FIPS 140-2
Non-Proprietary Security Policy
for
Absolute Encryption Engine

Level 1 Validation

(Software Version: 1.2.0.46)

Document Version 1.1

Date: December 15, 2011

AbsoluteSoftware

www.absolute.com

This document is prepared for Absolute Software Corp. by Uptown Telecom Consulting Services, Ltd.

Revision History

Version	Modification Date	Modified By	Description of Changes
0.0	February 24, 2010	Yvonne Chow	First draft based on a common DLL for both client and server and Product Development (PD) inputs.
0.1	February 25, 2010	Yvonne Chow	Updated draft based on review comments from PD.
0.2	February 26, 2010	Yvonne Chow	Updated based on review comments from PD and CTO.
0.3	September 29, 2010	Yvonne Chow	Updated as per July 8 and August 23, 2010 comments from EWA-Canada and revised implementation
0.4	December 3, 2010	Yvonne Chow	Updated as per November 26, 2010 comments from EWA-Canada
1.0	July 13, 2011	Yvonne Chow	Updated as per February 9, 2011 and June 10, 2011 and July 12, 2011 comments from EWA-Canada
1.1	December 15, 2011	Dale Quantz	Updated as per comments from EWA-Canada November 2, 2011

Table of Contents

Table of Figures

List of Tables

1 Introduction

1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the Absolute Encryption Engine from Absolute Software Corporation. This Security Policy describes how the Absolute Encryption Engine meets the security requirements of Federal Information Processing Standards (FIPS) Publication 140-2, which details the U.S. Government requirements for cryptographic modules. This policy was prepared as part of the Level 1 FIPS 140-2 validation of the module.

The cryptographic module, Absolute Encryption Engine, is referred to as the DLL, the cryptographic module or the module throughout this document.

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 Absolute website (http://www.absolute.com/) contains information on the full line of products from Absolute.
- The National Institute of Standards and Technology (NIST) Cryptographic Module Validation Program (CMVP) website (http://csrc.nist.gov/groups/STM/cmvp/index.html) contains information about the FIPS 140-2 standard and validation program. It also lists contact information for answers to technical or sales-related questions for the module.

1.3 Document Organization

This Security Policy document is one of the required documents in the FIPS 140-2 Submission Package. In addition to this document, the Submission Package also contains:

- Finite State Machine document
- Application Programming Interface (API) document
- Other supporting documentation as additional references

This Security Policy and the other validation submission documentation were produced by Uptown Telecom Consulting Services, Ltd. under contract to Absolute Software Corporation With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 Validation Documentation is proprietary to Absolute and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact Absolute.

2 Absolute Absolute Encryption Engine

2.1 Product Overview

Absolute Software Corporation provides security products for the central management of all IT assets. The Absolute Encryption Engine is a dynamic-linked library (DLL) defined as the encryption module on the client and server callable by applications via an Application Programming Interface (API). The module is currently used by the Absolute Computrace product.

Computrace by Absolute Software allows users to centrally manage all IT assets with a single interface. Users can locate any of their computers that have gone missing, enforce software policies, and maintain a fleet of optimally running devices. Absolute Encryption Engine monitors changes in asset information including user identification, physical location, and installation of software or hardware that may not comply with government and corporate regulations and provides advanced reporting capabilities. In the event of computer loss and theft, Absolute Encryption Engine can delete all sensitive information and generate reports to prove user compliance with government and corporate regulations. The process of locating a computer is done using either using the laptops global positioning device built into the laptop or by using IP information and determing the location on a more granular level.

Computrace is based on a client-server architecture. The Computrace Agent is a client software platform that allows you to access Absolute Software services while CTSRV is the Computrace server. A typical Computrace setup diagram is shown in Figure 1 below.

