SecureDoc® Disk Encryption Cryptographic Engine

Security Policy

Abstract: This document specifies Security Policy enforced by the SecureDoc Cryptographic Engine compliant with the requirements of FIPS 140-2 level 1. It includes definition of the SecureDoc Cryptographic Engine as multi-chip standalone cryptographic module and specifies security rules under which the SecureDoc Cryptographic Engine operates.

Author(s):	Alexandr Mazuruc
Module Version:	4.5
Document Version:	1.2 (approved)
Validation:	FIPS 140-2 level 1

Approvals:

Person approved the document	Date
Thi C. Nguyen-Huu, WinMagic Inc.	3 February 2006
Thi C. Nguyen-Huu, WinMagic Inc.	15 March 2006
Thi C. Nguyen-Huu, WinMagic Inc.	20 July 2006

Revision History Table:

Revision	Date	Changes Since Previous Revision	Revision Author
1.0	2 February 2006	This is the initial draft of the document.	Alexandr Mazuruc
1.1	15 March 2006	Modifications resulted from the testing are incorporated.	Alexandr Mazuruc
1.2	20 Jul, 2006	NIST comments addressed	Alexandr Mazuruc

Table of Contents

Introduction

1.1 Purpose

This document describes the non-proprietary FIPS 140-2 security policy for the SecureDoc Cryptographic Engine used by all SecureDoc® cryptographic products.

The document describes the various services offered by the SecureDoc Cryptographic Engine and the mechanisms provided to ensure that these services meet the FIPS 140-2 level 1 requirements. It also addresses storage of cryptographic data within the SecureDoc Cryptographic Engine and protection measures against tampering and data loss. The management of various roles and restrictions that can be applied to the user using these services and data is documented as well.

This document has been prepared in accordance with the requirements of FIPS 140-2 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

2.1 SecureDoc Cryptographic Engine Installation

The software comprising the SecureDoc Cryptographic Engine is provided as Windows Install Shield package and is installed by running a single executable file. The installation procedure consists of several stages:

1. On initial stage operator goes through general steps like choosing installation folder, confirming end user agreements, etc. Once the package is installed, the computer reboots to activate SecureDoc kernel driver.

2. On next login, SecureDoc Wizard starts and governs operator through initial configuration. The configuration procedure includes the following:

    a. Generation of user keys and creation crypto-officer's key file.

    b. Selecting the hard drives to be encrypted.

    c. Installation of Boot Logon component for pre-boot authentication.

    d. Optionally the Emergency Disk may be created on a floppy or USB stick.

    The stage also reboots computer once finished.

3. To finalize the installation operator has to pass pre-boot authentication after second reboot. Then disk encryption process starts automatically as operator logs in to Windows.

2.2 SecureDoc Cryptographic Engine Functionality

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 maintenance of the users' key files.

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

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 SecureDoc® applications such as SD Control Centre, SD Key Management, and SD Enterprise Server to control all aspects of key management and usage. The SecureDoc 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.

A particular operator is identified to the SecureDoc Cryptographic Engine by User Key File. The key file indicates which cryptographic keys and what privileges or restrictions are associated with the operator. Key database security in user's key file is ensured by collecting sensitive information including all user's cryptographic keys and privileges in a encrypted secure container. The secure container is protected with a User PIN or with external cryptographic tokens, smart cards, etc. to further secure access to the key file via multi-factor authentication.

3 Cryptographic Module Definition

3.1 SecureDoc Cryptographic Engine Security Boundary and Interface

From the point of view of FIPS 140-2, the SecureDoc Cryptographic Engine running on a general purpose computer (hence GPC) is validated as a multiple-chip standalone cryptographic module. Thus, two boundaries are distinguished:

- Logical Boundary that includes only the SecureDoc Cryptographic Engine validated as software implementation
- Physical Boundary that includes also the PC hardware needed to run the SecureDoc Cryptographic Engine

These boundaries are shown in the Figure 1 below:

