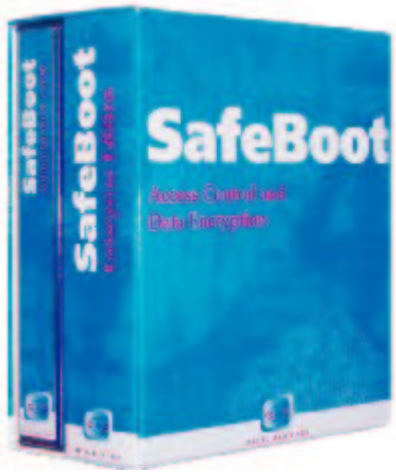
CONTROL BREAK INTERNATIONAL

Control Break International
SafeBoot Client Version 4.1



FIPS 140-1 Non-Proprietary Security Policy

Level 1 Validation

Revision 1.2, October 2002

Table of Contents

1 INTRODUCTION

1.1 Purpose

This is the non-proprietary FIPS 140-1 security policy for the Control Break International SafeBoot Client product. This Security Policy details the secure operation of the SafeBoot Client as required in Federal Information Processing Standards Publication 140-1 (FIPS 140-1) as published by the National Institute of Standards and Technology (NIST) of the United States Department of Commerce.

1.2 References

For more information on SafeBoot please visit www.controlbreak.net. For more information on NIST and the cryptographic module validation program, please visit www.nist.gov/cmvp.

1.3 Document Organization

This Security Policy document is one part of the complete FIPS 140-1 Submission Package. In addition to this document, the complete Submission Package contains:

- Vendor Evidence Document
- Finite State Machine
- Source Code Listing
- Other supporting documentation as additional references

This document outlines the functionality provided by the module and gives high-level details on the means by which the module satisfies FIPS 140-1 requirements. With the exception of this Non-Proprietary Security Policy, the FIPS 140-1 Submission Documentation may be Control Break International-proprietary or otherwise controlled and releasable only under appropriate non-disclosure agreements. For access to these documents, please contact Control Break International.

2 SafeBoot

SafeBoot is a Personal Computer (PC) security system that prevents the data stored on a PC's hard disk from being read or used by an unauthorized person. In simple terms, the SafeBoot client takes control of a user's hard disk away from the operating system. SafeBoot encrypts data written to the disk, and decrypts data read from the disk. If the hard disk drive is read directly, one would find only encrypted data, even in the Windows swap file and temporary file areas.

SafeBoot supports centralized management of SafeBoot protected machines. SafeBoot components include the SafeBoot Administrator, SafeBoot Server, SafeBoot Object Database, SafeBoot Client, SafeBoot File Encryptor and SafeBoot Connector Manager. Every time a SafeBoot protected machine boots, and optionally every time the user initiates a dial-up connection or after a set period of time, SafeBoot tries to contact its *"Object DataBase"*. This is a central store of configuration information for both machines and users, and is managed by *SafeBoot Administrators.* The *Object DataBase* could be on the users local hard disk (if the user is working completely stand-alone), or could be in some remote location and accessed over Transmission Control Protocol/Internet Protocol (TCP/IP) via a secure *SafeBoot Server* (in the case of a centrally managed enterprise).

The SafeBoot protected machine queries the Object Database for any updates to its configuration, and if needed downloads and applies them. Typical updates could be a new user assigned to the machine by an administrator, a change in password policy, or an upgrade to the SafeBoot operating system or a new file specified by the administrator. At the same time SafeBoot uploads details like the latest audit information, any user password changes, and security breaches to the *Object DataBase.* In this way, transparent synchronization of the enterprise becomes possible.

SafeBoot has the option of being configured in different ways. At installation, the SafeBoot Administrator can specify how the hard disk can be encrypted by choosing one of three encryption modes: full, partial, or none. Full encryption mode encrypts an entire partition. Partial encryption mode encrypts only a portion of a partition or hard disk. None encryption mode leaves the partition in plaintext with no encryption. (Refer to section 3 for FIPS compliant configuration.)

2.1 SafeBoot client

The SafeBoot client consists of a boot Operating System (OS) (the SafeBoot Client OS), a Basic Input Output System (BIOS) hook, Windows drivers, a system tray application and a set of Windows Dynamic Link Libraries (DLLs). These components comprise the validated module. SafeBoot installs a mini-operating system on the users hard drive, this is what the user sees when they boot the PC. SafeBoot looks and feels like Microsoft Windows, with mouse and keyboard support, moveable windows etc. The SafeBoot Client OS is completely contained and does not need to access any other files or programs on the hard disk, and is responsible for allowing the user to authenticate.

