SecureDoc®

Cryptographic Engine V3.2

Security Policy

Author(s):	Bill Oliver
	WinMagic Incorporated
Revision Date:	2002-04-01

Revision History:

Revision	Changes Since Previous Revision
Version 1.0	- Initial release
Version 1.1	- Clarified requirements for FIPS-approved operating modes - Documented tamper seal installation - Added recommendations for user PIN length - Documented recommend key archiving procedure - Clarified operator roles - Added sign and verify to list of available cryptographic services - AES can now be used in FIPS-approved mode
Version 1.2	- AES is also FIPS-approved for key wrapping (section 6.4)
Version 1.3	- AES-MAC is not yet FIPS-approved

Table of Contents

1 Introduction

This document describes the non-proprietary FIPS 140-1 security policy for the SecureDoc® Cryptographic Engine used in all SecureDoc® cryptographic products.

It describes the various services offered by the Cryptographic Module and the mechanisms provided to ensure that these services meet the FIPS 140-1 level 2 requirements. It also describes the storage of cryptographic data within the engine and how it is protected against tampering and data loss. The management of various roles and restrictions that can be applied to the user in using these services and data are documented.

This document has been prepared in accordance with the requirements of FIPS 140-1 and is not to be seen as a complete description of the product capabilities or applications. Please contact WinMagic at http://www.winmagic.com/ for further information.

2 Product Overview

The SecureDoc® Cryptographic Engine is the heart of all SecureDoc® products. It provides all cryptographic services as well as the services required for key management and to maintain the user key files.

Key database security is ensured by encrypting all the sensitive contents including all cryptographic keys and protecting them with a User PIN. This security is extended in SecureDoc® applications through the use of external tokens like smart cards to further secure and authenticate access to the key file. The Cryptographic Engine also incorporates features that ensure that the key data can be recovered securely by authorised personnel in the event that the user PIN is lost or becomes unavailable.

Key and user management is facilitated with a rich set of user privileges embedded in the key file and attributes associated with the keys themselves. These privileges and attributes can be exploited by applications such as the SecureDoc® Central Database software to control all aspects of key management and usage.

The Cryptographic Engine API is based on the PKCS-11 Cryptoki standard. This standard is widely used by Cryptographic service providers including many providers of security smart cards and other tokens. It provides a rich set of functions designed to support key generation and management as well as all common cryptographic functions and algorithms.

The Cryptographic Engine is designed to integrate closely with the Windows NT operating system taking full advantage of the operating system security. It is available in versions to run either in user mode, for standard windows applications; in kernel mode for use by low-level applications such as the SecureDoc® disk encryptor; or in real mode where it can be used at boot time, before the operating system is loaded.

The Cryptographic Module contains all the software that implements all the services provided by the Cryptographic Engine. It is distributed in one of four versions depending on the application.

The user mode version is distributed as a MS Windows .dll file for use by any windows application. A kernel mode version is also available which is installed as a MS Windows device driver for use by kernel level applications such as the SecureDoc® disk encryption product. The kernel mode version also has a .dll interface available so the driver may be accessed from a user application. A real mode version is also available which can be used at boot time and when the Windows OS is not available. The fourth version is a real mode version that is installed as an “INT 13” hook.

The first three versions of the module provide the same basic services. The cryptographic API conforms to the PKCS-11 Cryptoki standard with extensions. The INT 13 hook version is a special, stripped down version providing the base cryptographic services (encrypt/decrypt) only without the key or key file management functions.

Although all four versions of the module are based on the same source code, only the kernel mode version and its associated .dll interface is currently validated to FIPS 140-1.

Note that the cryptographic module implements multiple mechanisms for such services as encryption, key wrapping, message digesting etc. Some of these mechanisms have not yet been approved by FIPS for these applications. By definition, if an application uses any of these non-approved mechanisms id is no longer using the Cryptographic Engine in a “FIPS-approved” mode. Specifically, to be operating in FIPS-approved mode the application must use DES, Triple DES, or AES for encryption or key wrapping; SHA-1 for message digesting; and DES-MAC, or Triple DES-MAC for digital signatures. See the individual Cryptographic Services (section 6) for details of approved and non-approved mechanisms.