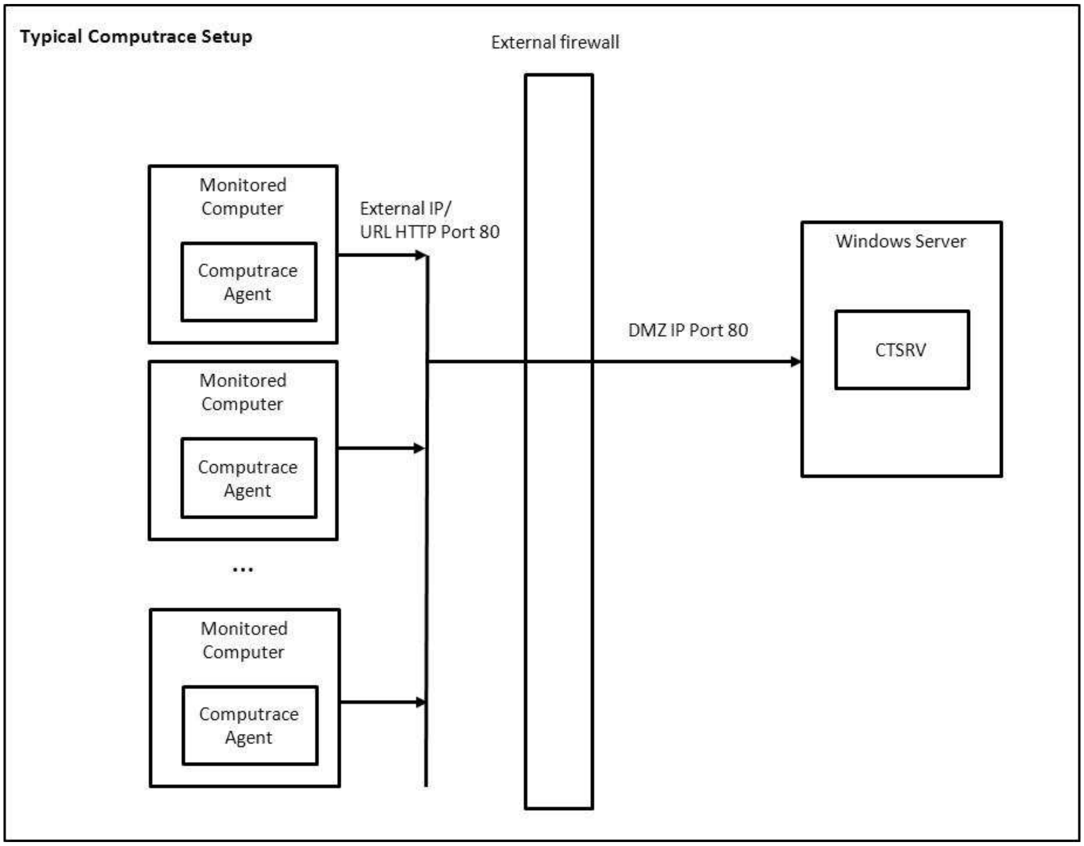

Figure 1 – Typical Computrace Setup

Other Absolute products also include the Absolute Hard Disk Encryption and Virtual Encrypted Disk. They provide better data protection for improved product security.

2.1.1 Product Platforms

The Absolute Computrace product currently supports the Microsoft Windows, Linux and Mac Operating Systems including the following:

PC

- Windows 7 (32 bit versions)
- Windows Vista (32 & 64-bit versions)
- Windows XP (32-bit only)
- Windows Server 2008

Linux

- Red Hat Enterprise Linux 6 Workstation (32 bit versions)

Mac

- Mac OS X 10.6.7

Other platforms supported and not tested are Mac OS X 10.3.x and higher.

2.2 Applicable FIPS 140-2 Sections

The cryptographic module is being submitted for validation to FIPS 140-2, Security Level 1. Table 1 – Applicable FIPS 140-2 Sections lists the sections applicable to the module.

Table 1 – Applicable FIPS 140-2 Sections

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	Electromagnetic Interference / Electromagnetic Compatibility (EMI/EMC)	1
9	Self-tests	1
10	Design Assurance	1
11	Mitigation of Other Attacks	N/A

3 Absolute Encryption Engine

3.1 Cryptographic Module Definition

This section defines the cryptographic module, the Absolute Encryption Engine which is being submitted for validation to FIPS 140-2, Security Level 1. The DLL is a multi-chip standalone cryptographic module in accordance with the FIPS 140-2 definition.

The Absolute Encryption Engine is a software cryptographic module for use on a general purpose computing platform. The module's software is within the logical cryptographic boundary as defined in Section 3.1.2.

The module is tested on the following platform pursuant to the FIPS140-2 requirements:

PC (Server)

- Windows Server 2008 (64-bit version)

PC (Agent)

- Windows 7 (32-bit version)
- Windows XP (32-bit version)
- Windows Vista (32-bit version)
- Windows Vista (64-bit version)

Linux (Agent)

- Red Hat Enterprise Linux 6 (32-bit version)

Mac (Agent)

- Mac OS X v10.6.7 (32-bit version)

The module consists of a set of processes and functions running on the client and server in a client-server architecture that consists of the following generic components:

1. A commercially available general-purpose hardware computing platform. A generic high-level block diagram for such a platform is provided in Figure 2.
2. A commercially available Operating System (OS) that runs on the above platform.
3. A commercially available hard disk that operates on the above platform.
4. The Absolute cryptographic module that runs on the above platform and operating system. This module is custom designed and written by Absolute in the 'C' and 'C++' programming languages and is identical, at the source code level, for all identified hardware platforms and operating systems. An Application Programming Interface (API) is defined as the interface to the cryptographic module.

