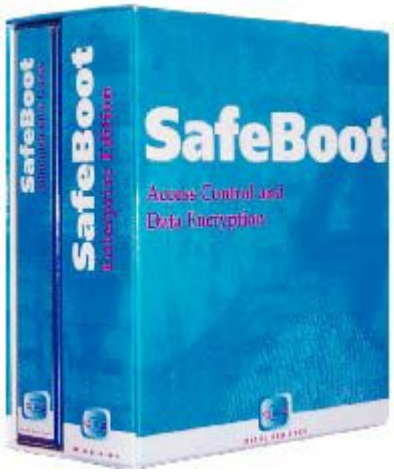
CONTROL BREAK INTERNATIONAL

Control Break International
SafeBoot Client Version 4.2



FIPS 140-2 Non-Proprietary Security Policy

Level 1 Validation

Revision 1.8, January 2005

Table of Contents

Table of figures

1 INTRODUCTION

1.1 Purpose

This is the non-proprietary FIPS 140-2 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 140-2 (FIPS 140-2) 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 (CMVP), please visit www.nist.gov/cmvp.

1.3 Document Organization

This Security Policy document is one part of the FIPS 140-2 Submission Package. In addition to this document, the 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 140-2 requirements. With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 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.

The cryptographic boundary of the module is the case of the PC on which it is installed. See Figure 1. SafeBoot replaces the boot sector of the hard disk to provide effective access control and optionally encrypts part or all of the hard disk drive. The module is a software module running on a standard PC. The processor of the PC executes all software. All software components of the module are stored on the hard disk, and, while executing, are stored in the RAM.

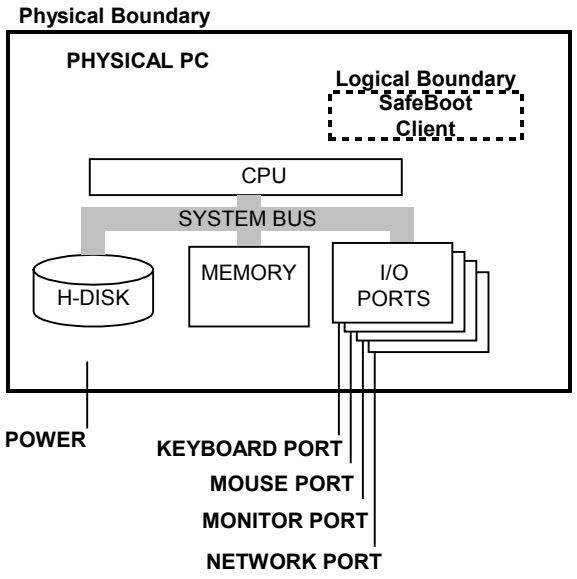


Figure 1 Block Diagram of cryptographic boundary

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 user's 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 user's 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 140-2 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 140-2 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 2.8) 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 Operational Environment

The cryptographic module has a modifiable operational environment and runs on a standard Intel-compatible PC running a variant of the Windows operating system. The module was tested under Windows 2000 Professional and Windows XP Professional, configured as single-user, using a DELL OptiPlex GX1 General Purpose Computer (GPC) as the test platform. The module may be ported to other GPCs and variants of the Microsoft Windows operating systems by following the guidelines of CMVP Implementation Guidance G.5.

The cryptographic module runs in its own operating system threads. This provides it with protection from all other processes, preventing access to all keys, intermediate key generation values and other CSPs.

The task scheduler and architecture of the operating system maintain the integrity of the cryptographic module.

The cryptographic module is protected using token-based access control. Users need to possess a token and provide a secret associated with the token in order to logon. SafeBoot supports a number of different tokens, including Floppy disk, Smartcard, Aladdin eToken Pro, or a simple password (password-only token). The Floppy disk and simple password are the only tokens supported by the validated version of SafeBoot. The driver files are themselves encrypted on the hard drive of the PC. So, any attacker would either need to possess the appropriate token, or would need to be able to decrypt the hard drive to gain access to the executable code of the cryptographic module. The source code is not included as part of the cryptographic module, and the keys and CSPs are not stored in a plaintext form on the module.

There is no upper limit to the number of Users of the module, although only one operator can have access to the PC that contains the module at a time.