3 The Cryptographic Module

3.1 Implementation

For the purpose of FIPS 140-1, the SecureDoc® Cryptographic Module is classified as a multi-chip standalone module. It has been tested for this purpose on an IBM NetVista Model 6833-40U PC running the Microsoft Windows NT-4 operating system, Service pack 6a. Versions are also available for Windows 95, 98, Windows 2000, and Windows XP.

3.2 Cryptographic Boundary and Module Interface

From the point of view of FIPS 140-1, the cryptographic boundary of the module as a multi-chip standalone module includes the cryptographic module (DLL) itself, the Windows operating system and the IBM PC hardware. The interface to the module is the physical interface to the IBM PC including the mouse, keyboard, modem etc.

Specifically, from this point of view, the interface for the module is defined as follows:

Data input interface	-	The keyboard port, mouse port, floppy disk drive, and Ethernet port
Data output interface	-	The floppy disk drive and Ethernet port
Control input interface	-	The mouse, keyboard, and CPU
Status output interface	-	The video monitor

More directly, an application interfaces to the Cryptographic Module using the standard Windows C language API. This interface corresponds to the PKCS-11 Cryptoki standard with extensions to meet the unique requirements of the application and of FIPS 140-1.

The call interface ensures that each service call results in a return code that positively reflects the success or failure of the request based on the current state of the module. The return value is always either **CKR_OK** indicating success or a specific error code value.

3.3 Operating Modes

The FIPS 140-1 approved version of the Cryptographic Module is installed in Windows NT to run in kernel mode as a device driver. In this mode it can be accessed by kernel mode applications such as the SecureDoc® disk encryptor. A Windows interface DLL provides access to all Cryptographic services in the driver from a regular Windows application.

The module design is such that only one function call can be processed at any one time in a given application thread. Any other functions that try to run/execute will not be processed until the current function processing is completed.

A complete set of self-checks are run before the Cryptographic Module services may be accessed. If an error occurs during the self checks, or a fatal error occurs during the subsequent execution of any of the services, the module enters an error condition and must be re-initialised before it can be used. There is no data output for the application thread while the self-checks are being executed or when it is in error mode.

3.4 Physical Security

Physical security for FIPS 140-1, level 2 is obtained by securing access to the PC components using of tamper seals.

The tamper seals must be installed on the PC in such a way that they will be broken if the cover is removed to gain access to the interior. On a laptop they should be installed on the hard-drive cover and all access covers (ethernet card, memory etc.). See the documentation provided with the product package for details.

4 The Key File

The User Key File identifies a particular user to the Cryptographic Engine and indicates what privileges or restrictions are associated with it. It also provides a secure container for the user's cryptographic keys.

The key file is divided into two parts: a public part and a private part. The private part is only accessible when the user has been properly authenticated to the cryptographic engine.

Following are the contents of the key file used by the Cryptographic Engine:

Public Data:

User ID	-	Identifies the owner of the key file
Access Parameters	-	Data used in conjunction with the user PIN to access the private portion of the key file
Recovery Data	-	Used to recover the key file in case the user's PIN has been lost

Private Data:

Authorisation Vector	-	Indicates what rights have been granted to the user of this key file
Secret Key	-	A special encryption key associated with this key file and used for special functions such as key backup and key file recovery. It is also available for regular cryptographic operations
User Keys	-	General purpose keys which for various cryptographic operations

Application may add additional objects to the key file provided the key file has sufficient privileges.

4.1 Key File Protection

The public data is encrypted by a fixed key known by SecureDoc® Cryptographic Engine. The private data in the key file is encrypted with a key file key, generated when the key file is created. This key is stored in the public part of the header wrapped with an encryption key derived from the User PIN using cryptographic parameters (salt, iteration count) stored in the Access Parameters.