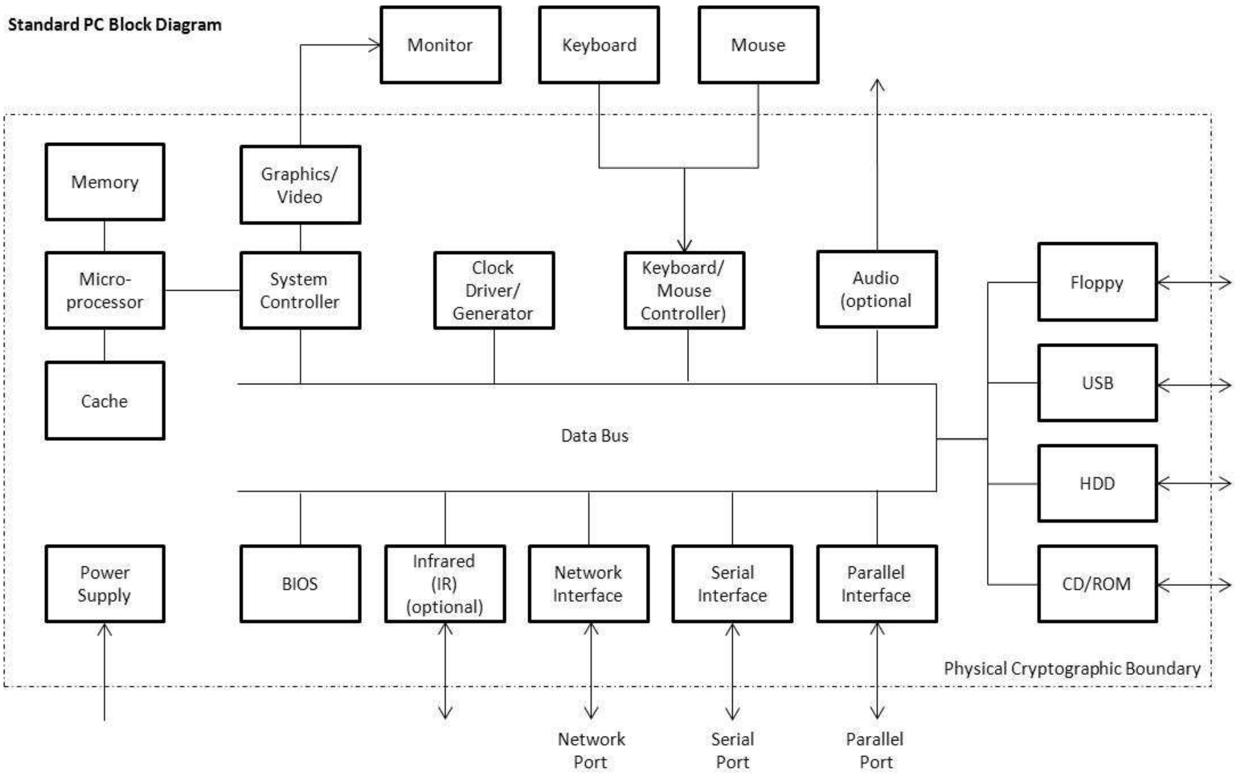

Figure 2 – Standard GPC Block Diagram

3.1.1 Physical Cryptographic Boundary

The module is considered as a multichip standalone module pursuant to the FIPS 140-2 definition. The physical cryptographic boundary contains the general purpose computing hardware of the system executing the application. The casing of the general purpose computer is a hard opaque metal and plastic enclosure. The system hardware includes the central processing unit(s), cache and main memory (RAM), system bus and peripherals including the disk drives and other permanent mass storage devices, network interface cards, keyboard, console and any terminal devices. Figure 2 illustrates the various components, connections, and information flows (the dashed line surrounding the various components makes up the module's physical cryptographic boundary).

3.1.2 Logical Cryptographic Boundary

The logical cryptographic boundary of the module is defined by the set of processes and functions residing on the client and server. The list of authorized services described in Table 3 defines the logical cryptographic boundary. The list of host operating systems being tested is in Section 3.1.

