# DINAMO Networks, Inc.

## DINAMO Pocket Hardware Security Module

Firmware Version: 5.0.8.0

`

# FIPS 140-2 Non-Proprietary Security Policy

**FIPS Security Level: 2**
**Document Version: 0.10**

**Prepared for:**

**DINAMO Networks, Inc.**
United Nations Avenue, 14401
ED. TARUMA
Sao Paulo

Phone: +55 11 3304 3120
www.dinamonetworks.com

**Prepared by:**

**Corsec Security, Inc.**
13921 Park Center Road, Suite 460
Herndon, VA 20171
United States of America

Phone: +1 703 267 6050
www.corsec.com

# Table of Contents

# List of Tables

# List of Figures

# 1. Introduction

## 1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the DINAMO Pocket Hardware Security Module from DINAMO Networks, Inc. (DINAMO). This Security Policy describes how the DINAMO Pocket Hardware Security Module meets the security requirements of Federal Information Processing Standards (FIPS) Publication 140-2, which details the U.S.[1] 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 the National Institute of Standards and Technology (NIST) and the Canadian Centre for Cyber Security (CCCS).

This document also describes how to run the module in validated configuration. This policy was prepared as part of the Level 2 FIPS 140-2 validation of the module. The DINAMO Pocket Hardware Security Module is referred to in this document as the DINAMO Pocket HSM, HSM, or 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 DINAMO website (https://www.dinamonetworks.com/en/dinamo/) contains information on the full line of products from DINAMO.
- The search page on the CMVP website (https://csrc.nist.gov/Projects/cryptographic-module-validation-program/Validated-Modules/Search) can be used to locate and obtain vendor contact information for technical or sales-related questions about the module.

## 1.3 Document Organization

The Security Policy document is organized into two primary sections. Section 2 provides an overview of the validated module. This includes a general description of the capabilities and the use of cryptography, as well as a presentation of the validation level achieved in each applicable functional area of the FIPS standard. It also provides high-level descriptions of how the module meets FIPS requirements in each functional area. Section 3 documents the guidance needed for the secure use of the module, including initial setup instructions and management methods and applicable usages policies.

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

[1] U.S. – United States

# 2. DINAMO Pocket HSM

## 2.1 Overview

DINAMO has been providing solutions to the Information Security segment for over fifteen years. DINAMO Pocket is part of a family of DINAMO Hardware Security Modules (HSMs) that reduce risk and operation costs by centralizing enterprise cryptographic key management.

The DINAMO Pocket HSM is a network-attached device that offers a secure environment for the storage and lifecycle management of cryptographic keys, as well as offering cryptographic services such as encryption, digital signatures, key generation, and authentication. It is a mini HSM that generates, stores and protects cryptographic keys and provides more performance for using digital certificates.

The DINAMO Pocket HSM is shown in Figure 1.

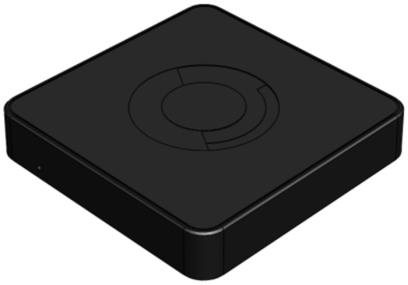


**Figure 1 – DINAMO Pocket**

The DINAMO Pocket HSM has one 100 Base-T Ethernet management interface and a Micro SD[2] card to store keys. Management of the HSM is accomplished via the following methods:

- DinamoCon, which is a fat client accessible via a browser using HTTPS[3] over Ethernet. The DinamoCon is accessible only to Crypto Officers and is used primarily for initialization, activation, and configuration of the module.

- Remote Console, which is accessible remotely via a cleartext session or TLS[4] v1.2 over the Ethernet interface. Certain services require a TLS v1.2 session (e.g., import/export of keys). These services will be blocked if attempted over a cleartext session. The services that require a TLS v1.2 session are indicated as such in Table 6 and Table 7. The Remote Console is accessible to all operators and is used for appliance management (Crypto Officers) and obtaining key management and cryptographic services (all operators).

- HTTPS Console, which is accessible remotely via HTTPS over the Ethernet interface and offers the same services as the Remote Console, with TLS v1.2 mandatory. The HTTPS Console is accessible to all operators

[2] SD – Secure Digital
[3] HTTPS – Hyper Text Transfer Protocol Secure
[4] TLS – Transport Layer Security

and is used for appliance management (Crypto Officers) and obtaining key management and cryptographic services (all operators).

Standard APIs, including MS[5] Crypto API, Java JCA[6]/JCE[7], PKCS[8]#11, and Native API are also available for integration with the HSM. Additionally, replication is provided between modules.

The DINAMO Pocket HSM is validated at the FIPS 140-2 section levels shown in Table 1.

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

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

## 2.2 Module Specification

The DINAMO Pocket HSM is a hardware cryptographic module with a multiple-chip standalone embodiment. The overall security level of the module is 2. The cryptographic boundary is defined by the physical enclosure of the HSM and includes all internal hardware as well as the HSM (Version 5.0.8.0) firmware.

The main hardware components consist of a processor, memories, Micro SD card, and the enclosure containing all of these components.

The module includes the cryptographic algorithm providers listed in Table 2.

[5] MS – Microsoft

[6] JCA – Java Cryptography Architecture

[7]CE – Java Cryptography Extension

[8] PKCS – Public Key Cryptography Standards

[9] N/A – Not Applicable

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

**Table 2 – Cryptographic Algorithm Providers**

| Certificate Number | Implementation Name | Version | Use |
|---|---|---|---|
| C1902 | DINAMO Hardware Security Module Cryptographic Library | 1.0 | Firmware-based cryptographic primitives (based on OpenSSL 1.1.1a) |
| C1905 | DINAMO Hardware Security Module KBKDF[11] and RSA[12] KeyGen | 1.0 | NIST SP[13] 800-108 KBKDF implementation and an RSA Key Generation implementation |
| C1906 | DINAMO Hardware Security Module TLS KDF[14] | 1.0 | TLS v1.2 KDF implementations (based on OpenSSL 1.1.1a) |

The module implements the FIPS-Approved algorithms listed in Table 3 below.

**Table 3 – FIPS Approved Algorithm Implementations**

| Certificate Number | Algorithm | Standard | Mode / Method | Key Lengths / Curves / Moduli | Use |
|---|---|---|---|---|---|
| C1902 | AES[15] | FIPS PUB[16] 197<br>NIST SP 800-38A | CBC[17], CTR[18], ECB[19] | 128, 192, 256 | encryption/decryption |
| | | NIST SP 800-38B | CMAC[20] | 128, 192, 256 | generation/verification |
| | | NIST SP 800 38D | GCM[21] | 128, 192, 256 | encryption/decryption |
| Vendor Affirmed | CKG[22] | NIST SP 800-133 | - | - | symmetric key generation<br>*Symmetric keys and generated seeds are produced using unmodified output from the Approved DRBG.* |
| Vendor Affirmed | KAS-SSC[23] | NIST SP 800-56Arev3 | ECDH[24] | P-224, P-256, P-384, P-521 | Key Agreement Scheme – shared secret computation per SP 800-56Arev3 and Key Derivation per SP 800-135rev1 (Certs. #C1902 and #C1906)<br>*The module implements P-521 but does not use it operationally.* |