The User PIN may be a password entered by the user or it may be a PIN of arbitrary length generated and kept secure in another location, for example using a cryptographic smart card. The user PIN should be at least five characters in length.

A "Recovery PIN" may also be generated by the cryptographic engine and used to wrap the key file encryption key. The PIN is calculated cryptographically from the Secret Key and may be used in conjunction with the recovery data to access the key file if the user PIN is lost or becomes unavailable.

4.2 Authorisation Vector (AV)

The AV contains a series of flags that indicate what privileges are granted to this key file. Note that the AV itself is part of the private portion of the key file and cannot be viewed until the user has successfully logged on.

The privileges used by the Cryptographic Engine include the following:

Modify PIN	-	Ability to change the User PIN
Create Keys	-	Ability to create user keys in the key file
Modify Keys	-	Ability to modify the attributes of the user keys in the key file
Delete Keys	-	Ability to delete the user keys in the key file
Export/View Keys	-	Ability to access the actual key data in the key file for the purpose of backing up or transferring the keys to another key file (note 1)
Modify AV	-	Ability to modify the privileges in the AV (note 2)
Perform Self Test	-	Ability to perform an explicit self test operation on the Cryptographic Engine (see section 6.1)
Create Special Object	-	Required to create special key file objects (note 3)

Notes:

1. Keys with the attribute “never exportable” cannot be accessed even if the “export/view keys” privilege is set (see 4.3).
2. The “modify AV” privilege allows privileges to be revoked only. Privileges once revoked may never be granted again.
3. The “Create Special Object” privilege is required to create the special Secret Key object in the key file. Normally this privilege will be cleared once the key file is created along with the secret key.

4.3 User Keys

When the key file is created, it does not contain any keys. Normally the application will immediately create and install the special secret key object. This key may be used by any application for functions associated with this user (e.g. file encryption etc.) and is also used for key file recovery.

Providing the key file has the necessary privileges, additional keys may also be generated or transferred into it from other sources. For example, a common key generated on one key file can be transferred into other key files to allow the users to share data.

The keys in the key file are associated with a number of attributes that govern their usage and how they can be accessed:

Key ID	-	Name of key (for identification purposes) (note 1)
Key type	-	Type of key (DES, Triple DES, AES) (note 2)
Usage	-	What the key can be used for (encrypt, decrypt, wrap, unwrap) (note 1)
Sensitive	-	The key cannot be extracted unless the key file has “Export/View Keys” privilege (note 3, 6)
Admin Sensitive	-	The key cannot be extracted even if the key file has “Export/View Keys” privilege (note 3, 6)

Extractable	-	Key can be extracted wrapped with the special key file “secret key” (note 5, 6)
Always Sensitive	-	The key has always had the sensitive attribute set since it was placed on this card (note 4)
Always Admin Sensitive	-	The key has always had the Admin Sensitive attribute set since it was placed on this card (note 4)
Never Extractable	-	The key has never had the Extractable attribute set since it was placed on the card (note 4)
Local Key	-	The key was generated in this key file rather than imported from another source (note 4)

Notes:

1. The key usage attributes may be modified only if the key file has “Modify Keys” privilege.
2. “Key ID” and “Key Type” cannot be changed once the key is created.
3. The “Sensitive” and “Admin Sensitive” attributes may be set only if the key file has “Modify Keys” privilege. Once the bit is set, it cannot be cleared.
4. The “Always Sensitive”, “Always Admin Sensitive”, “Never Extractable”, and “Local Key” attributes are set by the Cryptographic Engine (read only)
5. The “Extractable” attribute may be cleared only if the key file has “Modify Keys” privilege. Once it has been cleared it cannot be set again.
6. The “Extractable” attribute is not affected by “Sensitive” or “Admin Sensitive”. If both “Extractable and Admin Sensitive” are set, the key can only be extracted with the key file secret key.

See the section on key management (6.4) for more information on how the keys data is managed.

4.4 Archiving Keys

Keys can be archived or backed up from the key file in a number of ways. Typically, a key is created with the “Extractable” attribute. It may then be extracted from the key file wrapped with the key file secret key. It may be backed up either as part of a master key file or in another format.