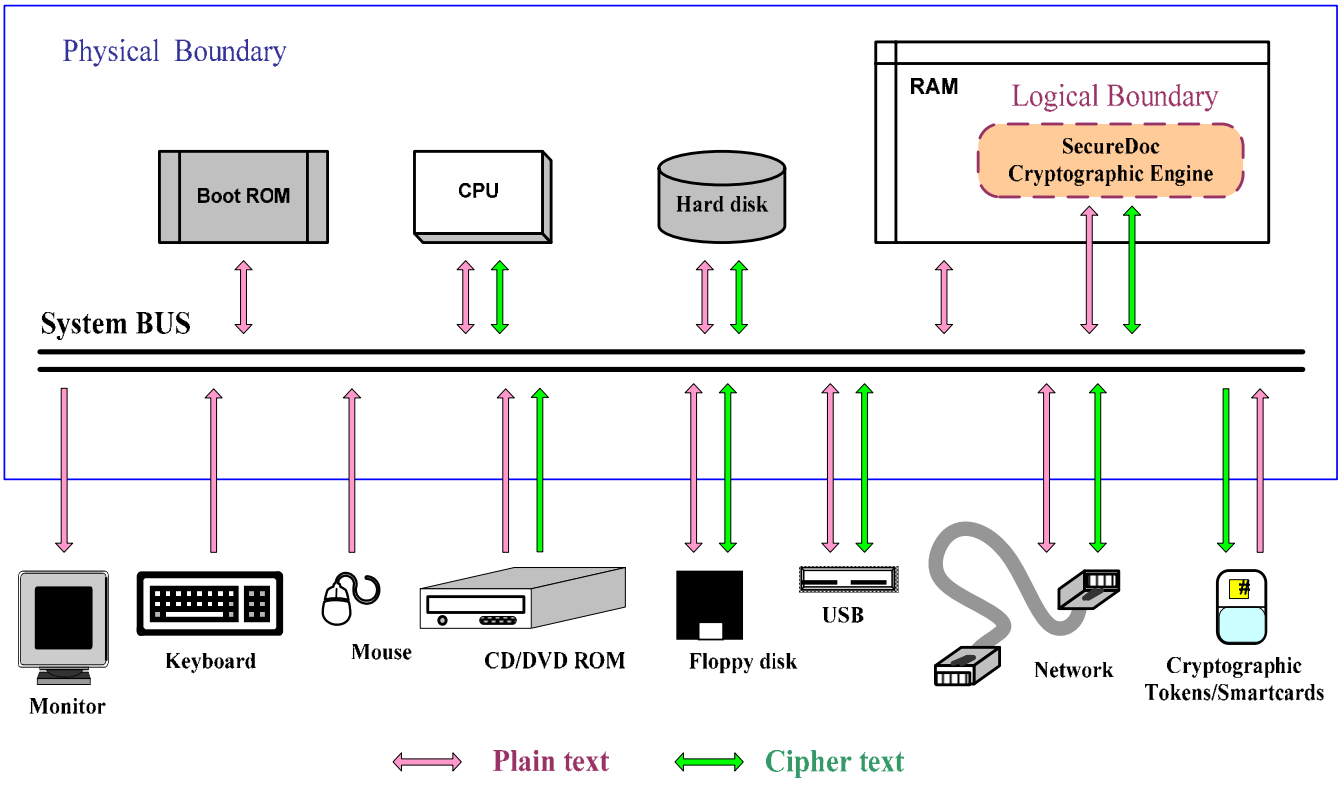

Figure 1. SecureDoc Cryptographic Engine Block-Diagram

The interface to the SecureDoc Cryptographic Engine is the physical interface to the GPC including the mouse, keyboard, video monitor, etc. as figured above. The correspondence between logical interfaces and physical ports of the module is provided in the table below. Different interfaces are kept logically separated while using the same physical ports through utilizing different sets of Input/Output commands sent to a physical port by each of the interfaces that share the port or by invoking different API calls for different interfaces.

Logical Interface	Interface purpose	Physical Port External Devices
Data input interface	Enter the data to be processed by the SecureDoc Cryptographic Engine	Keyboard port, mouse port, floppy disk drive, CD/DVD ROM, USB, Ethernet port, USB token, and Smart Card token
Data output interface	Output the data been processed by the SecureDoc Cryptographic Engine	Floppy disk drive, USB port, Ethernet port, USB token, and Smart Card token
Control input interface	Enter the data used to control operation of the SecureDoc Cryptographic Engine	Mouse, keyboard, and CPU
Status output interface	Show status of the SecureDoc Cryptographic Engine and error messages	Video monitor
Power interface	Provides power for the SecureDoc Cryptographic Engine operations	110V power interface

