr>
<td>RSA Sign</td>
<td>RSA Private Key</td>
<td>User/CO</td>
<td>read, execute</td>
</tr>
<tr>
<td>DSA Verify</td>
<td>DSA Public Key</td>
<td>User/CO</td>
<td>read, execute</td>
</tr>
<tr>
<td>DSA Sign</td>
<td>DSA Signing Private Key</td>
<td>User/CO</td>
<td>read, execute</td>
</tr>
<tr>
<td><strong>Symmetric Key Wrapping Service</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AES Key Encryption</td>
<td>AES Key Encryption Key</td>
<td>User/CO</td>
<td>read, execute</td>
</tr>
<tr>
<td><strong>Symmetric Key Generation Service</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generate Random Number</td>
<td>FIPS 186-2 PRNG Seed and Seed Key</td>
<td>User/CO</td>
<td>read, execute</td>
</tr>
<tr>
<td><strong>Asymmetric Key Generation Services</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSA Key Generation</td>
<td>RSA Public/Private Key Pair</td>
<td>User/CO</td>
<td>write</td>
</tr>
<tr>
<td>DSA Key Generation</td>
<td>DSA Public/Private Key Pair</td>
<td>User/CO</td>
<td>write</td>
</tr>
<tr>
<td><strong>Other Services</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generate Prime Number</td>
<td>FIPS 186-2 PRNG Seed and Seed Key</td>
<td>User/CO</td>
<td>read, execute</td>
</tr>
<tr>
<td>Modular Exponentiation</td>
<td>N/A</td>
<td>User/CO</td>
<td>N/A</td>
</tr>
<tr>
<td>EnableFIPSModule</td>
<td>N/A</td>
<td>User/CO</td>
<td>N/A</td>
</tr>
</tbody>
</table>
7. Cryptographic Key Management

Key Generation
The cryptographic module supports generation of DSA and RSA public and private keys, using the Approved FIPS 186-2 deterministic random number generator.

Key Storage
The module does not persistently store keys. Key material is provided for use through a defined API, stored in RAM, and then destroyed once processing is terminated. If the operator wishes to store keys they are responsible for doing so outside of the cryptographic module's logical boundary.

Zeroization
All key data exists in data structures allocated within the cryptographic module, and can only be returned to an authorized user using the defined API. The operating system protects system memory and process space from access by unauthorized users. The operator of the cryptographic module should follow the steps outlined in the module’s API specification to ensure sensitive data is protected by zeroizing the data from memory when it is no longer needed.

8. Operational Environment
The FIPS 140-2 Area 6 Operational Environment requirements are applicable because the module operates in a modifiable operational environment. The module was operational tested on the following Common Criteria evaluated platforms:

- Dell Optiplex GX400 running Windows 2000 Professional (SP 3)
- Dell Optiplex GX270 running Windows XP Professional (SP 2)

9. Security Rules
The OSC cryptographic module’s design corresponds to the module’s security rules. This section documents the security rules enforced by the cryptographic module to implement the security requirements of this FIPS 140-2 Level 2 module.

1. The Operating System must enforce authentication methods to prevent unauthorized access to the module. The passwords to authenticate to the Operating System must be at least six characters long (chosen from a 52 character set).
2. The cryptographic module shall perform the following tests:
   A. Power up Self-Tests:
      1. Cryptographic algorithm tests:
         a. AES KAT
         b. AES CCM KAT
         c. TDES KAT
         d. RSA Sign/Verify KAT
         e. RSA Encrypt/Decrypt KAT (for key transport only)
         f. DSA Sign/Verify KAT
         g. HMAC-SHA-1, HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, and
            HMAC-SHA-512 KATs
         h. SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512 KATs
         i. FIPS 186-2 DRNG KAT
      2. Software Integrity Test (HMAC-SHA-512)
   B. Conditional Self-Tests:
      1. Continuous Random Number Generator (RNG) test – performed on DRNG
      2. DSA pairwise consistency test
      3. RSA pairwise consistency test
   3. The operator shall be capable of commanding the module to perform the power-up self-test
      by reloading the module into memory or by calling the odFIPS_RunSelfTestAsynch
      function.
   4. Prior to each use, the internal RNG shall be tested using the conditional test specified in
      FIPS 140-2 §4.9.2.
   5. Data output shall be inhibited during key generation, self-tests, zeroization, and error states.
   6. Status information shall not contain CSPs or sensitive data that if misused could lead to a
      compromise of the module.
   7. The module does not support concurrent operators.

10. Physical Security
The FIPS 140-2 Area 5 Physical Security requirements are not applicable because the device is a
    software only module.

11. Mitigation of Other Attacks Policy
The module has not been designed to mitigate any specific attacks outside the scope of FIPS
    140-2 requirements.
12. Definitions and Acronyms

AES  Advanced Encryption Standard
API  Application Program Interface
CC  Common Criteria
CCEVS Common Criteria Evaluation and Validation Scheme
CO  Cryptographic Officer
CSP  Critical Security Parameter
DLL  Dynamic Link Library
DRNG Deterministic Random Number Generator
DSA  Digital Signature Algorithm
EAL  Evaluation Assurance Level
EMC  Electromagnetic Compatibility
EMI  Electromagnetic Interference
FIPS Federal Information Processing Standard
GPC  General Purpose Computer
HMAC Keyed-Hash Message Authentication Code
OSC Odyssey Security Component
RAM  Random Access Memory
RNG Random Number Generator
RSA Rivest, Shamir and Adleman Algorithm
TDES Triple-DES
SHA Secure Hash Algorithm