The library on each platform is provided below:

In Windows 7(32bit & 64bit), XP(32bit) and Vista(32bit & 64 bit):
Library: eprvst.dll

In Windows 2008 (64-bit):
Library: eprvst64.dll

In Mac:
Library: eprvst.framework

In Linux:
Library: libwceprv.so

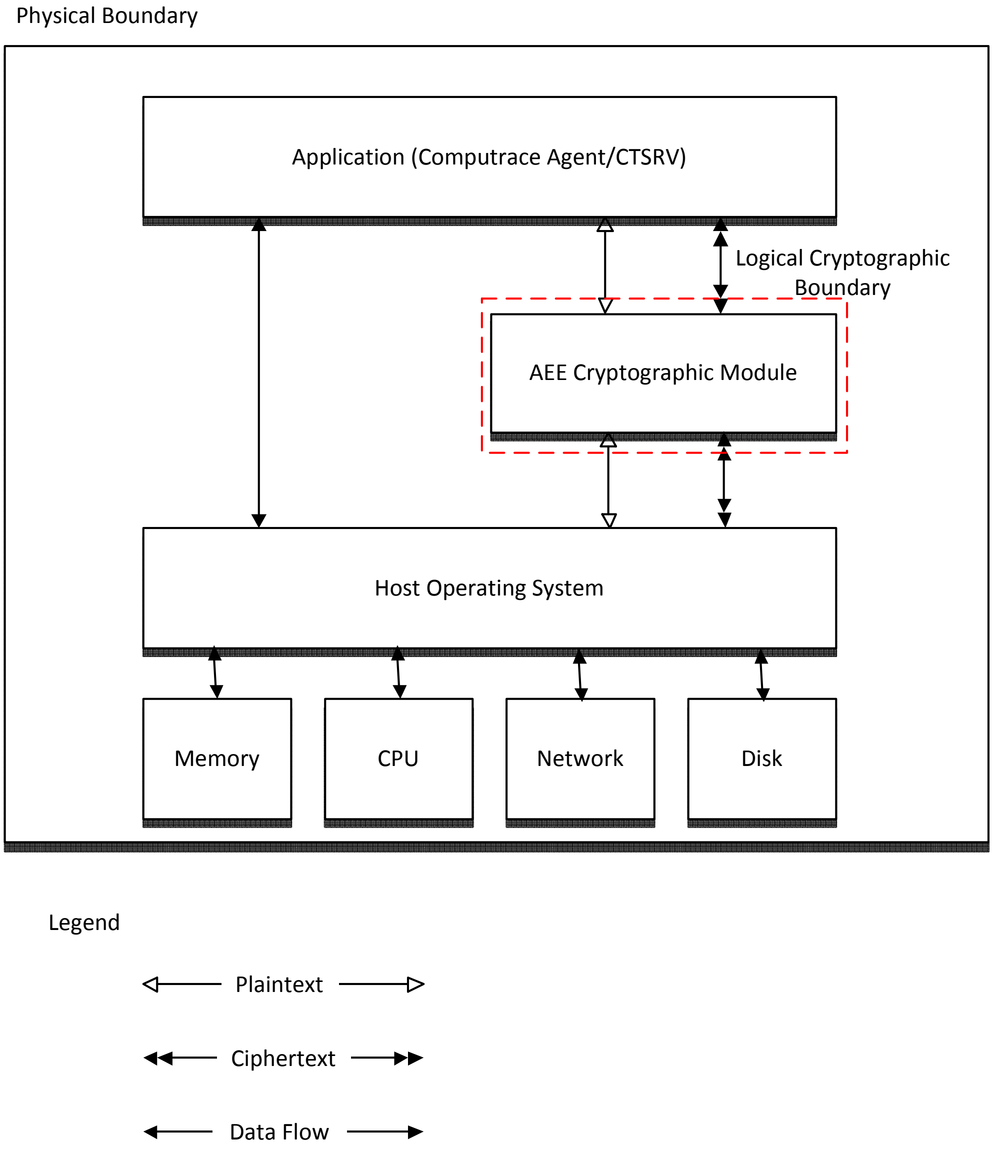

Figure 3 – Absolute Encryption Engine Logical Cryptographic Boundary

The red dashed box enclosing the "AEE Cryptographic Module" in Figure 3 above represents the logical cryptographic boundary.

3.2 Module Ports and Interfaces

The cryptographic module provides a logical interface via an Application Programming Interface (API). The API provides functions that may be called directly by the referencing application. The API provided by the module is mapped onto the FIPS 140-2 logical interfaces: data input, data output, control input, and status output.

