SecureDoc® Disk Encryption Cryptographic Engine

FIPS 140-2 Non-Proprietary Security Policy

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

Module Version:	4.7
Document Version:	1.9
Revision date:	September 9, 2010
Evaluation:	FIPS 140-2 level 1

Table of Contents

1 Introduction

1.1 Purpose

This document describes the non-proprietary FIPS 140-2 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 140-2 Level 1 requirements. It also describes the storage of cryptographic data within the engine and how the Cryptographic Module 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 140-2 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.

The target levels of validation by components are specified below.

Section	*Security Requirements Section*	*Level*
1	Cryptographic Module Specification	1
2	Cryptographic Module Ports and Interfaces	1
3	Roles and Services and Authentication	3
4	Finite State Machine Model	1
5	Physical Security	N/A
6	Operational Environment	1
7	Cryptographic Key Management	1
8	EMI/EMC	3
9	Self-Tests	1
10	Design Assurance	3
11	Mitigation of Other Attacks	N/A
	Overall Level of Certification	1

2 Product Overview

2.1 Module Installation

The software comprising the cryptographic module is provided as an installation package in the format of the corresponding OS. The installation process includes several steps.

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

2. On the next stage the operator works with the installation wizard that does the following:

    a. Generates user keys and creates crypto-officer's key file.

    b. Installs the Boot Logon component for pre-boot authentication.

    c. Encrypts the hard disk.

    d. Prepares Data Recovery Media (optional).

2.2 Cryptographic Engine

The SecureDoc Cryptographic Engine provides cryptographic and key management services for all SecureDoc® products. It is based on the widely adopted PKCS-11 Cryptoki standard.

Key database security is ensured by encrypting all the sensitive contents including all cryptographic keys and protecting them with a "User PIN" entered by the operator from the keyboard or/and a randomly generated "Strong PIN". The latter is used for multi-factor authentication with cryptographic devices like smartcard, biometrics, etc. or for the purpose of operator's password recovery in the event that the 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 used by applications such as the SecureDoc® Central Database software to control all aspects of key management and usage.

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

3 Cryptographic Module Definition

3.1 Cryptographic Module Boundary and Interface

From the point of view of FIPS 140-2, the cryptographic boundary of the module as a multiple-chip standalone module includes the cryptographic module itself, the Operating System (OS) and the General Purpose Computer (GPC) hardware.

The interface to the module is the physical interface to the GPC including the mouse, keyboard, video monitor, etc. as defined below: The hardware external to the physical boundary is connected either via a designated port or via USB if it is supported by the vendor of this hardware and the OS.

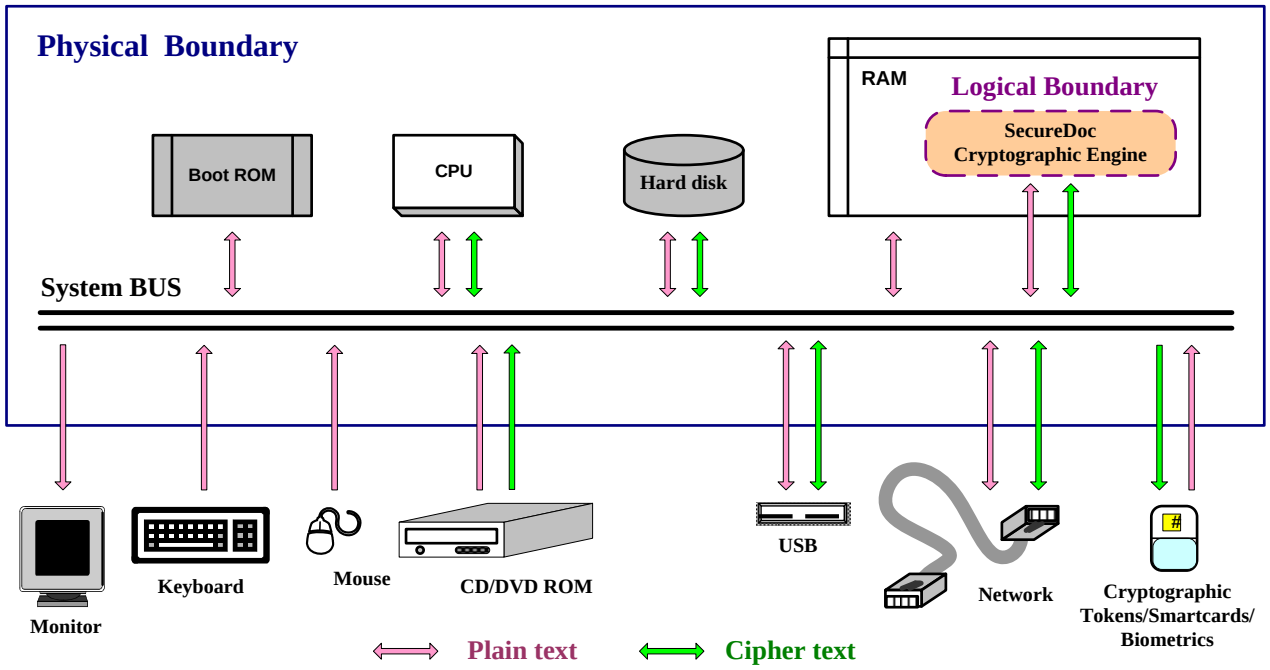

Figure 1 - Cryptographic Module Block-Diagram

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 module	Keyboard port, mouse port, USB, Ethernet port
Data output interface	Output the data been processed by the module	USB port, Ethernet port
Control input interface	Enter the data used to control operation of the module	Mouse, keyboard, and CPU
Status output interface	Show module status and error messages	Video monitor
Power interface	Provides power for module operations	110/220 VAC power interface