[11] KBKDF – Key-Based Key Derivation Function
[12] RSA – Rivest Shamir Adleman
[13] SP – Special Publication
[14] KDF – Key Derivation Function
[15] AES – Advance Encryption Standard
[16] PUB – Publication
[17] CBC – Cipher Block Chaining
[18] CTR – Counter
[19] ECB – Electronic Codebook
[20] CMAC – Cipher-Based Message Authentication Code
[21] GCM – Galois Counter Mode
[22] CKG – Cryptographic Key Generation
[23] KAS-SSC – Key Agreement Scheme - Shared Secret Computation
[24] ECDH – Elliptic Curve Diffie-Hellman

| Certificate Number | Algorithm | Standard | Mode / Method | Key Lengths / Curves / Moduli | Use |
|---|---|---|---|---|---|
| C1905 | KBKDF | NIST SP 800-108 | Counter mode | HMAC SHA2-256, HMAC SHA2-512 | key derivation |
| C1902 | CVL | NIST SP 800-135rev1 | ANS X9.63-2001 | - | key derivation<br>*No part of the ANS X9.63-2001 protocol, other than the KDF, has been tested by the CAVP[25] and CMVP.* |
| C1906 | CVL | NIST SP 800-135rev1 | TLS v1.2 | - | key derivation<br>*No part of the TLS protocol, other than the KDF, has been tested by the CAVP and CMVP.* |
| C1902 | DRBG[26] | NIST SP 800-90Arev1 | CTR-based | 256-bit AES | deterministic random bit generation |
| C1902 | ECDSA[27] | FIPS PUB 186-4 | PKG[28] | P-224, P-256, P-384, P-521 | key pair generation<br>*The module implements P-521 but does not use it operationally.* |
| | | | PKV[29] | P-192, P-224, P-256, P-384, P-521 | key pair verification<br>*The module implements P-521 but does not use it operationally.* |
| | | | SigGen | P-224, P-256, P-384, P-521 | digital signature generation<br>*The module implements P-521 but does not use it operationally.* |
| | | | SigVer | P-192, P-224, P-256, P-384, P-521 | digital signature verification<br>*The module implements P-521 but does not use it operationally.* |
| C1902 | HMAC[30] | FIPS PUB 198-1 | SHA2[31]-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512 | SHA2-224, SHA2-256, SHA2-384, SHA2-512, SHA3-224, SHA3-256, SHA3-384, SHA3-512 | message authentication |
| Vendor Affirmed | PBKDF2[32] | NIST SP 800-132 | Option 1a with HMAC SHA2-256 | - | password-based key derivation |

[25] CAVP – Cryptographic Algorithm Validation Program

[26] DBRG – Deterministic Random Bit Generator

[27] ECDSA – Elliptic Curve Digital Signature Algorithm

[28] PKG – Public Key (Q) Generation

[29] PKV – Public Key (Q) Validation

[30] HMAC – (keyed-) Hashed Message Authentication Code

[31] SHA – Secure Hash Algorithm

[32] PBKDF2 – Password-based Key Derivation Function 2

| Certificate Number | Algorithm | Standard | Mode / Method | Key Lengths / Curves / Moduli | Use |
|---|---|---|---|---|---|
| C1902 | KTS[33] | NIST SP 800 38D | AES | 128, 192, 256 | *This implementation has been tested for its compliance with AES GCM and this mode of AES is used for key wrapping.* |
| C1905 | RSA | FIPS PUB 186-4 | - | 2048, 3072 | key pair generation |
| C1902 | RSA | FIPS PUB 186-4 | ANSI X9.31<br>PKCS1.5<br>PSS[34] | 2048, 3072 (SHA2-224, SHA2-256, SHA2-384, SHA2-512) | digital signature generation |
| | | | ANSI X9.31<br>PKCS 1.5<br>PSS | 2048, 3072 (SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512) | digital signature verification |
| C1902 | SHS[35] | FIPS PUB 180-4 | SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512 | - | message digest |
| C1902 | SHA3 | FIPS PUB 202 | SHA3-224, SHA3-256, SHA3-384, SHA3-512 | - | message digest |
| C1902 | Triple-DES[36] | NIST SP 800-67rev2 | TCBC, TECB | Keying option 1 | decryption |

While the module includes CAVP-tested implementations of Triple-DES Keying option 1 (CBC, ECB) encryption and Triple-DES CMAC (#C1902), these algorithms are not used for any security-relevant functions.

The vendor affirms the following cryptographic security methods:

- Password-based key derivation – The module performs PBKDF2 in compliance with *NIST SP 800-132* using option 1(a) in Section 5.4 to derive the following keys: AES KEK[37], Triple-DES Decryption Key, and Backup Key. The PBKDF2 is used for storage applications only. HMAC SHA-256 is used as the approved PRF[38]. The iteration count is 16384 iterations. The length of the salt is 512 bits, and it is generated by the FIPS-Approved DRBG. Please refer to Section 3.2.1 for Crypto Officer guidance specific to this function.

- Symmetric key generation – Per *NIST SP 800-133*, the module uses the FIPS-Approved CTR-based DRBG specified in *NIST SP 800-90Arev1* to generate cryptographic keys. The resulting symmetric key or generated seed is an unmodified output from the DRBG. The module's DRBG is seeded via /dev/random, a non-deterministic random number generator (NDRNG) internal to the module.

- Key agreement scheme (shared secret computation) – The module implements an ECC CDH shared secret computation for its ECDH key agreement scheme. The shared secret computation is compliant with section 5.7.1.2 of *NIST SP 800-56Arev3*. This compliance claim follows scenario X1 of section D.8 of the *Implementation Guidance for FIPS PUB 140-2 and the CMVP*. This primitive is used by the Full Unified

[33] KTS – Key Transport
[34] PSS – Probabilistic Signature Scheme
[35] SHS – Secure Hash Standard
[36] DES – Data Encryption Standard
[37] KEK – Key Encryption Key
[38] PRF – Pseudo-Random Function

Model, Ephemeral Unified Model, One-Pass Unified Model, One-Pass Diffie-Hellman, and Static Unified Model schemes found in section 6 of that recommendation.

*Note that vendor affirmation of the KAS-SSC with NIST-recommended elliptic curves is included above in compliance with section A.2 of the Implementation Guidance for FIPS PUB 140-2 and the CMVP to support the allowed use of non-NIST-recommended elliptic curves in the ECDSA signature algorithm and the ECDH key agreement scheme in the approved mode of operation. The non-NIST-recommended curves implemented by the module are referenced in Table 4.*

The module implements the non-Approved but allowed algorithms shown in Table 4.

**Table 4 – Allowed Algorithm Implementations**

| Algorithm | Caveat | Use |
|---|---|---|
| EC Diffie-Hellman with non-NIST-recommended curves | key establishment methodology provides between 112 and 256 bits of encryption strength as follows:<br>• BrainpoolP224r1 (112-bits)<br>• BrainpoolP256r1 (128 bits)<br>• BrainpoolP320r1 (160 bits)<br>• BrainpoolP384r1 (192 bits)<br>• BrainpoolP512r1 (256 bits) | Key agreement |
| ECDSA with non-NIST-recommended curves | key establishment methodology provides between 112 and 256 bits of encryption strength as follows:<br>• BrainpoolP224r1 (112-bits)<br>• BrainpoolP256r1 (128 bits)<br>• BrainpoolP320r1 (160 bits)<br>• BrainpoolP384r1 (192 bits)<br>• BrainpoolP512r1 (256 bits) | Key pair generation, digital signature generation, digital signature verification |
| NDRNG | - | Seeding for the DRBG |
| RSA | Key establishment methodology provides between 112 and 128 bits of encryption strength | Key encapsulation/establishment - key transport in TLSv1.2 (NIST SP 800-56Brev1, section 9.2.3)<br>Key transport (PKCS#1 v2.2 RSAES_OAEP) |
| SHA-1 | Legacy-use | ECDSA and RSA signature verification |
| 2-key Triple-DES | Legacy-use | Decryption |

## 2.3 Module Ports and Interfaces

The module’s design separates the physical ports and interfaces into four logically distinct and isolated categories.

They are:

- Data Input Interface
- Data Output Interface

- Control Input Interface
- Status Output Interface

The DINAMO Pocket contains the physical ports and interfaces shown in Figure 2 and Figure 3.