Table 1. Ports and logical interfaces correspondence

3.2 Module Operational Levels

The SecureDoc Cryptographic Engine operates at three different levels as shown in the picture below. Once the module goes through Power-Up at hardware level, the SecureDoc Cryptographic Engine performs authentication of the operator at the boot level. Successful authentication results in initiation the procedure of loading operating system to go to the kernel level. Once operating system is loaded, the SecureDoc Cryptographic Engine is active in Windows environment and controls critical operations as a component of the SecureDoc kernel filter driver. On the next step the operator has to login to Windows to enter the user level in which SecureDoc and other general purpose applications work. At this level, all media access operations go through the kernel filter driver. Some SecureDoc applications may also run the SecureDoc Cryptographic Engine in user mode to perform cryptographic operations in memory.

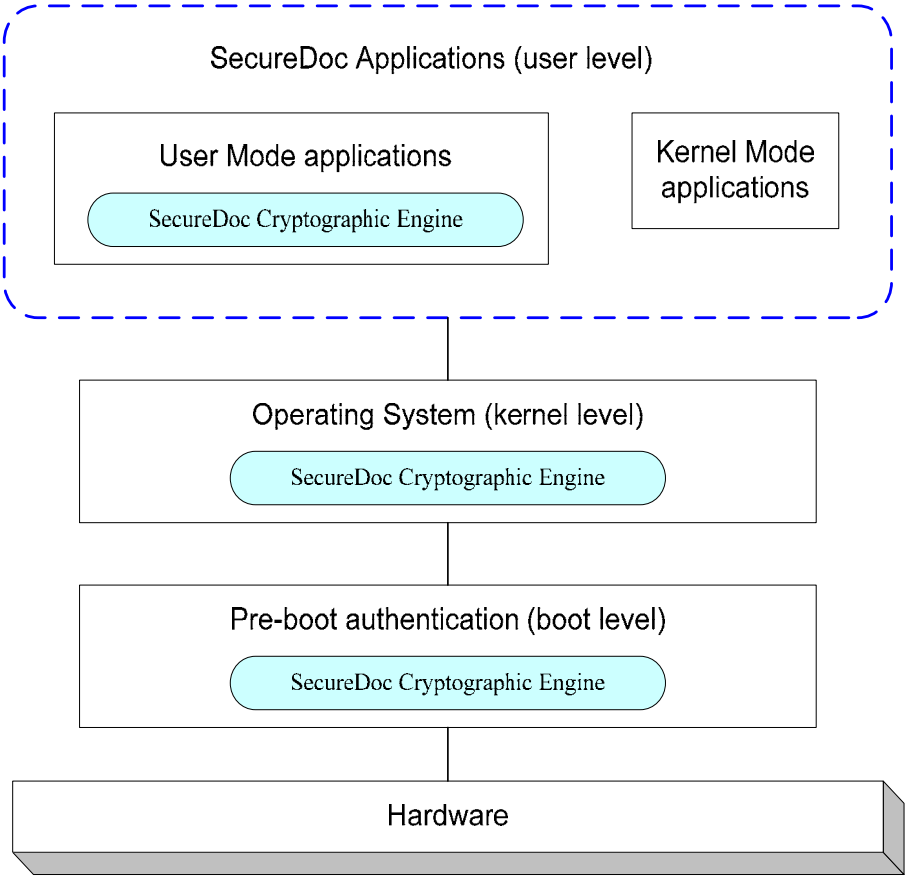

Figure 2. Module operational levels

3.3 Implementation

The SecureDoc Cryptographic Engine is designed to integrate closely with the Windows operating system taking full advantage of the operating system security. The SecureDoc Cryptographic Engine may run in user, kernel or real mode depending on the software component requesting a cryptographic service.

In user mode, the SecureDoc Cryptographic Engine runs as MS Windows DLL for use by Windows applications. In kernel mode, the SecureDoc Cryptographic Engine is installed as MS Windows device filter driver. Other kernel components and user mode applications access it in this case via kernel calls. At pre-boot time when Windows OS is not running, the SecureDoc Cryptographic Engine operates in real mode intercepting I/O requests to hard disks and other storage media.