The SecureDoc® Central Database application is available to administer a comprehensive and secure key archive for cryptographic engine key files.

5 Operator Roles

SecureDoc® uses identity-based authentication to authorise a given operator to access the various services in the module based on the Authorisation Vector in the Key File. An operator's role is defined as the set of services allocated by the AV to the user of a specific key file.

Following is the typical privileges that would be allocated to distinguish a cryptographic user role from that of a crypto-officer:

Privilege	User	Crypto-Officer	Comments
Modify PIN	X	X	- Allows an operator to change his PIN
Create Keys		X	- Allows a Crypto-officer to create new keys to allocate to specific users
Modify Keys		X	- Allows a Crypto-officer to modify the attributes of keys
Delete Keys		X	- Allows a Crypto-officer to delete keys no longer required
Export/View Keys		X	- Allows a Crypto-officer to backup keys or transfer them to other key files
Modify AV			- Normally cleared once the key file has been set up
Perform Self Test		X	- Crypto-officer can perform explicit self checks; operator performs self checks implicitly when he authenticates to the module
Create Special Object			- Normally cleared once the key file has been set up

Typically, the cryptographic officer's key file may contain all the keys used throughout the organisation. When he creates a key file for a new user, he transfers some enterprise keys from his key file into the new key file. In addition, he may generate some specific new keys allocated to that user. The attributes on the keys are set to restrict usage to encrypt and decrypt only (wrap and unwrap attributes are not set). The keys placed on the card would also be set so the operator cannot export them (set as sensitive, non-extractable). Once the new keys have been backed-up, all the privileges in the operator's key file not required for normal use will be deleted as shown above.

With the key file set up like this, once the operator has been authenticated he is restricted to access the module services as follows:

Service	User	Crypto-Officer	Comments
Cryptographic Services	X	X	- User can encrypt, decrypt, and/or digest using keys placed in the key file by the cryptographic officer

Key Generation		X	- User cannot create keys and store them in the key file (does not have “Create Key” privilege)
Key Export		X	- User cannot export keys from the key file (all keys are marked sensitive, non-extractable)
Key Import		X	- User cannot import keys into the key file (there is no unwrapping key available)
Create Key File	X	X	- The user can create a new key file, however he will not be able to transfer enterprise keys into that new file because he does not have export privileges for those keys

6 Cryptographic Services

6.1 Initialisation and Self Tests

Before any of the cryptographic services are accessed by an application, the module must be initialised. At this time a comprehensive set of self-tests are run on the various cryptographic algorithms to ensure that the module is functioning properly and that it has not been compromised.

Any user logged into a key file with "Perform Self Test" privilege may also re-run these tests on demand to validate the module.

If any self-test fails, the module enters an error state and must be successfully re-initialised before it can be used again.

The following services are provided for initialisation and self-tests:

Initialise	-	Initialises the Cryptographic Engine and performs self tests to ensure it is operational
Module Information	-	Retrieves information about the Cryptographic Engine including version number etc.
Self Test	-	Explicitly executes the self tests (requires "Perform Self Test" privilege)

Following are the self-tests that are performed by the module:

Software Integrity Test	-	Performed automatically when the module is initialised
	-	Verifies that the cryptographic module has not been compromised
Cryptographic Algorithm Test	-	Performed automatically when the module is initialised and on demand via the self-test service
	-	Executes a known answer test on the algorithms using all available mechanisms
PRNG Test	-	Performed continuously each time the pseudo random number generator is used
	-	Checks the PRNG output against previous values to ensure that it never returns the same result twice in succession.

6.2 Session Management

In order to access any of the services, the application must open a session on the module. A session is essentially an execution context for performing cryptographic operations on a given key file. Multiple sessions can access the same or different key file. One application can open several sessions accessing different key files to, for example, transfer keys from one to another. Once an application has been authenticated (logged in) to a key file a new session opened for that key file does not have to be re-authenticated.