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

Table 2 – FIPS 140-2 Logical Interfaces

FIPS 140-2 Logical Interface	Module Ports/Interfaces	Physical Ports/Interfaces
Data Input Interface	Module function calls	Keyboard, mouse and serial/USB/parallel/network ports
Data Output Interface	Return values of module function calls	Monitor and serial/USB/parallel/network ports
Control Input Interface	Module control function calls	Keyboard, CD/DVD-ROM, mouse, and serial/USB/parallel/network port
Status Output Interface	Return values from module status function calls	Monitor and serial/USB/parallel/network ports
Power	Not applicable	Power Interface

- Data input: function calls that accept data to be used or processed by the module
- Data output: output parameters from all functions that return data as return values
- Control input: function calls used to control the operation of the module
- Status output: module status provided by return values when invoked by the appropriate function calls

3.3 Roles, Services, and Authentication

3.3.1 Access Control Policy

The Absolute Encryption Engine provides no authentication of operators. However, the operating system platforms on which the module operates do provide authentication, but this is outside of the cryptographic boundary hence outside the scope of the current FIPS validation.

There are two roles in the cryptographic module that operators may assume: the Crypto Officer (CO) role and the User role. All services associated with the Computrace Server are assigned the Crypto Officer role and all Agent services are assigned the User role. Multiple concurrent operators are not allowed. Only a single user may access it at any given point in time.

The CO role has access to all the services on the server.

The User role can perform general security services on the agent. This role can also perform initialization, initiate self-tests on demand and check the status of the module.

A Maintenance role is not supported.

3.3.2 Roles and Services

The module operates only in the FIPS-Approved mode of operation. Both the CO and User will be able to utilize the functionality of the module.

Table 3 and Table 4 list the services available to the roles. Since the module is a dynamic-linked library, inputs to the services are provided in the form of function calls via the module's Application Programming Interface (API) and outputs are in return values and/or operations performed by the module.

The following abbreviations are used in Table 3 and Table 4 below:

R – Read, CSP/Key is read by the service
W – Write, CSP/Key is modified by the service
CID – Context Identifier

Table 3 – Mapping of the CO Role's Services to Inputs, Outputs, CSPs and Type of Access on the Server

Service	Description	Input	Output	CSP and Type of Access
Initialize	Load the module	API Call	Status	SW Key; R SW IV; R PRNG Seed; W PRNG Key; W
Perform self-tests	Execute power-up self-tests on demand	API Call	Status	SW Key; R SW IV; R PRNG Seed; W PRNG Key; W
AES context initialization	Create a new AES context	API Call	Status, AES CID	PRNG Seed; W PRNG Key; R
AES context de-initialization	Zeroize the AES transport key stored in the context and de-initialize the context	API Call, AES CID	Status	AES Trans; W
RSA context initialization	Create a new RSA context	API Call	Status, RSA CID	PRNG Seed; W PRNG Key; R
RSA private context de-initialization	Zeroize the RSA private key stored in the context and de-initialize the context	API Call, RSA CID	Status	RSA Trans Private; W
GCM context initialization	Create a new GCM context	API Call	Status, GCM CID	PRNG Seed; W PRNG Key; R
GCM context de-initialization	Zeroize the AES GCM key and IV stored in the context and de-initialize the context	API Call, GCM CID	Status	GCM Key; W GCM IV; W
Import AES key	Import plaintext AES key	API Call, AES CID, plaintext key	Status	AES Trans; W

Service	Description	Input	Output	CSP and Type of Access
Import RSA private key	Import encrypted RSA private key	API Call, AES CID, RSA CID, encrypted key	Status	RSA Trans Private; W
Import AES and RSA keys	Import plaintext AES key and encrypted RSA private key	API Call, RSA CID, plaintext key, encrypted key	Status, AES CID	PRNG Seed; W PRNG Key; R AES Trans; W RSA Trans Private; W
Import GCM key and IV	Import encrypted AES GCM key and IV	API Call, RSA CID, GCM CID, encrypted key, encrypted IV	Status	RSA Trans Private; R GCM Key; W GCM IV; W
Authenticated Encryption	Encrypt plaintext using AES GCM and calculate a GCM hash	API Call, GCM CID, plaintext, AAD	Status, ciphertext, hash	GCM Key; R GCM IV; W
Authenticated Decryption	Verify the GCM hash and decrypt ciphertext using AES GCM	API Call, GCM CID, ciphertext, hash, AAD	Status, success or failure indicator, plaintext	GCM Key; R GCM IV; W
Get Status	Get current status of the module	API Call	Status	None

