nShield Security Policy

nCIPHER™

nShield F3 4000, nShield F3 2000, nShield F3 2000 for NetHSM, nShield F3 500, nShield F3 500 for NetHSM in FIPS 140-2 level 2 mode

Version: **2.5.4**

Date: **25 January 2010**

Patents

UK Patent GB9714757.3. Corresponding patents/applications in USA, Canada, South Africa, Japan and International Patent Application PCT/GB98/00142.

Contents

Introduction

nShield tamper resistant Hardware Security Modules are multi-tasking hardware modules that are optimized for performing modular arithmetic on very large integers. The modules also offer a complete set of key management protocols.

The nShield Hardware Security Modules are defined as multi-chip embedded cryptographic modules as defined by FIPS PUB 140-2.

Unit ID	Model Number	RTC NVRAM	SEE	Potting	EMC	Crypto Accelerator	Overall level
nShield F3 4000	nC4033P-4K0	Yes	Optional	Yes	B	Broadcom 5821	2
nShield F3 2000	nC4033P-2K0	Yes	Optional	Yes	B	Broadcom 5821	2
nShield F3 500	nC4133P-500	Yes	Optional	Yes	B	Broadcom 5821	2
nShield F3 2000 for NetHSM	nC4033P-2K0N	Yes	Optional	Yes	B	Broadcom 5821	2
nShield F3 500 for NetHSM	nC4133P-500N	Yes	Optional	Yes	B	Broadcom 5821	2

The units are identical in operation and only vary in the processing speed and the support software supplied.

All modules are now supplied at build standard "N" to indicate that they meet the latest EU regulations regarding ROHS.

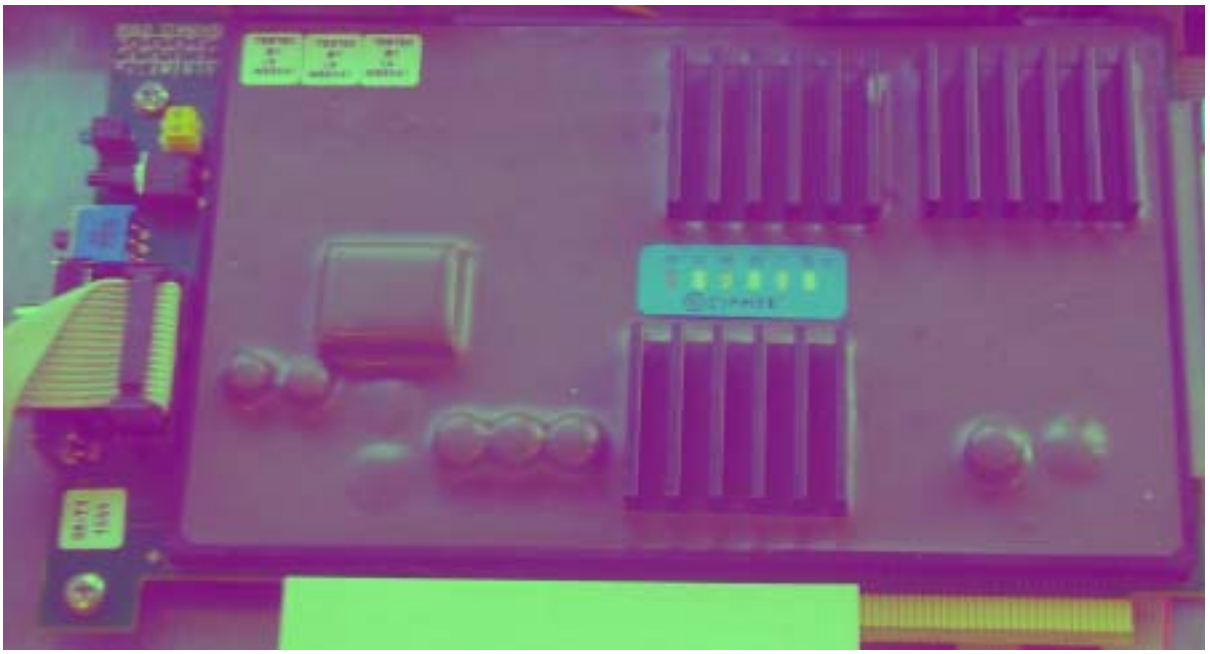
This release introduces the nShield F3 500. This model replaces the nShield F3 500 which cannot be produced due to the EU Reduction of Hazardous Substances directive. To determine which module you have examine the PCI interface. If the module has a 64-bit interface it is covered by this certificate.

The nShield F3 2000 is fitted inside the nCipher NetHSM 2000 and provides all the security functions for this appliance.

The nShield F3 500 is fitted inside the nCipher NetHSM 500.

The NetHSM is shown below.