From the programmatic point of view, application interface to the SecureDoc Cryptographic Engine uses the standard Windows
C-language API. This interface corresponds to the PKCS-11 Cryptoki standard with extensions needed to meet the unique requirements of the SecureDoc® Disk Encryption product and of FIPS 140-2.

The calling 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 SecureDoc Cryptographic Engine. The return value is always either **CKR_OK** indicating success, or a specific error code value.

The SecureDoc Cryptographic Engine design is such that only one function call can be processed at any time in a given application thread. Any other threads that try to run/execute the same cryptographic service will not be processed until the current function processing is completed.

3.4 Operational Environment

The SecureDoc Cryptographic Engine classified as a multiple-chip standalone module runs in general purpose operational environment. This environment includes:

- GPC hardware required for the SecureDoc Cryptographic Engine to operate
- Microsoft Windows 2000, XP or 2003 operating system

SecureDoc Cryptographic Engine is "vendor affirmed" compliant to FIPS 140-2 Level 1 on other compatible single user OS platforms that meet the requirements of IG G.5 of the FIPS PUB 140-2 Standard.

3.5 Physical Security

The SecureDoc Cryptographic Engine does not provide physical security for the module as it is implemented completely in software. Nevertheless, the whole content of operational environment including OS and the SecureDoc Cryptographic Engine are protected with encryption that prevents an attacker from breaking the logical boundary even if the physical boundary is not secure maintained.

Due to above no physical security policy is imposed.

3.6 Mitigation of Other Attacks

The SecureDoc Cryptographic Engine is not designed to mitigate any known specific attacks.

3.7 FIPS Approved Mode of Operation

The SecureDoc Cryptographic Engine implements cryptographic algorithms listed in the tables below: Only FIPS 140-2 approved algorithms are employed in the SecureDoc Cryptographic Engine v.4.5 and it consequently always operates in FIPS approved mode.

Algorithm	Cryptographic Function	Modes / Mechanisms	Output Block (bytes)	Key Size (bits)	Certificate #
AES	Encipherment	ECB, CBC	16	256	359
SHA	Hashing	SHA-1,256, 384, 512	20, 32, 48, 64		434
HMAC	Message authentication	SHA-1, 256, 384, 512	20, 32, 48, 64	256	158
PRNG	Random number generation	ANSI X9.31 AES	16	256	172

Table 2 Algorithms in Approved Mode of Operation

3.8 Self-Tests

At the time initialisation and before any cryptographic service may be accessed, the SecureDoc Cryptographic Engine runs a comprehensive set of self-tests on the implemented cryptographic algorithms to ensure that the SecureDoc Cryptographic Engine is functioning properly and that it has not been compromised.

If an error occurs during a self-test or a fatal error occurs during the subsequent execution of any of the services, the SecureDoc Cryptographic Engine enters the error state and must be re-initialised before it can be used again. There is no data output for the application thread while self-tests are being executed or when it is in the error state.

Below is the list of self-tests performed by the SecureDoc Cryptographic Engine:

Test	Type	Actions performed
Software Integrity Test	Power-Up	Performed automatically when the SecureDoc Cryptographic Engine is initialised Verifies that the SecureDoc Cryptographic Engine has not been compromised
Cryptographic Algorithms Test	Power-Up	Performed automatically when the SecureDoc Cryptographic Engine is initialised and on demand via the self-test service Executes Known Answer Tests for all employed algorithms and available mechanisms
PRNG Continuous Test	Conditional	Performed each time the pseudo random number generator is used PRNG output is compared with the previous block of generated data

Table 3. Self-tests performed by the SecureDoc Cryptographic Engine

Self-test for PRNG additionally includes the following statistical tests:

Monobit Test, Poker Test, Runs Test, Long Runs Test.

4 Cryptographic Key Management

4.1 Encryption Keys

To encrypt users objects (hard drive, floppies, USB media, files and folders) the SecureDoc Cryptographic Engine creates Data Encrypting Keys (DEK) and Key Encrypting Keys (KEK).

Outside the cryptographic boundary, these keys are stored as PKCS-11 cryptographic objects. Each object contains the following data:

- KEK ID – identifies the key used to wrap the key (for DEK only)
- Key ID – name of the key (for KEK only)
- Key value – actual encryption key (wrapped in case of DEK)
- Algorithm Info – algorithm type (AES), base IV, etc.

Objects are obfuscated to protect information from casual browsing.