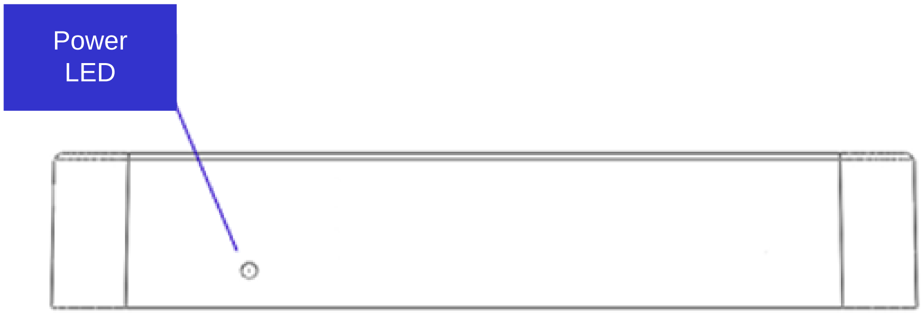


**Figure 2 – DINAMO POCKET Ports and Interfaces (Front)**

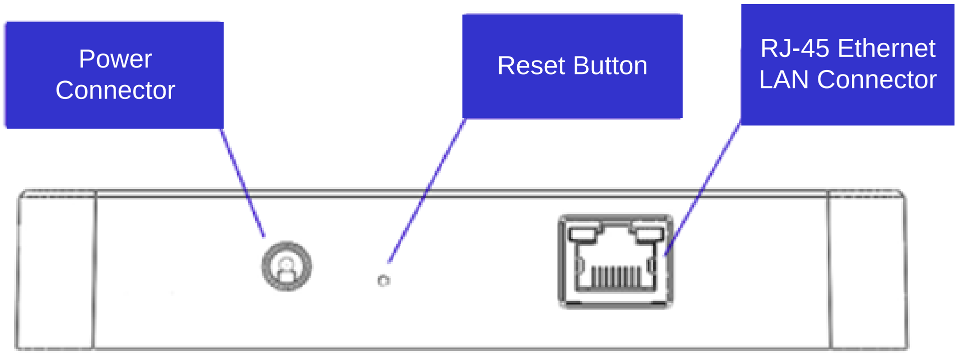


**Figure 3 – DINAMO POCKET Ports and Interfaces (Rear)**

Table 5 provides the mapping from the physical interfaces to logical interfaces as defined by FIPS 140-2 for the DINAMO Pocket.

**Table 5 – FIPS 140-2 Logical Interface Mappings for the DINAMO Pocket**

| Physical Port/Interface | Quantity | Description | FIPS 140-2 Logical Interface |
|---|---|---|---|
| Front Panel | | | |
| Power LED | 1 | Power status indicator:<br>• Solid blue = System on<br>• Off = No power present | • Status Output |
| Rear Panel | | | |

| Physical Port/Interface | Quantity | Description | FIPS 140-2 Logical Interface |
|---|---|---|---|
| RJ-45 Ethernet 10/100 Mbps LAN[39] connector | 1 | Interface to DinamoCon, Remote Console, and HTTPS Console for cryptographic services | • Data Input<br>• Data Output<br>• Control Input<br>• Status Output |
| Reset Button | 1 | Used to reset Wi-Fi (Wi-Fi is disabled in FIPS-Approved mode) | • Control Input |
| Power connector | 1 | Power connector | • Power Input |

## 2.4 Roles and Services

The sections below describe the module's roles and services and define any authentication methods employed.

### 2.4.1 Authorized Roles

The module supports two roles that operators may assume:

- Crypto Officer (CO) role – The CO is responsible for initializing the module for first use. The CO has all permissions over the module and access to all services provided over the DinamoCon, Remote Console, and HTTPS Console. The CO has its own cryptographic key partition in which to store keys. The CO also has the ability to give Users specific system permissions that allow them to perform limited management functionality (including creating and removing users, listing users, accessing log records, and creating and restoring backups). This is done by enabling these system permissions in the User's account using the Remote Console **Create** or **Attributes** menu items.

- User role – The User creates, removes, imports, exports, performs cryptographic services with, and grants permissions to keys in its own user partition. The User does not manage the module unless given specific system permissions by the CO as described above. The User accesses the module only via the Remote Console and HTTPS Console.

The module implements two types of authentication:

- DinamoCon authentication - At the DinamoCon, the module implements identity-based authentication. Only Crypto Officers can authenticate over this interface. Authentication requires the entry of two 48 hexadecimal character secrets created when the module is initialized. Both secrets can be held by a single Crypto Officer, but the secrets can be distributed among two Crypto Officers in order to further control access to the module.

- Remote Console and HTTPS Console authentication - At the Remote Console and HTTPS Console, the module implements identity-based authentication using username and password to individually identify

[39] LAN – Local Area Network

each operator and explicitly select the role assigned to that operator. The Remote Console and HTTPS Console are enabled only after CO authentication at the DinamoCon, after services are enabled.

### 2.4.2 Strength of Authentication Mechanisms

The strength of the authentication mechanism is such that for each attempt to use the authentication mechanism, the probability shall be less than one in 1,000,000 that a random attempt will succeed or a false acceptance will occur. Details are provided below for both the DinamoCon and Remote Console or HTTPS Console authentication:

- DinamoCon authentication - Crypto Officers authenticate to the DinamoCon using two secrets. Each secret has a length of 48 hexadecimal characters, with any combination of 16 symbols; thus, the probability that a random attempt will succeed is 1 in $16^{96}$ = 1: 3.94 x $10^{115}$.

Assuming 60 attempts can be performed in a one minute period at the DinamoCon, the probability of a successful random attempt during this period is:
1: ($16^{96}$ possible attempts / 60 attempts per minute)
1: 6.57 x $10^{113}$

- Remote Console and HTTPS Console authentication - Operators authenticate to the Remote Console and HTTPS Console of the module with a username and password. Each password is a minimum of 8 characters, which is enforced by the module. Each password can contain any combination of upper- and lower-case letters [A-z, a-z] and numbers [0-9]. Each character of the 8-character password could be 1 of 62 printable ASCII[40] characters, providing for a password strength of (1/$62^{8}$ =) 1 in 218,340,105,584,896.

The module allows for seven consecutive failed authentication attempts at the Remote Console and HTTPS Console before an operator (CO or User) is blocked and can no longer authenticate until the operator is unblocked by a CO via the Remote Console. Hence at most seven password attempts can be made in a one minute period. Therefore, the probability that a random attempt will succeed or a false acceptance will occur in one minute is:
1: ($62^{8}$ possible passwords / 7 passwords per minute)
1: 31,191,443,654,985

This is less than one in 100,000, as required by FIPS 140-2.

### 2.4.3 Operator Services

Descriptions of the services available to the CO role and User role are provided in Table 6 and Table 7. Please note that the keys and Critical Security Parameters (CSPs) listed in Table 6 and Table 7 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.

[40] ASCII – American Standard Code for Information Interchange

The CO has access to the services listed in Table 6 using the DinamoCon. CO and User services available using the Remote Console and HTTPS Console are listed in Table 7.

**Table 6 – Operator Services via the DinamoCon**

| Service | Operator | | Description | Input | Output | CSP and Type of Access |
|---|---|---|---|---|---|---|
| | CO | User | | | | |
| Initialize module | ✓ | | Perform module Initialization<br>(at first boot from factory the Master Key is created within the module and its components output encrypted to COs for subsequent activations of the module) | Command and parameters | Command response; status output | Master Key – R/W/X<br>Master Key Component Key 1 – R/W/X<br>Master Key Component Key 2 – R/W/X<br>Replication TLS-PSK Key – W<br>Data Protection Key – W<br>AES GCM IV[41] – W/X<br>DRBG Seed – R/W/X<br>Entropy Input String – R/X<br>DRBG 'V' Value – R/W/X<br>DRBG Key Value – R/W/X |
| Establish TLS session | ✓ | ✓ | Establish DinamoCon session using TLS protocol | Command | Command response/ Status output | TLS Public Key – W/X<br>TLS Private Key – W/X<br>TLS Bundle Public Key – W/X<br>TLS Bundle Private Key – W/X<br>ECDH Private Component – X<br>ECDH Public Component – R/X<br>TLS Pre-Master Secret – W/X<br>TLS Master Secret – W/X<br>TLS Session Key – R/W/X<br>TLS Authentication Key – W/X<br>AES GCM IV – W/X<br>DRBG Seed – R/W/X<br>Entropy Input String – R/X<br>DRBG 'V' Value – R/W/X<br>DRBG Key Value – R/W/X |
| Start service or Stop service | ✓ | | Enable or disable module services available through the Remote Console and HTTPS Console | Command | Status output | Master Key Component Key 1 – R/X<br>Master Key Component Key 2 – R/X |
| Shutdown | ✓ | | Shutdown module | Command | Status output | All ephemeral keys/CSPs – W<br>Master Key Component Key 1 – R/X<br>Master Key Component Key 2 – R/X |
| Reboot | ✓ | | Reboot module | Command | Command response; status output | All ephemeral keys/CSPs – W<br>Master Key Component Key 1 – R/X<br>Master Key Component Key 2 – R/X |