Table 4 – Mapping of the User Role's Services to Inputs, Outputs, CSPs and Type of Access on the Agent

Service	Description	Input	Output	CSP and Type of Access
Initialize	Load the module	API Call	Status	SW Key; R SW IV; R PRNG Seed; W PRNG Key; W
Perform self-tests	Execute power-up self-tests on demand	API Call	Status	SW Key; R SW IV; R PRNG Seed; W PRNG Key; W
RSA context initialization	Create a new RSA context	API Call	Status, RSA CID	PRNG Seed; W PRNG Key; R
RSA public context de-initialization	Zeroize the RSA public key stored in the context and de-initialize the context	API Call, RSA CID	Status	RSA Trans Public; W

Service	Description	Input	Output	CSP and Type of Access
GCM context initialization	Create a new GCM context	API Call	Status, GCM CID	PRNG Seed; W PRNG Key; R
GCM context de-initialization	Zeroize the AES GCM key and IV stored in the context and de-initialize the context	API Call, GCM CID	Status	GCM Key; W GCM IV; W
Import RSA public key	Import plaintext RSA public key	API Call, RSA CID, plaintext key	Status	RSA Trans Public; W
Export GCM key and IV	Export encrypted AES GCM key and IV	API Call, RSA CID, GCM CID, encrypted key, encrypted IV	Status	RSA Trans Private; R GCM Key; W GCM IV; W
Generate GCM key and IV	Generate AES GCM key and IV using PRNG, and export generated key and IV using RSA	API Call, RSA CID, GCM CID	Status	PRNG Seed; W PRNG Key; R GCM Key; W GCM IV; W RSA Trans Public; R
Authenticated Encryption	Encrypt plaintext using AES GCM and calculate a GCM hash	API Call, GCM CID, plaintext, AAD	Status, ciphertext, hash	GCM Key; R GCM IV; W
Authenticated Decryption	Verify the GCM hash and decrypt ciphertext using AES GCM	API Call, GCM CID, ciphertext, hash, AAD	Status, success or failure indicator, plaintext	GCM Key; R GCM IV; W
Get Status	Get current status of the module	API Call	Status	None

3.3.3 Operator Authentication

As a Level 1 cryptographic module, the module does not support identification or authentication mechanisms that would distinguish between the two supported roles. These roles are assumed implicitly.

Table 5 – Roles and Required Identification and Authentication

Role	Type of Authentication	Authentication Data
Crypto Officer	N/A	N/A
User	N/A	N/A

Table 6 – Strengths of Authentication Mechanisms

Authentication Mechanism	Strength of Mechanism
N/A	N/A
N/A	N/A

3.4 Finite State Model

Refer to the Finite State Model document.

3.5 Physical Security

The cryptographic module is a software module and does not include physical security mechanisms. Therefore, the Physical Security requirements do not apply.

3.6 Operational Environment

The cryptographic module operates on a general purpose computing platform. Please refer to 3.1 for a list of supported product platforms.

All of the supported platforms are configured for single operator mode as per NIST guidance. Since the module is a dynamic-linked library, the operator is the application calling the module. No concurrent users are allowed. As such, all keys, intermediate values, and other CSPs remain only in the process space of the operator using the module. The operating systems use their native memory management mechanisms to ensure that outside processes cannot access the process space and memory used by the module.

3.7 Cryptographic Key Management

The module implements the following FIPS-approved algorithms:

- AES ECB (128-bit), AES GCM (128-bit) - (Certificate #1610)
- Pseudo-Random Number Generator (ANSI X9.31) - (Certificate #864)

As of January 2011, ANSI X9.31 RNG is deprecated. Please refer to NIST Special Publication 800-131A for more information.

The module implements the following allowed non-Approved security function:

RSA (key wrapping; key establishment methodology provides 128 bits of encryption strength.) RSA is implemented and used only for key transport.