Data Encrypting Keys are stored with the encrypted data as part of the encryption header being wrapped with KEK.

Key Encrypting Keys are stored in User Key File in its private part protected with DEK or as separate objects wrapped with another KEK in the public part.

To protect SecureDoc internal data on the encrypted media and to supply the cryptographic algorithms based on secret keys (HMAC, ANSI X9.31) SecureDoc utilizes a set of 256-bit fixed keys. These keys are stored inside the SecureDoc Cryptographic Engine and are never exported outside the cryptographic boundary.

4.2 User Keys

User Keys are used by the SecureDoc Cryptographic Engine as wrapping keys for DEKs that encrypt user data on hard drive or removable media.

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.). The key 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. The table below details the attributes meaning with references to the Notes.

Attribute	Purpose
Key ID	Name of key (for identification purposes) (1)
Key type	Type of key (AES) (2)
Usage	What the key can be used for (encrypt, decrypt, wrap, unwrap) (1)
Sensitive	The key cannot be extracted unless the key file has “Export/View Keys” privilege (3,6)
Admin Sensitive	The key cannot be extracted even if the key file has “Export/View Keys” privilege (3,6)
Extractable	Key can be extracted wrapped with the special key file “secret key” (5,6)
Always Sensitive	The key has always had the sensitive attribute set since it was placed in key file (4)
Always Admin Sensitive	The key has always had the Admin Sensitive attribute set since it was in key file (4)
Never Extractable	The key has never had the Extractable attribute set since it was placed in key file (4)
Local Key	The key was generated in this key file rather than imported from another source (4)

Table 4. User Key attributes

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 SecureDoc 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.

4.3 Key File Protection

The public data is encrypted by a fixed key known by the 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 a 256-bit Key Encryption Key.

The User PIN may be a password entered by the user or it may be a randomly generated strong PIN (256 bits) and kept secure in another location, for example using a cryptographic smart card. The User PIN should consist of at least five alpha/numeric/special characters. Randomly generated PIN can be also kept in public part of key file wrapped with a key kept on the smart card.

A “Recovery PIN” may be generated by the SecureDoc 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.

4.4 Key Generation

The SecureDoc 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 externally to the SecureDoc Cryptographic Engine.

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

All keys are generated by the PRNG described in 3.6.3 basing on ANSI X9.31 algorithm.

4.5 Key Zeroization

All keys created by the SecureDoc Cryptographic Engine exist as separate PCKS-11 cryptographic objects or as a part of larger ones. Any object including (containing) cryptographic keys can be zeroied by calling C_DestroyObject() API function. When a cryptographic object is destroyed, the memory occupied by the object is cleared by zeros. SecureDoc calls this function when it performs such operation as decryption, removal keys, key files, etc.

4.6 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 key files.

5 Operator Roles

5.1 User privileges

SecureDoc controls access to the various services in the SecureDoc Cryptographic Engine through the Authorisation Vector (AV) in the Key File. An operator's role is defined as a subset of privileges allocated by the AV to the owner of a specific key file after successful authentication.

User Role corresponds to minimal subset of privileges that restricts him or her to default services only. Privileges assigned to an operator in crypto-officer role determine which services are actually available while he or she operates the module.

5.2 Operator Authentication

To be successfully authenticated an operator has to login to a key file that contains the AV defining the role accepted by operator. The next table specifies authentication required for existing roles:

Role	Type of authentication	Authentication data
User	Identity-based	Password, hardware token or smartcard
Crypto-Officer	Identity-based	Password, hardware token or smartcard

Table 5. Roles implemented by the SecureDoc Cryptographic Engine

Authentication data required depend on the type of key file protection as described in Cryptographic Key Management. If randomly generated strong PIN is used to login to key file, then operator must posses the hardware token or smart card that allow to access the PIN.

Strength of authentication mechanisms employed is shown below:

Authentication Mechanism	Strength of Mechanism
Password	The minimal mandatory length and complexity of password may be configured by crypto-officer for any key file via password rules. SecureDoc manual recommends this length to be at least 5 alpha/digit/special characters. The number of possible passwords in this case (7,339,040,224) is big enough to guarantee the required 1 in 1,000,000 probability of guessing the password. The SecureDoc Cryptographic Engine may also be configured to block the input interface after a certain number of unsuccessful login attempts forcing the operator to reset the SecureDoc Cryptographic Engine. For the length of password mentioned above the limit of 10 unsuccessful attempts gives a possibility less than 1 to 100,000 to guess the password within one minute.
Hardware token or smartcard	With recommended length of token PIN set to 6 and more symbols and taking in attention the fact that tokens and smartcards become completely blocked after3-5 unsuccessful login attempts the estimation made above is still applicable to token protection as well.