[41] IV – Initialization Vector

| Service | Operator | | Description | Input | Output | CSP and Type of Access |
|---|---|---|---|---|---|---|
| | CO | User | | | | |
| Configure network parameters | ✓ | | Configure network parameters | Command and parameters | Command response; status output | Master Key Component Key 1 – R/X<br>Master Key Component Key 2 – R/X |
| Perform self-tests on demand | ✓ | | Perform on-demand self-tests by issuing "reboot" command | Command | Status output | Master Key Component Key 1 – R/X<br>Master Key Component Key 2 – R/X |
| Zeroize keys | ✓ | | Zeroize keys and CSPs via Reboot, Shutdown, or Reset HSM Database commands | Command | Command response;<br>Status output | All ephemeral keys/CSPs – W<br>Master Key Component Key 1 – R/X<br>Master Key Component Key 2 – R/X |
| Reset HSM Database | ✓ | | Return module to factory state | Command | Command response; status output | Master Key Component Key 1 – R/X<br>Master Key Component Key 2 – R/X<br>All ephemeral and persistent keys/CSPs – W |

**Table 7 – Operator Services via the Remote Console and HTTPS Console**

| Service | Operator | | Description | Input | Output | CSP and Type of Access |
|---|---|---|---|---|---|---|
| | CO | User | | | | |
| Show status | ✓ | | Display the current mode of the module | Command | Status output | None |
| Manage users | ✓ | ✓* | List users; create or remove a user (removing a user deletes all keys and CSPs in the user's partition)<br>Creating a user requires TLS v1.2 session to Remote Console | Command and parameters | Command response; status output | CO Password – R/W<br>User Password – R/W<br>All persistent keys and CSPs in partition of removed user – W |
| View logs | ✓ | ✓* | View module log data | Command | Command response; status output | None |
| Generate AES key | ✓ | ✓ | Generate and return an AES key<br>Requires TLS v1.2 session to Remote Console | Command and parameters | AES key, status output | AES Key – W<br>AES GCM IV – W/X<br>Data Protection Key – R/X |

| Service | Operator | | Description | Input | Output | CSP and Type of Access |
|---|---|---|---|---|---|---|
| | CO | User | | | | |
| Generate asymmetric key pair | ✓ | ✓ | Generate and return the specified type of asymmetric key pair (RSA or ECDSA)<br>Requires TLS v1.2 session to Remote Console | Command and parameters | Key, status output | RSA Public Key – W<br>RSA Private Key – W<br>ECDSA Public Key – W<br>ECDSA Private Key – W<br>Data Protection Key – R/X |
| Destroy key | ✓ | ✓ | Deletes a cryptographic key from a user or CO partition | Key id | Status output | Key associated with Key id – R/W<br>Data Protection Key – R/X |
| Perform symmetric encryption | ✓ | ✓ | Encrypt plaintext data using specified key id<br>Requires TLS v1.2 session to Remote Console | Command, parameters, and plaintext | Ciphertext, status output | AES Key – R/X<br>Data Protection Key – R/X |
| Perform symmetric decryption | ✓ | ✓ | Decrypt ciphertext using specified key id | Command, parameters, and ciphertext | Plaintext, status output | AES Key – R/X<br>Data Protection Key – R/X |
| Generate Signature | ✓ | ✓ | Generate and return a digital signature for supplied message using specified key id | Command, parameters, and message | Digital signature, status output | RSA Private Key – R/X<br>ECDSA Private Key – R/X<br>Data Protection Key – R/X |
| Verify Signature | ✓ | ✓ | Verify a digital signature on supplied message using specified key id | Command, parameters, and message | Command response; status output | RSA Public Key – R/X<br>ECDSA Public Key – R/X<br>Data Protection Key – R/X |
| Generate Hash | ✓ | ✓ | Generate and return hash using specified hash type | Command, parameters, and message | Hash, status output | None |
| Generate keyed hash (HMAC) | ✓ | ✓ | Generate and return message authentication code using specified HMAC type | Command, parameters, and message | Hash, status output | HMAC Key – R/X<br>Data Protection Key – R/X |
| Import key | ✓ | ✓ | Import certificates and keys using specified KEK or RSA key pair and import method<br>Requires TLS v1.2 session to Remote Console | Command, parameters, and key material | Status output | AES KEK – R/W/X<br>AES GCM KEK – R/W/X<br>AES GCM IV – W/X<br>RSA Public Key – R/W<br>RSA Private Key – R/W<br>PBKDF Import/Export Password – R/X<br>Triple-DES Decryption Key – R/W/X<br>Data Protection Key – R/X |

| Service | Operator | | Description | Input | Output | CSP and Type of Access |
|---|---|---|---|---|---|---|
| | CO | User | | | | |
| Export key | ✓ | ✓ | Export certificates and keys using specified KEK or RSA key pair and export method<br>Requires TLS v1.2 session to Remote Console | Command and parameters | AES Key; RSA Public/Private Key (PKCS#7, PKCS#8, PKCS#12); ECDSA Public/Private Key; status output | AES KEK – R/W/X<br>AES GCM KEK – R/W/X<br>AES GCM IV – W/X<br>RSA public key – R/W<br>RSA private key – R/W<br>PBKDF2 Import/export password – R/X<br>Data Protection Key – R/X |
| List keys | ✓ | ✓ | List the keys in a partition | Command | Status output | None |
| Change permissions | ✓ | ✓ | Change permissions on keys in partition | Command and parameters | Status output | None |
| Change one's own password | ✓ | ✓ | Modify existing login passwords | Command and parameters | Status output | CO Password – R/W<br>User Password – R/W |
| Establish TLS session | ✓ | ✓ | Establish Remote Console session using TLS protocol | Command | Command response/ Status output | TLS Public Key – W/X<br>TLS Private Key – W/X<br>TLS Bundle Public Key – W/X<br>TLS Bundle Private Key – W/X<br>ECDH Private Component – X<br>ECDH Public Component – R/X<br>TLS Pre-Master Secret – W/X<br>TLS Master Secret – W/X<br>TLS Session Key – R/W/X<br>TLS Authentication Key – W/X<br>AES GCM IV – W/X<br>DRBG Seed – R/W/X<br>Entropy Input String – R/X<br>DRBG 'V' Value – R/W/X<br>DRBG Key Value – R/W/X |
| Set TLS bundle (HTTPS Console only) | ✓ | | Import a TLS bundle (certificate and private key) | Command/Data | Command response/Status output | TLS Bundle Public Key – W/X<br>TLS Bundle Private Key – W/X |
| Perform global backup | ✓ | ✓* | Backup HSM database<br>Requires TLS v1.2 session to Remote Console | Command and parameters | HSM global encrypted backup; status output | Data Protection Key – R/X<br>PBKDF2 Backup Password – R/X<br>Backup Key – W/X |
| Perform global restore | ✓ | ✓* | Restore HSM database<br>Requires TLS v1.2 session to Remote Console | Command and parameters, local backup file to read | Status output | Data Protection Key – R/X<br>PBKDF2 Backup Password – R/X<br>Backup Key – W/X |

✓*: these services are only available if the CO has enabled permission on the User's account, as described in Section 2.4.1.