Once the user has entered the correct authentication information, the SafeBoot operating system starts a driver in memory and boots the protected machine's original operating system. From this point on the machine will look and behave as if SafeBoot was not installed.

2.2 Module Interfaces

The SafeBoot Client is classified as a multi-chip standalone module for FIPS 140-1 purposes. As such, the module includes a computer running an operating system (OS) and interfacing with the computer keyboard, mouse, screen, LAN ports, floppy drive, CD-ROM drive, speaker, disk drive, microphone inputs, serial ports, parallel ports, and power plug.

SafeBoot provides a logical interface via an Application Programming Interface (API) and a Graphical User Interface (GUI). This logical interface exposes services (described in section 2.3) that the User, the operating system and SafeBoot Client applications may utilize directly.

The logical interfaces provided by the SafeBoot Client are mapped onto the FIPS 140-1 logical interfaces: data input, data output, control input, and status output as follows:

- Data Input – Input to all driver functions
- Data Output – Output from all driver functions
- Control Input – Input from TCP/IP interface, IPC interface, GUI
- Status Output – Return codes from driver functions, Show Status GUI option

The Data Input and Data Output interfaces are the interfaces through which data is encrypted with the chosen algorithm (more information found in section 2.6) prior to being written to a disk and encrypted data is decrypted when read from a disk. The Control Input interface is the means by which the client is configured. All configuration information is applied via synchronization operations with the associated Object Database. Synchronization can be initiated by several means, including: TCP/IP connections to/from the management software, IPC (inter-process communications) functions and GUI options on the system tray application. The Status Output interface consists of text information displayed in a dialog box when the "Show Status" option is selected on the system tray application menu.

2.3 Roles and Services

The SafeBoot Client meets all FIPS 140-1 level 2 requirements for Roles and Services, implementing both a Crypto Officer role and a User role. The module performs identity-based authentication for User operators and role-based authentication for Crypto Officer operators. The following table summarizes the services available to each role.

Role	Purpose	Services
Crypto Officer	Module configuration	- Connect to module via an encrypted session to transmit control data
User	Usage of module functionality	- Utilize hard disk encryption services - Initiate synchronization with management software - View status

Table 1 Roles

The following table lists each service and the corresponding role that can utilize the service:

Service	User	Crypto Officer
Synchronization	X	X
Encryption/Decryption	X	X
Show Status Functions	X	X
Self-test Functions	X	X
User/machine recovery requests		X
Change User Password	X	
Configuration		X
File Updates		X
Manage SafeBoot (Cryptographic & Key Management Functions)		X
Software Updates		X
User/machine recovery		X
Set User Attributes (passwords, access rights, etc.)		X
Change User Attributes		X
Create User Groups		X
Modify User Groups		X
Delete User Groups		X
Create Users		X
Modify Users		X
Delete Users		X

Table 2 Services

The User role is assumed when a SafeBoot protected machine is booted and proper username and password is entered into the login prompt displayed by the boot SafeBoot Client OS. Once authenticated, user specific information and key material are loaded from the SBFS (SafeBoot File System) and the original operating system (with SafeBoot drivers installed) is launched. The necessary key material and machine state information is loaded into the drivers and the transparent encryption/decryption of disk-based information begins. A system tray application, which may be configured to start automatically, may be used to view the status of the module or to initiate a synchronization operation.

The Crypto Officer role may be assumed by establishing an authenticated encrypted session with the SafeBoot client for purposes of configuring the module. All communications between the management software and the client are encrypted using DES (with a session key generated using Diffie-Hellman key agreement). DSA is also used during the Diffie-Hellman key agreement to authenticate the server to prevent server spoofing.

2.4 Physical Security

The SafeBoot Client is a software module intended for use with the Microsoft Windows 95 SR2 operating system but will operate under Microsoft Windows 98, Microsoft Windows NT 4.0 or Microsoft Windows 2000. For FIPS 140-1 purposes, the module was validated against Level-1 FIPS 140-1 physical security requirements when running on a standard Intel-compatible personal computer with the Windows 95 SR2 operating system. This platform meets all Level-1 FIPS 140-1 physical security requirements, providing a multi-chip standalone module with production grade equipment, standard passivation, and a strong enclosure.