Table 1 - Ports and logical interfaces correspondence

3.2 Module Operational Levels

The software comprising cryptographic module operates at three different levels as shown in the picture below. Once the module goes through Power On at hardware level the module 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 cryptographic engine is active and controls critical operations as part of the SecureDoc kernel filter driver. On the next step the operator has to login to OS session to enter the user level in which SecureDoc (and other) applications work. At this level all media access operations as well as SecureDoc configuration management go through kernel filter driver. Other SecureDoc application may run the cryptographic engine in user mode to perform cryptographic operations in memory.

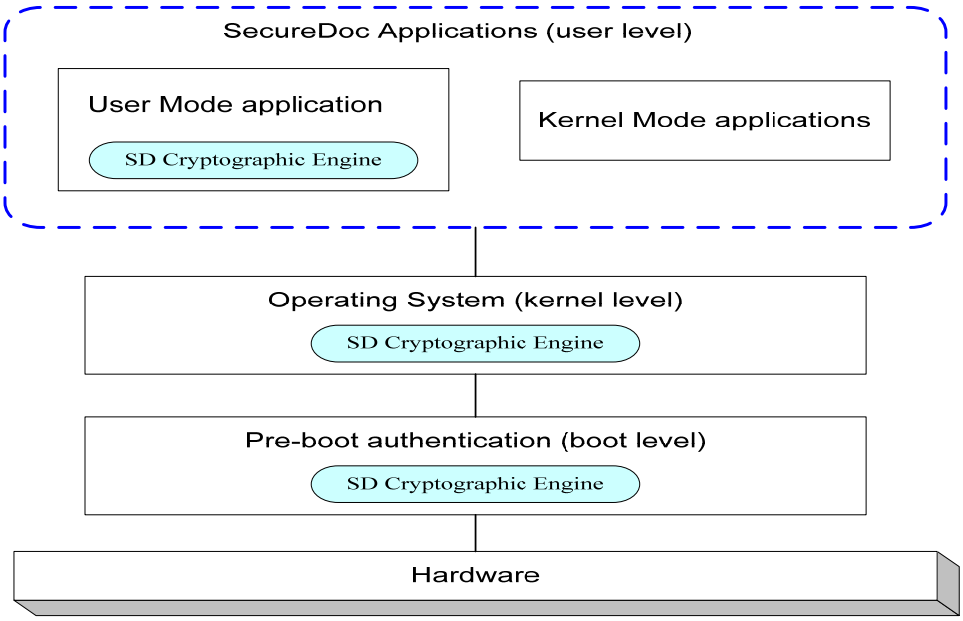

Figure 2 - Cryptographic module operational levels

3.3 Implementation

The Cryptographic Engine is designed to integrate closely with the operating system taking full advantage of the operating system security. It is available in various instances to run either in user mode, for general purpose applications; in kernel mode for use by low-level applications such as the SecureDoc® Disk Encryptor; or in real mode at pre-boot.

The user mode instance is distributed as a dynamic or shared library. The kernel mode instance is installed as an OS driver. The real mode instance is a binary file loaded by the boot code.

From the programmatic point of view, the application interface to the Cryptographic Module is using the standard C-language API. This interface corresponds to the PKCS-11 Cryptoki standard with extensions introduced to meet the unique requirements of the SecureDoc product and FIPS 140-2.

The module design is such that only one function call can be processed at any time in a given application thread. Any other function requests will not be executed until the current function processing is completed.

The API ensures that each service call results in a return code that unambiguously 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.

3.4 Operational Environment

For the purpose of FIPS 140-2 level 1 evaluation, the SecureDoc® Cryptographic Module is classified as a multiple-chip standalone module. The module has been tested on GPCs running the MacOS X 10.5 and Windows Vista.

Other supported operating systems are:

- Windows 2000 Professional / Server / Advanced Server
- Windows XP / Server 2003 / Server 2008
- Windows 7
- MacOS X 10.4, 10.6
- Red Hat Enterprise Linux 5.3, 5.4, 5.5
- Fedora 10, 11, 12, 13
- SuSe Linux Enterprise 11
- OpenSuSe 10.2, 11.0, 11.1
- Debian 5.0
- Ubuntu 9.04, 10.4

3.5 Physical Security

SecureDoc Cryptographic Engine is implemented as a software component and thus the FIPS 140-2 physical security requirements are not applicable.