The module provides a limited number of services for which the operator is not required to assume an authorized role. Table 8 lists the services for which the operator is not required to assume an authorized role. None of these services disclose or substitute cryptographic keys and CSPs or otherwise affect the security of the module.

**Table 8 – Additional Services**

| Service | Description | Input | Output | CSP and Type of Access |
|---|---|---|---|---|
| Zeroize | Zeroize ephemeral keys and CSPs | Power cycle | Status output | All ephemeral keys/CSPs – W |
| Perform on-demand self-tests | Perform self-tests on demand | Power cycle | Status output | All ephemeral keys/CSPs – W |
| Authenticate to DinamoCon | Authenticate CO operators to DinamoCon | Command and parameters | Status output | Master Key Component Key 1 – R/X<br>Master Key Component Key 2 – R/X |
| Authenticate to Remote Console or HTTPS Console | Authenticate operators to Remote Console or HTTPS Console | Command and parameters | Status output | Crypto Officer Password – X<br>User Password – X |
| About<br>(HTTPS Console Only) | Displays some system aspects | Command | Status Output | N/A |

## 2.5 Physical Security

The contents of the module, including all hardware, firmware, software, and data are protected via the following mechanisms:

- module enclosure – The module enclosure consists of a hard, opaque case made of an Acrylonitrile butadiene styrene (ABS) polymer. While the enclosure has no doors or removable covers, it is comprised of two parts secured together by internal snaps as shown in Figure 4. Any attempt to open the enclosure will result in permanent damage and will leave visible evidence of the attempt.

- tamper-evident seals – The module employs two tamper-evident seals consisting of thin gauged vinyl to cover the area adjoining the top and bottom parts of the enclosure on two sides, as shown in Figure 5 and Figure 6.

In addition, the DINAMO Pocket enclosure does not have any air inlet or outlet ventilation panels/holes that could allow for the direct observation of the internal components of the module.

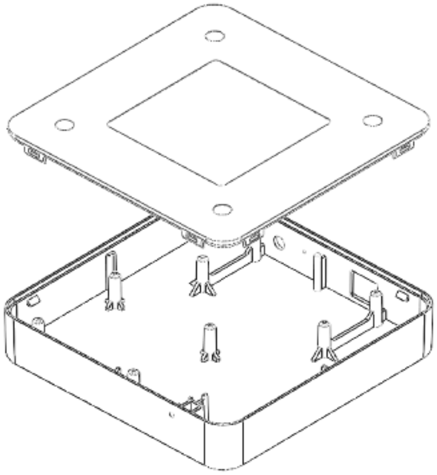


**Figure 4 – Module Enclosure**

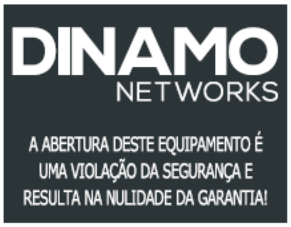
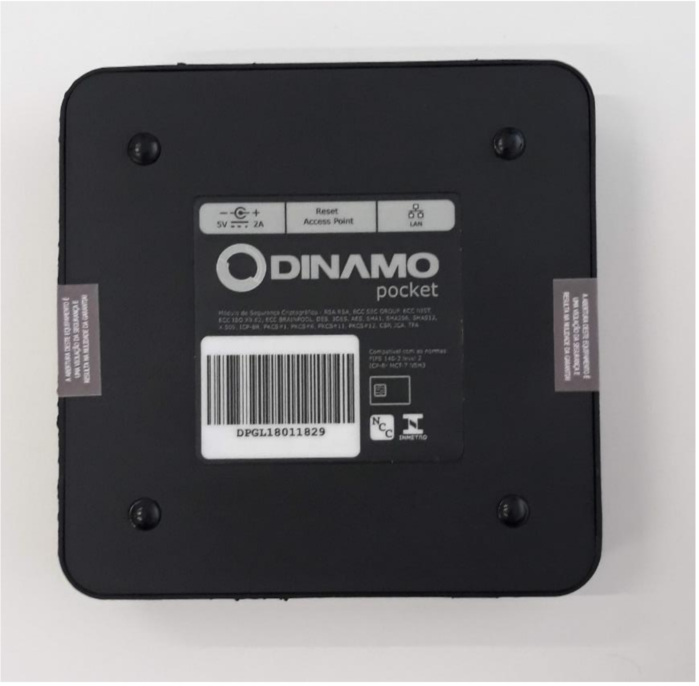

DINAMO
NETWORKS
A ABERTURA DESTE EQUIPAMENTO É
UMA VIOLAÇÃO DA SEGURANÇA E
RESULTA NA NULIDADE DA GARANTIA!

**Figure 5 – Tamper-Evident Seal**



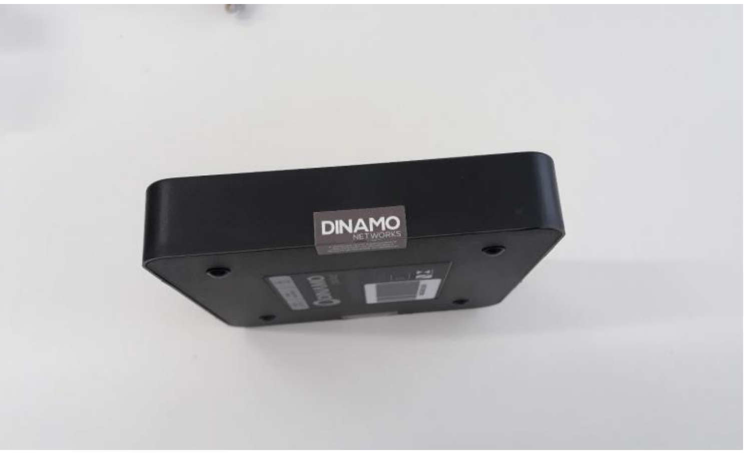


FIGURE 6 – EXAMPLE OF TAMPER-EVIDENT SEALS ADJOINING TOP AND BOTTOM OF POCKET ENCLOSURE

## 2.6 Operational Environment

The operational environment of the module does not provide a general-purpose OS to module operators.

The module employs a non-modifiable operating environment. The module's firmware is executed by a Broadcom BCM2387 SoC[42] with a 64-bit 1.2 GHz quad-core ARM Cortex-A53 processor on a Raspberry Pi 3B board running CentOS 7.2.

The module provides no mechanisms for operators to update the firmware or the operational environment. These activities can only be performed by the vendor.

[42] SoC – System on a Chip

## 2.7 Cryptographic Key Management

The module supports the CSPs listed below in Table 9.

**Table 9 – Cryptographic Keys, Cryptographic Key Components, and CSPs**

| CSP | CSP Type | Generation / Input | Output | Storage | Zeroization | Use |
|---|---|---|---|---|---|---|
| Replication TLS-PSK Key | 128-bit AES-CBC key | Derived from the Master Key via KBKDF per SP 800-108 | Never exits the module | Plaintext in volatile RAM | At end of TLS session, power cycle, shutdown, HSM database reset | Encryption/decryption of TLS sessions used for replication between modules |
| AES GCM IV | 96-bit value | Generated internally via FIPS-Approved DRBG with RBG construction per Section 8.2.2 of NIST SP 800-38D | Never exits the module | Plaintext in volatile RAM | Power cycle, shutdown, reboot, HSM database reset | IV for AES GCM function |
| AES GCM KEK | 128, 192, 256-bit AES-GCM key | Internally generated via Approved DRBG | Never exits the module | Plaintext in volatile RAM | Power cycle, shutdown, reboot, HSM database reset | Key transport |
| AES CMAC key | 128, 192, 256-bit AES-CMAC key | Internally generated via Approved DRBG | Never exits the module | Plaintext in volatile RAM | Power cycle, shutdown, reboot, HSM database reset | MAC generation and verification |
| Master Key components:<br>Master Key Component Key 1<br>Master Key Component Key 2 | 48 hexadecimal character (24-byte) components of Master Key | Generated internally via Approved DRBG (first boot or before creation of new Master Key)<br>Electronically input encrypted via DinamoCon | Exported encrypted via DinamoCon | Plaintext in volatile RAM | Power cycle, shutdown, HSM database reset | CO authentication to the module via DinamoCon<br>Recovering Master Key |