The following figure shows the module mounted inside the NetHSM



The nShield F3 6000e is fitted in the Thales nShield Connect, shown below:

nCipher also supply modules to third party OEM vendors for use in a range of security products.

The module runs firmware provided by nCipher. There is the facility for the administrator to upgrade this firmware. In order to determine that the module is running the correct version of firmware they should use the **NewEnquiry** service which reports the version of firmware currently loaded.

The validated firmware versions is 2.38.7.

The module can be initialised to comply with the requirements for Roles and Services at either level 2 or level 3

- When initialized in level 2 mode the firmware version is 2.38.7-2 (level 2 mode) and the level 2 certificate applies.
- When initialized in level 3 mode the firmware version is 2.38.7-3 (level 3 mode) and the level 3 certificate applies.

The initialization parameters are reported by the **NewEnquiry** and **SignModuleState** services. An operator can determine which mode the module is operating in using the KeySafe GUI or the command line utilities supplied with the module, or their own code - these operate outside the security boundary.

The modules must be accessed by a custom written application. Full documentation for the nCore API can be downloaded from the nCipher web site: http://www.nCipher.com.

nShield modules have on-board non-volatile memory. There are services that enable memory to be allocated as files. Files have Access Control Lists that determine what operations can be performed on their contents. nShield modules have on-board Real-time clock.

nShield modules include a technology called, the Secure Execution Environment (SEE). This enables operators to load a SEE machine. A SEE machine is operator written code that implements a specific Software Interrupt interface. This enables operators to implement non-cryptographic code in a protected memory space on the module that is outside the logical security boundary.

SEE code is executed in a protected environment. Whenever the SEE machine is running the nCore kernel is locked. Whenever the nCore kernel is active the SEE machine is locked. The SEE machine is excluded from the requirements of FIPS PUB 140-2. While the SEE machine is active the module is running in a non-FIPS mode.

The SEE machine has no direct access to objects stored on the module. In order to use cryptographic functions it must submit a job to the nCore kernel using the nCore API. The testing shows that the interface between the nCore kernel and the SEE machine is secure and that a malicious SEE machine cannot gain access to objects protected by the nCore kernel.

Before a operator can send commands to the SEE machine they must create an instance called a **SEE World**. A **SEE World** is treated as a separate operator by the module and must present the correct authentication before they can submit commands to the nCore kernel.

nShield modules are supplied with the SEE functions disabled. In order to use these functions the customer must purchase a feature-enable certificate enabling the functions for a specific module. The SEE feature is export controlled and therefore is not available in some territories.

All payShield modules are supplied with a SEE licence that enables them to run the nCipher Secure payments Processing library which supports a number of payments specific algorithms and functions which provide support for 3-D Secure, EMV (Europay, MasterCard, Visa) and PIN processing.

The module can be connected to a computer running one of the following operating systems:

- Windows XP and Vista
- Solaris
- HP-UX
- AIX
- Linux x86
- FreeBSD x86

Solaris 2.10, Windows Vista and Linux were used to test the module for this validation.

The following table lists the logical interfaces to the module and how they map to physical interfaces.

Logical Interface	Physical Interface
Data In	PCI bus, Serial Interface, 16-way header
Data Out	PCI bus, Serial Interface, 16-way header
Control In	PCI bus, Temperature Sensor, PSU Monitor, Reset Switch, Mode Switch, 16-way header
Status Out	PCI bus, 16-way header,LED
Power	PCI bus

The following components are excluded from FIPS 140-2 validation:

The nShield module offers protection from over and under voltage and over and under temperature.

The module is designed to operate in a PCI bus drawing power from the PCI 5V rail.

If the power on this rail fall below about 4.5V the PSU monitor shuts the module down, clearing all loaded keys. The PSU monitor will not allow the module to restart until sufficient voltage is applied, going through the standard power on procedure, including clearing all memory and performing all self tests.

If the voltage is increased to around 8V the power supply will continue to perform normally: drawing proportionally less current. Above 8.5V the power supply's over voltage protection device will trigger and all input power will be shorted to earth. This protect the module's cryptographic circuitry from damage. However, as the power supply is potted, it cannot be replaced and the module is therefore irreversibly destroyed.

The module has an on-board temperature sensor mounted close to the processors. If this sensor reaches seventy degrees centigrade or drops to zero degrees centigrade the module is put into its error state. The module turns off all communication on external buses and the Status LED flashes the Morse pattern SOS T (... --- ... -). The module can only be reset from an error state by turning the power supply off and back on - which for a PCI module means turning off power to the host computer. The module performs all normal power on procedures including clearing all memory and performing all self tests.

The module defines the following roles:

Unauthorised

All connections are initially unauthorized. If the module is initialized in level 3 mode, an unauthorized operator is restricted to status commands, and commands required to complete authorization protocol.