Although the SafeBoot Client consists entirely of software, the FIPS 140-1 validated platform is a standard PC which has been tested for and meets applicable Federal Communication Commission (FCC) Electromagnetic Interference (EMI) and Electromagnetic Compatibility (EMC) requirements for business use as defined in Subpart B of FCC Part 15.

2.5 Cryptographic Key Management

The module uses a variety of keys, including: hard disk encryption key, user encryption keys, session keys, recovery keys, database key (when used with a local Object Database only), integrity check keys and server public key. The following table lists all keys. Currently, AES is the only approved encryption algorithm in the SafeBoot Client product and all encryption keys are AES keys. The server public key is a DSA key.

Key type	Purpose
Hard disk encryption key	To encrypt hard disk contents; to authenticate client to the Object Database; to encrypt database key (when local Object Database is used)
User encryption keys	To encrypt secure user attributes
Server public key	To authenticate the Crypto Officer communications
Machine recovery key	Encryption key used to recover the hard disk
User recovery keys	To recover user encryption keys
Database key	Used only with local Object Database to protect certain

	attributes
Integrity check key	Used to perform module integrity check
Session keys	To encrypt traffic between client and remote server
Diffie-Hellman Keys	Used to establish session keys

Table 3 Keys used by SafeBoot Client

2.5.1 Key generation

The SafeBoot Client generates symmetric key material using a FIPS 186-2 Appendix 3.3 compliant pseudo-random number generator.

2.5.2 Key entry and output

All key material, excluding recovery key information, is entered and output from the module in encrypted form. Recovery key information can be entered manually in plaintext form, electronically in plaintext form or electronically in plaintext form. When entered manually, correct key entry is verified using a checksum.

2.5.3 Key storage

Key material is stored in the SafeBoot File System (SBFS). All key material is encrypted using a FIPS-approved algorithm prior to storage. All sectors of the SBFS feature a checksum to guard against modification.

2.5.4 Protection of key material

The SafeBoot Client securely manages key material for the lifetime of the key. All key material is encrypted with AES prior to storage in the SBFS and prior to export.

2.5.5 Zeroization of key material

All key material mentioned in table 3 above (the complete list of unprotected CSPs), associated with a machine is zeroized when the SafeBoot Client is uninstalled. All user encryption key material associated with users is zeroized when the user is deleted.

2.6 Cryptographic Algorithms

The SafeBoot Client supports the following algorithms:

- FIPS-approved algorithms: AES, DSA, and SHA-1.
- Non FIPS-approved algorithms: Diffie-Hellman

2.7 Self-Tests

The SafeBoot Client implements both power-up and conditional self tests as required by FIPS 140-1. The following two sections outline the tests that are performed.

2.7.1 Power-up self-tests

The following power-up self-tests performed by the module:

SHA-1 known answer test
DSA pair-wise consistency test
AES known answer test
Critical Functions (Configuration file signature verification test)
Software Firmware load test (Signature verification)

Each of these tests are executed when the computer is turned on and the module first executes. If any of these tests fail, the module will not load. The module must be reset to re-execute these tests.

2.7.2 Conditional self-tests

There are two conditional tests that are run by the module. A continuous random number generator test is run every time the module requests a random number. Failure of this test may result in keys not being generated and an appropriate error message will be given. A test is also done when a software update occurs. All files are digitally signed and this signature is checked prior to any update of the software. There is also a manual key entry test that verifies correct entry of the user recovery keys and machine recovery key. More information on this test can be found in chapter 18 of the Administrator's Guide.

3 FIPS Mode

The following two criteria must be met to operate the SafeBoot Client product in a FIPS approved mode:

1. The SafeBoot Client must be installed using a FIPS approved algorithm. You can check which algorithms are certified by looking in the file changes.html" in the root of the install cd, or by checking NIST's web site. The AES crypto algorithm is certified for use in FIPS 140-1 implementations. The validated version of the SafeBoot Client presents AES the only option for the encryption algorithm.

2. All data and operating system partitions on the machines where the SafeBoot client has been installed MUST be fully encrypted. You can check the conformance to this issue by viewing the SafeBoot client status window – if any drives are highlighted in red then they are not fully encrypted.