The module supports the following keys and critical security parameters:

Table 7 – List of Cryptographic Keys and CSPs

The following abbreviations are used in Table 7 below:

ED – Electronic Distribution
EE – Electronic Entry

Key/ CSP	Type	Size	Generation / Input	Output	FIPS-Approved Establishment Mechanism	Storage	Zeroization	Owner Usage
SW Key	AES GCM Key	128 bits	Generated outside the module/embedded when module is built	Never exits the module	EE	In binary	Module is uninstalled	Server, Agent Software integrity test
SW IV	AES GCM IV	128 bits	Generated outside the module/embedded when module is built	Never exits the module	EE	In binary	Module is uninstalled	Server, Agent Software integrity test
PRNG Key	AES Key	128 bits	Generated internally	Never exits the module	EE	In RAM* as plaintext while module is running	Module is unloaded	Server, Agent Random number generation
PRNG Seed	Seed value	128 bits	Generated internally	Never exits the module	EE	In RAM* as plaintext while module is running	Module is unloaded	Server, Agent Random number generation
RSA Trans Private	RSA private key	3072 bits	Generated outside the module and imported into the module via API call	Never exits the module	EE	In RAM* as plaintext while RSA context is active	RSA context is de-initialized	Server Import of AES GCM Key and AES GCM IV generated by a client
RSA Trans Public	RSA public key	3072 bits	Generated outside the module and imported into the module via API call	Never exits the module	EE	In RAM* as plaintext while RSA context is active	RSA context is de-initialized	Agent Export of AES GCM Key and AES GCM IV

Key/ CSP	Type	Size	Generation / Input	Output	FIPS-Approved Establishment Mechanism	Storage	Zeroization	Owner Usage
AES Trans Key	AES Key	128 bits	Generated outside the module and imported into the module via API call	Never exits the module	EE	In RAM* as plaintext while AES context is active	AES context is de-initialized	Server Import of encrypted RSA private key
GCM Key	AES GCM Key	128 bits	Generated internally by a client or generated outside the module and imported into a server via API call	Encrypted	ED/EE	In RAM* as plaintext while AES GCM context is active	AES GCM context is de-initialized	Server, Agent Authenticated encryption and decryption
GCM IV	AES GCM IV	128 bits	Generated internally by a client or generated outside the module and imported into a server via API call	Encrypted	ED/EE	In RAM* as plaintext while AES GCM context is active	AES GCM context is de-initialized	Server, Agent Authenticated encryption and decryption

* Inside the module's cryptographic boundary.

3.8 Electromagnetic Interference/ Electromagnetic Compatibility (EMI/EMC)

The module conforms to the EMI/EMC requirements specified by 47 Code of Federal Regulations, Part 15, Subpart B, Unintentional Radiators, Digital Devices, Class A. The EMI/EMC requirements are met by the standard PC platform.

3.9 Self-Tests

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

- Software Integrity Test
  - A 128-bit hash is computed over the entire module using AES GCM and compared to a stored hard value to verify the integrity of the module.
- Cryptographic Algorithm Test

- A Known Answer Test (KAT) is performed for: AES encryption, AES decryption, AES GCM authenticated encryption using GHASH, AES GCM decryption using GHASH, and PRNG.

Since the module is a dynamic-linked library, the power-up self-tests are initiated automatically when the module is loaded. If any of the self-tests fails, the module on the Windows platform will fail to load while the module on the Mac and Linux platforms will load but all cryptographic operations will fail. The module will enter the Error state. To recover from the error state, a new module must be installed. To show the status of the module, the user can perform a get status after the module has loaded successfully.

Power-up self-tests can also be run on demand by the API call, DoFIPSTests.

No other critical functions are tested so no critical functions test is performed.

The module performs the following **conditional self-test**.

- Continuous Random Number Generation Test
  - Each output block is compared with the previous output block to ensure they are not the same.

Status is reported automatically at the completion of the conditional self-test execution. If the conditional self-test fails, the module on the Windows platform will fail to load while the module on the Mac and Linux platforms will load but all cryptographic operations will fail. The module will enter the Error state. To recover from the error state, a new module must be installed. To show the status of the module, the user can perform a get status after the module has loaded successfully.

Since the module does not generate any public or private keys, the pair-wise consistency test is not performed. No software or firmware components are externally loaded into the module; therefore, the software/firmware load test is not required. Since no cryptographic keys or key components are manually entered, manual key entry test is not applicable. The module does not support a bypass capability hence no bypass test is performed.

3.10 Design Assurance

3.10.1 Configuration Management

Absolute uses the Visual Studio Team System (VSTS) Team Foundation Server (TFS), an automated configuration management system for work item tracking, source control and software version control. Code must be checked out to edit and then checked in to become part of the final source tree for the software build. This configuration management system also keeps track of the versions of files used for each build and release. Microsoft SharePoint is used as a document repository which provides document version control.