Table 6. Strength of authentication mechanisms implemented by the SecureDoc Cryptographic Engine

6 Cryptographic Engine Services

6.1 Services implemented

The table below contains a full list of services implemented in the SecureDoc Cryptographic Engine.

Service	Category	Actions performed
Initialise / Self Test	Administrative	Initialises the SecureDoc Cryptographic Engine and performs self tests to ensure it is operational
On Demand Self Test	Administrative	Explicitly executes the self tests (requires "Perform Self Test" privilege)
Show Module Status	Administrative	Indicates status of the SecureDoc Cryptographic Engine (disk encryption, etc.), shows error codes produced
Change PIN	Administrative	Updates the user PIN
Generate Key	Key Management	Generate a new key
Zeroize Key	Key Management	Deletes a key and zeros the memory
Import Key	Key Management	Import key from another key file
Export Key	Key Management	Export key from key file
Archive Key	Key Management	Backup key wrapped with another key
Encrypt	Cryptography	Encrypt using AES algorithm
Decrypt	Cryptography	Decrypt using AES algorithm
Digest	Cryptography	Digest a block of data using SHA-1 or SHA-256 algorithm
Message Authentication	Cryptography	Sign / Verify data using HMAC-SHA-1 or HMAC-SHA mechanism

Table 7. Services provided by the SecureDoc Cryptographic Engine

6.2 Administrative Services

Before any of the cryptographic services are accessed by an application, the SecureDoc Cryptographic Engine must be initialised. An operator logged into a key file with "Perform Self Test" privilege may also re-run these tests on demand to validate the SecureDoc Cryptographic Engine.

Any user that has been successfully authenticated to the SecureDoc 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.

6.3 Key Management Services

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 4.1) 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.

A key that a service operates with is always wrapped with another key when exported from the SecureDoc Cryptographic Engine. An exported key may be located in the encryption header of an encrypted data or a key file.

6.4 Cryptographic Services

To perform any of the above services with a given key, the key must have the appropriate “usage” attribute set (see 4.1).

The result of the “Decrypt Service” is plaintext deciphered data. “Verify Service” reports whether or not a cryptographic signature is valid. Other services always produce cipher-text.

7 Access Rules

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. Fixed keys are used to protect some system information involved in self-testing and logging events related to the SecureDoc Cryptographic Engine.

With the key file set up like this, once the operator has been authenticated he is restricted to access the key material and CSP in the accepted role through available services as specified in the table below:

Service	Role	Key Material and CSP			
		DEK	KEK	Fixed Keys	User PIN
Initialise / Self-Test	Crypto-Officer			Read	
	User			Read	
On Demand Self-Test	Crypto-Officer			Read	
	User			n/a	
Show Status	Crypto-Officer				
	User				
Change PIN	Crypto-Officer				Read/Write
	User				Read/Write
Generate Key	Crypto-Officer	Write	Write		
	User	n/a	n/a		
Zeroize Key	Crypto-Officer	Write	Write	Write	
	User	Write	Write	Write	
Import Key	Crypto-Officer		Read/Write		
	User		n/a		
Export Key	Crypto-Officer		Read		
	User		n/a		
Archive Key	Crypto-Officer		Read/Write		
	User		n/a		
Encrypt	Crypto-Officer	Read	Read	Read	
	User	n/a	n/a	n/a	
Decrypt	Crypto-Officer	Read	Read	Read	
	User	Read	Read	Read	
Digest	Crypto-Officer				
	User				
Message Authentication	Crypto-Officer			Read	
	User			Read	

Table 8. Access rules implemented by the SecureDoc Cryptographic Engine

Empty cells in the table above mean that an operator performing a particular service does not need access to corresponding Key Material or CSP.

Cells marked with “n/a” mean that the service is not available for operator in the specified role.

While an operator is in crypto-officer role, his or her access to services listed above may be restricted according to the privileges of his or her key file and described in Section 4.2.