2.4 Roles and Services

The SafeBoot Client meets all FIPS 140-2 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, Figure 2, 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

Figure 2 Roles

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 AES (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.

Figure 3 summarizes the authentication mechanism for each of these roles, and Figure 4 describes the strength of these mechanisms.

Role	Type of Authentication	Authentication Data
User	Identity-based	Password
Crypto Officer	Role-based*	DSS authenticated challenge-response mechanism

Figure 3 Roles and Required Identification and Authentication

*DSS authenticated challenge-response is used to verify the role of Crypto Officer on the machines taking part in the exchange, not the individual or identity assuming that role on those machines. For this reason, this authentication mechanism is regarded as role-based.

Authentication Mechanism	Strength of Mechanism
Password	It is possible to configure the minimum password length and the type of characters that can be used in a password. It is also possible to configure the client to lock up after a specified number of unsuccessful password entry attempts. If a minimum password length of 4 is used, and the password is restricted to alphanumeric characters, this gives a chance of success of 1 in 62^4 or 1 in 14,776,336 for guessing a password, which is greater than required. Control Break International recommend a minimum password length of 5 characters, giving a random chance of success of 1 in 916,132,832. If the software is configured to lock up after 10 unsuccessful attempts, this gives a chance of successfully guessing the password at 1 in 91,613,283. This is significantly better than the acceptable probability of 1 in 100,000.
DSS authenticated challenge-response mechanism	DSS provides a strength of 80 bits, that amounts to 2^{80} (approximately 1.2×10^{24}) possible outcomes. This greatly exceeds the requirement that the probability shall be less than one in 1,000,000 that a random attempt will succeed or a false acceptance will occur. The key size is sufficiently large that many repeated attempts will not reduce the likelihood of success to 1 in 100,000.

Figure 4 Strength of Authentication Mechanisms

2.5 Access to Services

The following table, Figure 5, lists the authorized services linked to each of the Roles offered by the module.

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

Figure 5 Services Authorized for Roles

2.6 Physical Security

The SafeBoot Client is a software module intended for use with the Microsoft Windows 2000 Professional and Windows XP Professional operating systems but will operate under Microsoft Windows 95 SR2, Microsoft Windows 98, Microsoft Windows NT 4.0 or other Microsoft Windows 2000 or XP operating systems. For FIPS 140-2 purposes, the module was validated against Level-1 FIPS 140-2 physical security requirements when running on a standard Intel-compatible personal computer for both the Windows 2000 Professional and Windows XP Professional operating systems. This platform meets all Level-1 FIPS 140-2 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 140-2 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.

Physical security of the SafeBoot Client is currently specified as production grade hardware. The SafeBoot Client is a software component and as such is dependent on the physical security of the device that houses the software. The user is expected to take reasonable measures to ensure the physical security of the module. The SafeBoot Client needs to be run in a secure physical environment. It is intended that the SafeBoot Client be run under a single-operator operating system environment. The physical device should be in a secure location so that it is protected under normal operating conditions.

2.7 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, Figure 6, lists all keys and CSPs. 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/CSP 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
User Password	Used to authenticate user during logon.
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
DSA Test KAT parameters	A fixed set of parameters used to verify the DSA functionality during the DSA known answer test performed at power-up
AES KAT data	Test data is taken from NIST Special Publication 800-38A 2001 Edition "Recommendations for Block Cipher Modes of Operation" This uses fixed parameters to generate cipher text and plain text which is then verified against expected values
PRNG seed values and seed keys	These are used to prevent the output from the PRNG from being predictable to an attacker. Knowledge of the seed values and seed keys is required to predict key values.
PRNG KAT data	Used to test the pseudo-random number generator to verify that it is operating correctly

Figure 6 Keys/CSPs used by SafeBoot Client

2.7.1 Key generation

The SafeBoot Client generates symmetric key material (and the Diffie-Hellman public/private key pair used in session key establishment,) using a FIPS 186-2 Appendix 3.3 compliant pseudo-random number generator. The only symmetric keys generated in this way are the Hard Disk Encryption Key, the Machine Recovery Key and the Session Keys.