| CSP | CSP Type | Generation / Input | Output | Storage | Zeroization | Use |
|---|---|---|---|---|---|---|
| Master Key | 192-bit symmetric key | Generated internally from Master Key components via key combination per section 6.3 of NIST SP 800-133 (at first boot)<br>Recovered from Master Key components via key combination per section 6.3 of NIST SP 800-133 (subsequent boots) | Never exits the module | Plaintext in volatile RAM | Power cycle, shutdown, HSM database reset | Derivation of Data Protection Key |
| Data Protection Key | 256-bit AES-GCM key | Derived from the Master Key via KBKDF per SP 800-108 | Never exits the module | Plaintext in volatile RAM | Power cycle, shutdown, HSM database reset | Encryption/decryption of all data in persistent storage |
| ECDH Private Component | Private component of ECDH: P-256 BrainpoolP224r1, BrainpoolP256r1, BrainpoolP320r1, BrainpoolP384r1, BrainpoolP512r1 | Generated internally using FIPS-Approved DRBG | Never exits the module | Plaintext in volatile RAM | At end of TLS session, power cycle, shutdown, reboot, HSM database reset | Generation of TLS shared secrets |
| ECDH Public Component | Public component of ECDH: P-256 BrainpoolP224r1, BrainpoolP256r1, BrainpoolP320r1, BrainpoolP384r1, BrainpoolP512r1 | Generated internally using FIPS-Approved DRBG | Exits in plaintext form | Plaintext in volatile RAM | At end of TLS session, power cycle, shutdown, reboot, HSM database reset | Generation of TLS shared secrets |
| TLS Private Key | 2048 or 3072-bit RSA private key<br>P-256 ECDSA private key | Generated internally via FIPS-Approved DRBG | Never exits the module | Plaintext in volatile RAM | At end of TLS session, power cycle, shutdown, reboot, HSM database reset | TLS authentication |

| CSP | CSP Type | Generation / Input | Output | Storage | Zeroization | Use |
|---|---|---|---|---|---|---|
| TLS Bundle Private Key | 2048 or 3072-bit RSA private key<br>P-256 ECDSA private key | Generated externally and imported in encrypted form via HTTPS Console using Set TLS bundle command | Never exits the module | Encrypted (via Data Protection Key) on SSD | N/A[43] | TLS authentication |
| TLS Public Key | 2048 or 3072-bit RSA public key<br>P-256 ECDSA public key | Generated internally via FIPS-Approved DRBG | Exits the module in plaintext form | Plaintext in volatile RAM | N/A | TLS authentication |
| TLS Bundle Public Key | 2048 or 3072-bit RSA public key<br>P-256 ECDSA public key | Generated externally and imported via HTTPS Console | Exits the module in plaintext form | Encrypted (via Data Protection Key) on SSD | N/A | TLS authentication |
| TLS Pre-Master Secret | [for RSA cipher suites] 384-bit random value<br>[for ECDH cipher suites] ECDH shared secret | [for RSA cipher suites] Generated externally and imported in encrypted form via RSA key transport<br>[for ECDH cipher suites] Derived internally via ECDH shared secret computation | Never exits the module | Plaintext in volatile RAM | Upon completion of TLS Master Secret computation, power cycle, shutdown, reboot, HSM database reset | Derivation of the TLS Master Secret |
| TLS Master Secret | 256 or 384-bit shared secret | Derived internally using the TLS Pre-Master Secret via TLS KDF | Never exits the module | Plaintext in volatile RAM | At end of TLS session, power cycle, shutdown, reboot, HSM database reset | Derivation of the TLS Session Key and TLS Authentication Key |
| TLS Session Key | 128 or 256-bit AES key<br>128 or 256-bit AES GCM key | Derived internally using the TLS Master Secret via TLS KDF | Never exits the module | Plaintext in volatile RAM | At end of TLS session, power cycle, shutdown, reboot, HSM database reset | Encryption and decryption of TLS session packets |

[43] N/A – Not Applicable

| CSP | CSP Type | Generation / Input | Output | Storage | Zeroization | Use |
|---|---|---|---|---|---|---|
| TLS Authentication Key | 160, 256, or 384-bit HMAC key | Derived internally using the TLS Master Secret via the TLS KDF | Never exits the module | Plaintext in volatile RAM | At end of TLS session, power cycle, shutdown, reboot, HSM database reset | Authentication of TLS session packets |
| DRBG Seed | 384-bit value | Generated internally using entropy input string | Never exits the module | Plaintext in volatile RAM | Power cycle, shutdown, reboot, reset HSM database | Seeding material for DRBGs |
| Entropy Input String | 256-bit value | Generated internally | Never exits the module | Plaintext in volatile RAM | End of DRBG function, power cycle, shutdown, reboot, HSM database reset | Entropy material for SP 800-90A DRBG |
| DRBG Key Value | Internal DRBG state value<br>256 bits | Generated internally | Never exits the module | Plaintext in volatile RAM | Power cycle, shutdown, reboot, HSM database reset | Random number generation |
| DRBG 'V' Value | Internal DRBG state value<br>128 bits | Generated internally | Never exits the module | Plaintext in volatile RAM | Power cycle, shutdown, reboot, HSM database reset | Random number generation |
| CO Password | Alphanumeric string<br>Minimum of eight (8) characters | Input electronically in encrypted form via Remote Console or HTTPS Console (over TLS session) | Never exits the module | Hashed and encrypted (via Data Protection Key) on Micro SD Card | N/A | Authentication to the module via Remote Console or HTTPS Console |
| User Password | Alphanumeric string<br>Minimum of eight (8) characters | Input electronically in encrypted form via Remote Console or HTTPS Console (over TLS session) | Never exits the module | Hashed and encrypted (via Data Protection Key) on Micro SD Card | N/A | Authentication to the module via Remote Console or HTTPS Console |
| RSA Public Key | 1024, 2048 or 3072-bit RSA public key | Imported in encrypted form<br>(1024, 2048, 3072)<br>or<br>Generated internally via FIPS-Approved DRBG<br>(2048, 3072) | Exported in encrypted form | Encrypted (via Data Protection Key) in User/CO partitions on Micro SD Card | N/A | Verifying digital signatures and encrypting symmetric key envelopes<br>1024 Verification only |

| CSP | CSP Type | Generation / Input | Output | Storage | Zeroization | Use |
|---|---|---|---|---|---|---|
| RSA Private Key | 2048 or 3072-bit RSA private key | Imported in encrypted form<br>or<br>Generated internally via FIPS-Approved DRBG | Exported in encrypted form (if export attribute of “exportable” is assigned to key during key generation) | Encrypted (via Data Protection Key) in User/CO partitions on Micro SD Card | N/A | Generating digital signatures and decrypting symmetric key envelopes |
| ECDSA Private Key | ECDSA private key:<br>P-224,<br>P-256,<br>P-384<br>BrainpoolP224r1,<br>BrainpoolP256r1,<br>BrainpoolP320r1,<br>BrainpoolP384r1,<br>BrainpoolP512r1 | Generated internally using FIPS-Approved DRBG<br>or<br>Imported in encrypted form | Exported in encrypted form (if export attribute of “exportable” is assigned to key during key generation) | Encrypted (via Data Protection Key) in User/CO partitions on Micro SD Card | N/A | Digital signature generation |
| ECDSA Public Key | ECDSA public key:<br>P-192,<br>P-224<br>P-256,<br>P-384<br>BrainpoolP224r1,<br>BrainpoolP256r1,<br>BrainpoolP320r1,<br>BrainpoolP384r1,<br>BrainpoolP512r1 | Generated internally using FIPS-Approved DRBG<br>or<br>Imported in encrypted form | Exported in encrypted form | Encrypted (via Data Protection Key) in User/CO partitions on Micro SD Card | N/A | Digital signature verification |
| Triple-DES Decryption Key | 112 or 168-bit Triple-DES key | Derived internally using PBKDF2 Import/Export Password per NIST SP 800-132 | Never output from module | Plaintext in volatile RAM | Power cycle, shutdown, reboot, HSM database reset | Decryption of PKCS#8/12 imports |
| AES Key | 128, 192, and 256-bit AES key<br>(ECB, CBC, CTR, GCM) | Imported in encrypted form<br>or<br>Generated internally via FIPS-Approved DRBG | Exported in encrypted form (if export attribute of “exportable” is assigned to key during key generation) | Encrypted (via Data Protection Key) in User/CO partitions on Micro SD Card | N/A | Encryption/decryption |