3.10.2 Delivery and Operation

The module has to be installed on one of the tested platforms, as specified in Section 3.1 of the Security Policy. The platform must be set up for single-user operation mode.

The module is loaded by the application (the Computrace Agent on the client and the CTSRV server application on the server) after the application starts up. Please refer to the Agent Install Guide for installation details.

3.10.3 Installation

Please refer to the Agent Install Guide for details. Chapters 2, 3 and 4 provide the installation procedures for the agent on the Windows, Mac and Linux platforms respectively.

The Server installation and configuration are performed internally. No installation instructions are available.

3.10.4 Configuring Single User Mode

This section describes how to configure the single user mode for the different operating system platforms supported by the module.

3.10.4.1 Microsoft Windows

To configure the single user mode for systems running a Microsoft Windows 7 (32-bit version), Windows XP (32-bit version), Windows Vista (32-bit version), Windows Vista (64-bit version) and Windows Server 2008 (64-bit version), the user must ensure that all remote guest accounts are disabled in order to ensure that only one operator can log into the OS at a time. The services that should be disabled include the following:

- Server services
- Terminal services
- Remote registry service
- Remote desktop and remote assistance service

3.10.4.2 Red Hat Linux

To configure the single user mode for systems running Red Hat Enterprise Linux 6 (32-bit version), the user must follow the following procedures:

1. Log in as the “root” user.
2. All the users except “root” and the pseudo-users should be removed from the system files /etc/passwd and /etc/shadow. Password fields in /etc/shadow for the pseudo-users should be either an asterisk (*) or double exclamation mark (!!). This prevents login as the pseudo-users.
3. The system file /etc/nsswitch.conf needs to be edited such that “files” the only option for “passwd”, “group”, and “shadow”. This disables NIS and other name services for users and groups.
4. In the /etc/xinetd.d directory, the value of “disable” needs to be set to “yes” for the following files: “rexec”, “rlogin”, “rsh”, “rsync”, “telnet”, and “wu-ftpd”.

5. The system has to be rebooted for the changes to take effect

3.10.4.3 Macintosh

To configure the single user mode for systems running Mac OS X v10.6.7 (32-bit version), the user must follow the following procedures:

1. Boot (or reboot) with the command key (apple key) and the 'S' key held down. Follow the directions on the screen. This boots into a command line mode (no GUI) 2. Single User with GUI:
2. In System Preferences, select the Accounts preference.
3. In the Accounts preference, select the "Login Options" button (below the list of users). From the Login Options, ensure that "Show fast user switching menu as:" checkbox is unchecked.
4. In System Preferences, select the Sharing preference.
5. In the Sharing preference, ensure that "Remote Login" is unchecked

3.10.5 Management

No specific management activities are required for the module.

3.10.6 Zeroization

The software integrity key and Initialization Vector will remain until the module has been uninstalled. The PRNG seed and PRNG key will be zeroized when the module is unloaded. All other keys and CSPs are zeroized as soon as the associated context is de-initialized.

3.10.7 Development

The design documents show the correspondence of the design to the Security Policy.

3.10.8 Crypto Officer and User Guidance

Anyone who has access to the module must not attempt to modify the configuration of the module as designed or reveal any of the CSPs used by the module to other parties.

3.11 Mitigation of Other Attacks

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

4 Abbreviations

Table 8 – Abbreviations

Acronym	Definition
AAD	Additional Authentication Data
AEE	Absolute Encryption Engine
AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
ANSI	American National Standards Institute
API	Application Programming Interface
BIOS	Basic Input/Output System
CD/ROM	Compact Disk Read-Only Memory
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameter
CTSRV	Computrace Server
DLL	Dynamic-Linked Library
DMZ	Demilitarized Zone
DVD/ROM	Digital Video Disc Read-Only Memory
ED	Electronic Distribution
EE	Electronic Entry (Input/Output)
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FIPS	Federal Information Processing Standard
GCM	Galois Counter Mode
GPC	General Purpose Computer
HDD	Hard Disk Drive
HTTP	Hypertext Transfer Protocol
IO	Input/Output
IP	Internet Protocol
IR	Infrared
IV	Initialization Vector
KAT	Known Answer Test
NIST	National Institute of Standards and Technology
NVLAP	National Voluntary Lab Accredited Program
OS	Operating System
PC	Personal Computer
PRNG	Pseudo Random Number Generator
RSA	Rivest, Shamir and Adleman

Acronym	Definition
TFS	Team Foundation Server
URL	Uniform Resource Locator
USB	Universal Serial Bus
VSTS	Visual Studio Team System