Sessions opened by one application (process) cannot be accessed by a different application. Session specific data is protected between applications by the operating system.

The following services are used for session management:

Open Session	-	Initiates a new session
Close Session	-	Terminates a session, cleaning up all temporary session data
Login	-	Authenticates the user to a key file for this session
Logout	-	Logs the user out of a key file

6.3 User and Key File Management

Any user that has been successfully authenticated to the Cryptographic Engine using one key file can create a second key file. A new key file is created with full privileges allowing the user, once logged in, access to all the services.

Note that although the user can create new keys in the new key file, he will not be able to transfer existing keys from another key file unless the second key file has the necessary privileges (export/view keys).

When a new key file is created, the Cryptographic officer will commonly generate the key file secret key any other keys required by the user and either back them wrapped with one of his keys or transfer them to his key file for escrow purposes. He may then transfer one or more enterprise keys from his key file to the new one so the new user may access them. After setting the necessary attributes on the keys, he will revoke all privileges for the new key file except “Use Keys” and “Modify PIN”. The key file may then be copied to the new user’s PC.

The following services are supported for user and key file management:

Create Key File	-	Creates an empty key file
Set Key File Attributes	-	Sets the various key file attributes (user name, access parameters, AV etc.)
Open Key File	-	Opens a specified key file, making it available to be logged in etc.
Set PIN	-	Updates the user PIN

6.4 Key Generation and Key Management

The cryptographic engine supports generation of two types of keys: temporary keys and permanent keys. Temporary keys may be generated by any user logged into any key file. Temporary keys are destroyed when the current session ends unless the values have been backed-up somewhere external to the cryptographic engine.

Permanent keys are stored in the key file. To create a permanent key the key file must have “Create Keys” privilege.

The cryptographic engine supports incorporates a pseudo-random number generator (PRNG) used for key generation. The output of the generator is continuously supervised for “failure to a constant value”. To ensure randomness, the PRNG must be seeded from a good random source. SecureDoc® applications continuously seed the random number generator with random data based on mouse movement, keyboard activity, and the system clock. The PRNG uses the ANSI X9.17 random number generation algorithm.

The values of the user keys in the key file can be accessed either directly or by wrapping with another key. Using these techniques the keys can be backed up or transferred between key files.

The key attributes “Sensitive”, “Admin Sensitive”, and “Exportable” (see 4.3) control how individual keys can be viewed or backed up. The “Always Sensitive”, “Always Admin Sensitive”, and “Never Exportable” attributes can be used to indicate if the keys have ever been backed up or viewed.

The following Key Management services are supported:

Generate Key	-	Generate a new key
Create Key	-	Create key from it’s value
Delete Key	-	Deletes a key and zeros the memory
Modify Key Attributes	-	Modifies selected key attributes (see 4.3)
Get Key Attributes	-	Modifies key attributes (see 4.3)
Find Key	-	Looks up keys in the key file
Wrap Key	-	Wraps a key with another key (using DES, Triple DES, or AES)
Unwrap Key	-	Unwraps a key which has been wrapped with another key
Generate Random Number	-	Generates a random number using built in pseudo-random number generator
Seed PRNG	-	Seeds the pseudo-random number generator

6.5 Cryptographic Services

The following Cryptographic services are supported by the Module:

Encrypt	-	Encrypt using DES, Triple DES, or AES mechanisms
Decrypt	-	Decrypt using DES, Triple DES, or AES mechanisms
Digest	-	Digest a block of data using SHA-1 or RIPEMD 160 mechanisms
Sign	-	Sign data using DES-MAC, Triple DES-MAC, or AES-MAC mechanisms
Verify	-	Verify signed data using DES-MAC, Triple DES-MAC, or AES-MAC mechanisms

To perform any of the above services with a given key, the key must have the appropriate “usage” attribute set (see 4.3).

Note that the RIPEMD 160 and AES-MAC mechanisms have not yet been approved by FIPS. For the cryptographic module to operate in FIPS approved mode SHA-1 must be used for message digest and DES-MAC or Triple DES-MAC for signing and verification.