| CSP | CSP Type | Generation / Input | Output | Storage | Zeroization | Use |
|---|---|---|---|---|---|---|
| PBKDF2 Import/Export Password | 16-character value | Input electronically in encrypted form via Remote Console or HTTPS Console (over TLS session) | Never output from module | Plaintext in volatile RAM | Power cycle, shutdown, reboot, HSM database reset | Used to derive AES KEK for PKCS#8/#12 import/export<br>Used to derive Triple-DES Decryption Key for PKCS#8/#12 import (decrypt only) |
| HMAC Key | Variable-bit HMAC key | Internally generated via FIPS-Approved DRBG | Exported in encrypted form (if export attribute of "exportable" is assigned to key during key generation) | Encrypted (via Data Protection Key) in User/CO partitions on Micro SD Card | N/A | Message authentication with SHS/SHA3 |
| AES KEK | 256-bit AES-CBC key | Derived internally using PBKDF2 Import/Export Password per NIST SP 800-132 | Never output from module | Plaintext in volatile RAM | Power cycle, shutdown, reboot, HSM database reset | Encrypting/decrypting RSA private keys in PKCS#8/12 envelopes during import/export |
| PBKDF2 Backup Password | Minimum eight-character value | Input electronically in encrypted form via Remote Console or HTTPS Console (over TLS session) | Never output from module | Plaintext in volatile RAM | Power cycle, shutdown, reboot, HSM database reset | Deriving Backup Key |
| Backup Key | 128-bit AES-CBC Key | Derived internally using PBKDF2 Backup Password per NIST SP 800-132 | Never output from module | Plaintext in volatile RAM | Power cycle, shutdown, reboot, HSM database reset | Encrypting backups |

******Keys derived from the PBKDF2 function shall only be used for storage applications.*

The AES GCM IV is used for generating the Data Protection Key, operator AES GCM keys, AES GCM KEKs for key wrap/unwrap, and AES GCM keys used in the TLS protocol. The AES GCM IV is internally generated via RBG-based construction in compliance with Section 8.2.2 of *NIST SP 800-38D* using the Approved DRBG within the module's physical boundary and is 96 bits in length.

## 2.8 EMI / EMC

The module was tested and found to be conformant to the EMI/EMC requirements specified by 47 Code of Federal Regulations, Part 15, Subpart B, Unintentional Radiators, Digital Devices, Class A (i.e., for business use).

## 2.9 Self-Tests

The module performs power-up self-tests, conditional self-tests, and critical function tests. These tests are described in the sections that follow.

### 2.9.1 Power-Up Self-Tests

The HSM performs the following self-tests at power-up to verify the integrity of the firmware images and the correct operation of the FIPS-Approved algorithm implementations:

- Firmware Integrity Test on DINAMO HSM firmware components and crypto library using HMAC SHA-256
- Cryptographic algorithm tests:
    - AES ECB encrypt KAT[44]
    - AES ECB decrypt KAT
    - AES CTR encrypt KAT
    - AES CTR decrypt KAT
    - AES GCM encrypt KAT
    - AES GCM decrypt KAT
    - AES CBC encrypt KAT
    - AES CBC decrypt KAT
    - AES CMAC KAT
    - Triple-DES 3-key encrypt KAT (CBC, ECB)
    - Triple-DES 3-key decrypt KAT (CBC, ECB)
    - Triple-DES 2-key decrypt KAT (CBC, ECB)
    - Triple-DES CMAC KAT (2-key and 3-key)
    - HMAC SHA2-224, HMAC SHA2-256, HMAC SHA2-384, and HMAC SHA2-512 KATs
    - HMAC SHA3-224, HMAC SHA3-256, HMAC SHA3-384, and HMAC SHA3-512 KATs
    - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512 KATs
    - SHA3-224, SHA3-256, SHA3-384, SHA3-512 KAT
    - CTR DRBG KAT
    - Critical Functions Tests for DRBG (Instantiate, Generate, Reseed, Uninstantiate)
    - RSA signature generation KAT
    - RSA signature verification KAT
    - ECDSA signature generation KAT
    - ECDSA signature verification KAT
    - Primitive “Z” Computation KAT
    - KBKDF KAT

[44] KAT – Known Answer Test

### 2.9.2 Conditional Self-Tests

The HSM performs the following conditional self-tests:

- CRNGT for the NDRNG entropy source
- RSA Pairwise Consistency Test
- ECDSA Pairwise Consistency Test

### 2.9.3 Self-Test Failure Handling

If the module fails a power-up self-test, the module enters a "Critical" error state and reports this error to a Crypto Officer via the DinamoCon and logs the error. Once in a Critical error state, the module remains there until the Crypto Officer acknowledges the error (via the DinamoCon), at which point the module shuts down.

If an error occurs during the power-up self-tests, the module is in a state in which it cannot be activated for operation, so authentication is not possible and network ports cannot be enabled for Remote Console or HTTPS Console access. Consequently, all access to the cryptographic functionality and CSPs is disabled. All data outputs via data output interfaces are inhibited and the management interfaces will not respond to any commands, other than the self-test error acknowledgement (shutdown).

There is no action the operator can take to repair a critical error state and the module should be returned to the vendor for servicing. The only option for the operator is to acknowledge the error (which shuts down the module).

If an error occurs during conditional self-tests, the module transitions to a soft error state. The module logs the error, clears the error state, and then resumes normal operation.

## 2.10 Mitigation of Other Attacks

Fault-based attacks on a system such as high temperatures or voltage manipulation may be used to induce errors or corrupt messages. Offline analysis of these errors can be used to reverse engineer the cryptographic module, potentially extracting the private key from the cryptographic routines it executes.

The module protects against fault-based attacks that are aimed at creating erroneous RSA and ECDSA digital signatures, by affecting the result of the modular exponentiation algorithm. The protection mechanism involves validating the correctness of every RSA and ECDSA digital signature created by the module using the associated public key. In case of errors, the operation is marked as invalid, and the module returns only TAC ERR OPERATION FAILED. This protection mechanism is always active.

# 3. Secure Operation

The sections below describe how to place and keep the module in its validated configuration. **Any operation of the module without following the guidance provided below will result in non-compliant use and is outside the scope of this Security Policy.**

## 3.1 Installation and Configuration

The CO shall be responsible for receiving, installing, and configuring the HSM. To operate the module in its validated configuration, the CO shall configure the module via the DinamoCon as directed by this Security Policy. The following sections provide the CO with important instructions and guidance for the secure installation and configuration of the HSM.

### 3.1.1 Package Contents Inspection

Upon receiving the HSM hardware, the CO shall check that the system is not damaged and that all required parts and instructions are included. The CO shall refer to the *Package content* section of the administration guide here for a list of package contents: https://www.dinamonetworks.com/manualpocket/.

### 3.1.2 Physical Inspection

For the module to be considered running in its validated configuration, the factory-installed tamper-evident seals must be in place as specified in section 2.5. Upon receipt, the CO shall inspect the module to ensure the tamper-evident seals have been properly installed.

The CO is also required to periodically inspect the module for evidence of tampering at intervals specified per end-user policy. The CO must visually inspect the tamper-evident seals for tears, rips, dissolved adhesive, and other signs of tampering. If evidence of tampering is found during periodic inspection, the Crypto Officer must zeroize the keys and contact DINAMO Customer Support.