3.6 Mitigation of Other Attacks

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

3.7 FIPS Approved Mode of Operation

The SecureDoc Cryptographic Engine implements FIPS 140-2 approved cryptographic algorithms listed in the tables below:

Algorithm	Cryptographic Function	Modes / Mechanisms	Output Block (bytes)	Key Size (bits)	Certificate #
AES	Encipherment	ECB, CBC	16	256	1046, 1047
SHA	Hashing	SHA-1,256, 384, 512	20, 32, 48, 64		996
HMAC	Message authentication	SHA-1, 256, 384, 512	20, 32, 48, 64	256	588
PRNG	Random number generation	ANSI X9.31 AES	16	256	597

Table 2 - Algorithms in Approved Mode of Operation

3.8 Self-Tests

A complete set of self-checks is 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-initialized 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.

The table below details the self-tests performed by the cryptographic module:

Test	Type	Actions performed
Cryptographic Algorithms Test	Power-Up	Performed automatically when the module is initialized and on demand via the self-test service. Executes Known Answer Tests for all employed algorithms and available (AES, PRNG, SHA-1, SHA-256, SHA-384, SHA-512, HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA512)
Software Integrity Test	Power-Up	Performed automatically when the module is initialized Verifies that the cryptographic module has not been compromised by calculating HMAC-SHA-256 checksum.
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 module

4 Cryptographic Key Management

4.1 Encryption Keys

To encrypt users objects (hard drive, floppies, USB media, files and folders) SecureDoc creates through the cryptographic engine 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 cryptographic engine and are never exported outside the cryptographic boundary.

4.2 User Keys

User Keys are used by 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.) 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. 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 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 proprietary (fixed) key known by the SecureDoc® Cryptographic Engine. The private data in the key file are 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 that is derived from the PIN provided by the operator and serving for the purpose of authentication.

4.4 Key Generation

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.

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 cryptographic engine exist as separate PKCS-11 cryptographic objects or as a part of larger ones. Any object including (containing) cryptographic keys can be zeroized 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 cryptographic engine key files.

5 Operator Roles

5.1 User privileges

SecureDoc controls access to the various services in the module through the Authorization 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 which restrict 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 cryptographic 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	"User PIN" or "Strong PIN"
Crypto-Officer	Identity-based	"User PIN" or "Strong PIN"

Table 5 - Roles implemented by the module

Authentication data are either entered by the operator from the keyboard or recovered from a secure storage by using an attached cryptographic device. The authentication data entered by the operator are obscured during the entering via replacement with the star-symbol ('*').

Authentication data are generated outside of the SecureDoc® Cryptographic Module and must follow certain constraints. The password rules imposed by SecureDoc on "User PIN" requires at least **8** (eight) alpha/digit/special characters with at least one of each kind as well as both upper and lower case letters present. In case of "Strong PIN" SecureDoc utilizes 256-bit randomly generated values.

Strength of the authentication mechanism employed is demonstrated below:

Authentication data	Strength of Mechanism
"User PIN"	The number of all passwords matching the rules specified above (344,934,890,528) is big enough to guarantee the required 1 in 1,000,000 probability of guessing the "User PIN". The module also blocks the input interface after 3-5 sequentially failed login attempts forcing the operator to reset the module. In case of a GPC it means full reboot that allows in average 10-12 login attempts per minute. For the length of "User PIN" specified above this limitation gives a possibility less than 1 to 100,000 to guess the "User PIN" within one minute.
"Strong PIN"	The key space of a randomly generated 256-bit "Strong PIN" is apparently bigger than calculated for "User PIN" because the generated values are both longer than and also not restricted by the limited set of symbols available on the keyboard. Thus the requirements for probability of guessing the "Strong PIN" are met with this type of authentication data as well.

Table 6 - Strength of authentication mechanisms implemented by the module

Once logged in to the key file the operator is assigned a session inside the SecureDoc® Cryptographic Module. This session is associated with the keys, AV, etc. from the key file which become available for the SecureDoc® Cryptographic Module after a successful authentication. Sessions are assigned individually to authenticated operators and user interface utilized by the operator then refers to this session while accessing the cryptographic services provided. When the operator logs out, the session is destroyed and they (the operator) will have to re-authenticate to access the module services.

6 Cryptographic Module Services

6.1 Services implemented

The table below contains a full list of services implemented in the cryptographic module all of which can only be performed by an authenticated role.

Service	Category	Actions performed
Initialize / Self Test	Administrative	Initializes the 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 module (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-256,384,512 mechanisms

Table 7 - Services provided by the module

6.2 Administrative Services

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

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 which a service operates with is always wrapped with another key when exported from the cryptographic module. 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. “Message Authentication 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 organization. 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 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
Initialize / 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		
Zeroize Key	Crypto-Officer	Write	Write	Write	
	User	Write	Write	Write	
Import Key	Crypto-Officer		Read/Write		
Export Key	Crypto-Officer		Read		
Archive Key	Crypto-Officer		Read/Write		
Encrypt	Crypto-Officer	Read	Read	Read	
	User	Read	Read	Read	
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 module

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 his or her access to services listed above may be restricted according the privileges of his or her key file and described in Section 5.1.