2.7.2 Key entry and output

All key material, excluding recovery key information and the Diffie-Hellman public key used in session key establishment, is entered and output from the module in encrypted form. Recovery key information can be entered manually in plaintext form, electronically in plaintext. When entered manually, correct key entry is verified using a checksum.

2.7.3 Key storage

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

2.7.4 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.

2.7.5 Zeroization of key material

All key material mentioned in Figure 6 above (the complete list of unprotected critical security parameters - 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.

2.7.6 Access to key material

The following matrices (Figure 7 and Figure 8) show the access that operators have to specific keys or other critical security parameters when performing each of the services relevant to his/her role.

User role

	Key									
Service	**hdek**	**uek**	**spk**	**mrk**	**urk**	**dk**	**ick**	**sk**	**dhk**	**pwd**
Synchronization	R	R	R				R	R	R	
Encryption/ Decryption	R	R								
Show Status Functions										
Self-test Functions							R			
Change User Password		RW								W

Figure 7 User Role

Crypto Officer role

	Key								
Service	**hdek**	**uek**	**spk**	**mrk**	**urk**	**dk**	**ick**	**sk**	**dhk**
Synchronization	R	R	R			R	R	R	W
Encryption/Decryption	R	R							
Show Status Functions									
Self-test Functions							R		
User/machine recovery requests	RW	RW		R	R		R		
Configuration	R	RW					R		
File Updates							R		
Manage SafeBoot (Cryptographic & Key Management Functions)	R	RW	R	RW	RW	R	R	R	R
Software Updates							R		
User/machine recovery	R	W		R	R				
Set User Attributes (passwords, access rights, etc.)		R			WD				
Change User Attributes		R		R	RW				
Create User Groups – na									
Modify User Groups – na									
Delete User Groups – na									
Create Users	R	RW			RW				
Modify Users		RW			RW				
Delete Users	R	RD			RD				

Figure 8 Crypto Officer role

Access rights

blank None
W Write access
R Read Access
D Delete Access

Note: If a service requires read or write access, it is the service as realized by module processes that requires access to the keys. The operator (either User or Crypto Officer) does not have access to the keys themselves. The operator may change keys or use keys, but in all cases has no plaintext access to key material. The exception to this is the User password, where the User is required to enter the password in plaintext and so has genuine write access to it.

Keys

hdek Hard disk encryption key
uek User encryption keys
spk Server public key
mrk Machine recovery key
urk User recovery keys
dk Database key
ick Integrity check key
sk Session keys

dhk Diffie-Hellman Keys
pwd Password

2.8 Cryptographic Algorithms

The SafeBoot Client supports the following algorithms:

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

2.9 Self-Tests

The SafeBoot Client implements both power-up and conditional self tests as required by FIPS 140-2. The following two sections outline the tests that are performed.

2.10 Power-up self-tests

The following table, Figure 9, lists the power-up self-tests performed by the module:

SHA-1 known answer test
DSA known answer test
AES known answer test
Critical Functions (Configuration file signature verification test)
Software/Firmware integrity test (Signature verification)
Pseudo-Random Number Generator Known Answer Test

Figure 9 Power-up self-tests

Each of these tests is 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.

2.10.1 Conditional self-tests

There are three 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.

2.11 Design Assurance

Control Break International employ industry standard best practices in the design, development, production and maintenance of the SafeBoot product, including the FIPS 140-2 module.

This includes the use of an industry standard configuration management system that is operated in accordance with the requirements of FIPS 140-2, such that each configuration item that forms part of the module is stored with a label corresponding to the version of the module and that the

module and all of its associated documentation can be regenerated from the configuration management system with reference to the relevant version number.

Design documentation for the module is maintained to provide clear and consistent information within the document hierarchy to enable transparent traceability between corresponding areas throughout the document hierarchy, for instance to correspondence between elements of this Cryptographic Mode Security Policy (CMSP) and the design documentation.

Guidance appropriate to an operator's Role is provided with the module and provides all of the necessary assistance to enable the secure operation of the module by an operator, including the Approved security functions of the module.

3 FIPS Mode

The following three 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. The validated version of the SafeBoot Client presents AES as the only option for the encryption algorithm. The AES encryption algorithm is validated for use in FIPS 140-2 implementations.

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.

3. The PC used to run the SafeBoot Client must be built using production grade components.