### 3.1.3 Configuration

The DinamoCon application can be used to configure the module. Customers may contact DINAMO to receive the DinamoCon application.

To place the module in its validated configuration, the CO must perform the following steps:

1. Install Pocket on the same network as administrator workstation.
2. Load and start the DinamoCon application on the administrator workstation.
3. Click **Search** to automatically find HSM.
4. Click **OK** on *Select HSM* window.
5. Click **Pocket**.
6. Click **OK** on window with warning message that says “The HSM is on Non Restricted Mode…”

7. Create a new password and enter it twice.
8. DinamoCon window opens in Non Restricted Mode.
9. Click on **Pocket** on top left of window.
10. Select Stop Service. Click **OK** on warning message.
11. Change mode by selecting “Restricted Mode 2”, the equivalent of FIPS-Approved mode, and click **Change**.
12. Click **OK** on *Changing Mode* window.
13. Authenticate with password created in Step 7.
14. Click **OK** on the window explaining that HSM needs to be initialized.
15. A reboot is triggered and the module performs the required FIPS self-tests.
16. Upon completion of the reboot, the *Custodian Procedure window* appears.
17. Click **Next** on the *Custodian Procedure* window.
18. Click **Next** again.
19. Select Copy Shadow to clipboard (Master Key Component Key 1) and save, then click **Next**. (Note: be sure to store shadow in safe place).
20. Paste shadow twice and click **Next**.
21. Click Next.
22. Repeat steps 19 and 20 to copy and paste second shadow (Master Key Component Key 2).
23. Click **Finish**.
24. Click **Start Service** to allow for access to the Remote Console and HTTPS Console.

When configured by following the setup procedure above, the module only operates in a FIPS-compliant manner. Thus, the current status of the module when operational is always in the FIPS-Approved mode.

The appliance’s operational status is indicated on the top of the main menu of the DinamoCon and the **Info** option on the main menu of the Remote Console and HTTPS Console.

## 3.2 Crypto Officer Guidance

The Crypto Officer is responsible for initialization and security-relevant configuration and management of the module. The module’s “Operator” profile has all possible permissions and is equivalent to the official Security Officer (Crypto Officer). The module’s “User” profile has the permissions for the User role described in Section 2.4.1 and Table 7. The Operator profile, being a superset of the User profile, also has its own partition of cryptographic keys.

### 3.2.1 Management

Once installed and configured, the CO is responsible for maintaining and monitoring the status of the module to ensure that it is running in its validated configuration. Please refer to Section 3.1.3 for guidance that the CO must follow for the module to be considered running in its validated configuration. COs should ensure that physical security of the module is maintained via periodic inspection of tamper-evident seals.

### 3.2.2 On-Demand Self-Tests

The power-up self-tests are automatically performed at power-up. The CO may initiate the power-up self-tests through the following methods:

- issuing the *reboot* command at the **Main** menu of the DinamoCon
- power-cycling the module (issuing the *shutdown* command at the **Main** menu of the DinamoCon and powering up the module)
- issuing the *Reset HSM database* command from the **Configuration** menu of the DinamoCon

### 3.2.3 PBKDF2 Passwords

The CO can save an encrypted database backup file. The generation of the key used for the encryption of this file is performed by an SP 800-132 PBKDF2. When the CO is prompted to enter a new password, the CO shall enter a password no less than 8 characters in length. The password shall consist of upper-case and lower-case letters and numbers. The probability of guessing the password will be equal to 1:$62^{8}$, or 1:$2.18\text{x}10^{11}$. The key derived by the PBKDF2 is used solely for storage purposes. Likewise, a PBKDF2 may be used by operators to derive the AES Key and Triple-DES Decryption Key used during import/export of RSA keys (PKCS#8/12 digital envelopes). Operators must enter a password no less than 16 characters in length. The password shall consist of upper-case and lower-case letters and numbers. The probability of guessing the password will be equal to 1:$62^{16}$, or 1:$4.77\text{x}10^{28}$.

### 3.2.4 Zeroization

Please refer to Table 9 for the key zeroization techniques and the applicable keys.

## 3.3 User Guidance

The User does not have the ability to configure sensitive information on the module, except for their password. The User must be diligent to pick strong passwords, and must not reveal their password to anyone. Additionally, the User should be careful to protect any secret or private keys in their possession.

## 3.4 Additional Guidance and Usage Policies

This section notes additional policies below that must be followed by module operators:

- If the module fails a power-up self-test, the module is considered to be compromised or malfunctioned and should be sent back to DINAMO for repair or replacement.

- In the event that the module's power is lost and then restored, a new key for use with the AES GCM encryption shall be established.

- The module does not allow for the loading of new firmware.

## 3.5 Non-Approved Mode of Operation

When initialized and configured according to the guidance in this Security Policy, the module does not support a non-Approved mode of operation.

# 4. Acronyms

Table 10 provides definitions for the acronyms used in this document.

**Table 10 – Acronyms**

| Acronym | Definition |
|---|---|
| AES | Advanced Encryption Standard |
| ANSI | American National Standards institute |
| API | Application Programming Interface |
| CA | Certificate Authority |
| CAVP | Cryptographic Algorithm Validation Program |
| CBC | Cipher Block Chaining |
| CCCS | Canadian Centre for Cyber Security |
| CKG | Cryptographic Key Generation |
| CLI | Command Line Interface |
| CMAC | Cipher-Based Message Authentication Code |
| CMVP | Cryptographic Module Validation Program |
| CO | Crypto Officer |
| CPU | Central Processing Unit |
| CRNGT | Continuous Random Number Generator Test |
| CSP | Critical Security Parameter |
| CSR | Certificate Signing Request |
| CTR | Counter |
| CVL | Component Validation List |
| DES | Data Encryption Standard |
| DH | Diffie-Hellman |
| DRBG | Deterministic Random Bit Generator |
| ECB | Electronic Codebook |
| ECC | Elliptic Curve Cryptography |
| ECC CDH | Elliptic Curve Cryptography Cofactor Diffie-Hellman |
| ECDH | Elliptic Curve Diffie-Hellman |
| ECDSA | Elliptic Curve Digital Signature Algorithm |
| EMI/EMC | Electromagnetic Interference/Electromagnetic Compatibility |
| FIPS | Federal Information Processing Standard |
| GCM | Galois/Counter Mode |
| GHz | Gigahertz |

| Acronym | Definition |
|---|---|
| Acronym | Definition |
| HMAC | (keyed-) Hash Message Authentication Code |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hypertext Transfer Protocol Secure |
| IV | Initialization Vector |
| JCA | Java Cryptography Architecture |
| JCE | Java Cryptography Extension |
| KAS | Key Agreement Scheme |
| KAS-SSC | Key Agreement Scheme – Shared Secret Computation |
| KAT | Known Answer Test |
| KBKDF | Key Based Key Derivation Function |
| KDF | Key Derivation Function |
| KEK | Key Encryption Key |
| KPG | Key Pair Generation |
| LAN | Local Area Network |
| Mbps | Megabits per second |
| MS | Microsoft |
| N/A | Not Applicable |
| NDRNG | Non-Deterministic Random Number Generator |
| NIST | National Institute of Standards and Technology |
| OS | Operating System |
| PBKDF2 | Password-based Key Derivation Function 2 |
| PCT | Pairwise Consistency Test |
| PKCS | Public Key Cryptography Standard |
| PKG | Public Key (Q) Generation |
| PKV | Public Key (Q) Validation |
| PRF | Pseudo-Random Function |
| PSK | Pre-shared Key |
| PSS | Probabilistic Signature Scheme |
| PUB | Publication |
| RAM | Random Access Memory |
| RSA | Rivest Shamir Adleman |
| SD | Secure Digital |
| SHA | Secure Hash Algorithm |
| SHS | Secure Hash Standard |
| SP | Special Publication |

| Acronym | Definition |
|---|---|
| SSL | Secure Socket Layer |
| TLS | Transport Layer Security |
| U.S. | United States |

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

13921 Park Center Road, Suite 460
Herndon, VA 20171
United States of America

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