User

An operator enters the user role by providing the required authority to carry out a service. The exact accreditation required to preform each service is listed in the table of services.

In order to perform an operation on a stored key, the operator must first load the key blob. If the key blob is protected by a logical token, the operator must first load the logical token by loading shares from smart cards.

If the module is initialized in level 3 mode, the user role requires a certificate from the security officer to import or generate a new key. This certificate is linked to a token protected key.

Once an operator in the user role has loaded a key they can then use this key to perform cryptographic operations as defined by the Access Control List (ACL) stored with the key.

Each key blob contains an ACL that determines what services can be performed on that key. This ACL can require a certificate from a Security Officer authorizing the action. Some actions including writing tokens always require a certificate.

nCipher Security Officer

The nCipher Security Officer (NSO) is responsible for overall security of the module.

The nCipher Security Officer is identified by a key pair, referred to as K_{NSO}. The hash of the public half of this key is stored when the unit is initialized. Any operation involving a module key or writing a token requires a certificate signed by K_{NSO}.

The nCipher Security Officer is responsible for creating the authentication tokens (smart cards) for each operator and ensuring that these tokens are physically handed to the correct person.

An operator assumes the role of NSO by loading the private half of K_{NSO} and presenting the **KeyID** for this key to authorize a command.

Junior Security Officer

Where the nCipher Security Officer want to delegate responsibility for authorizing an action they can create a key pair and give this to their delegate who becomes a Junior Security Officer (JSO). An ACL can then refer to this key, and the JSO is then empowered to sign the certificate authorizing the action. The JSO's keys should be stored on a key blob protected by a token that is not used for any other purpose.

In order to assume the role of JSO, the operator loads the JSO key and presents the **KeyID** of this key, and if required the certificate signed by K_{NSO} that delegates authority to the key, to authorize a command.

A JSO can delegate portions of their authority to a new operator in the same way. The new operator will be a JSO if they have authority they can delegate, otherwise they will assume the user role.

For more information on each of these services refer to the nCipher Developer's Guide and nCipher Developer's Reference.

The following services provide authentication or cryptographic functionality. The functions available depend on whether the operator is in the unauthenticated role, the user or junior security officer (JSO) roles, or the nCipher Security Officer (NSO) role. For each operation it lists the supported algorithms. Algorithms in square brackets are not under the operator's control. Algorithms used in optional portions of a service are listed in italics.

Note *Algorithms marked with an asterisk are not approved by NIST. If the module is initialised in its level 3 mode, these algorithms are disabled. If module is initialized in level 2 mode, the algorithms are available. However, if you choose use them, the module is not operating in FIPS approved mode.*

Key Access	Description
Create	Creates a in-memory object, but does not reveal value.
Erase	Erases the object from memory, smart card or non-volatile memory without revealing value
Export	Discloses a value, but does not allow value to be changed.
Report	Returns status information
Set	Changes a CSP to a given value
Use	Performs an operation with an existing CSP - without revealing or changing the CSP

Command / Service	Role			Description	Key/CSP access	Key types
	Unauth	JSO / User	NSO			
Bignum Operation	Yes	Yes	Yes	Performs simple mathematical operations.	No access to keys or CSPs	
Change Share PIN	No	pass phrase	pass phrase	Updates the pass phrase used to encrypt a token share. The pass phrase supplied by the operator is not used directly, it is first hashed and then combined with the module key. To achieve this the command decrypts the existing share using the old share key derived from old pass phrase, module key and smart card identity. It then derives a new share key based on new pass phrase, module key and smart card identity, erases old share from smart card and writes a new share encrypted under the new share key.	*Sets* the pass phrase for a share, *uses* module key, *uses* share key, *uses* module key, *creates* share key, *uses* new share key, *exports* encrypted share, *erases* old share	[SHA-1 and AES or Triple DES]
Channel Open	No	handle, ACL	handle, ACL	Opens a communication channel which can be used for bulk encryption or decryption. Channels using DES* or Triple DES in CBC mode use the Broadcom 5821 to perform the encryption.	*Uses* a key object	AES, DES*, Triple DES, Arc Four*, Aria*, Camellia*, SEED*,

Channel Update	No	handle	handle	Performs encryption / decryption on a previously opened channel. The operation and key are specified in **ChannelOpen**.	*Uses* a key object	AES, DES*, Triple DES, Arc Four*, Aria*, Camellia*, SEED*,
CheckUserACL	No	handle	handle	Determines whether the ACL associated with a key object allows a specific operator defined action.	*Uses* a key object	
Clear Unit	Yes	Yes	Yes	Zeroises all loaded keys, tokens and shares. Clear Unit does not erase long term keys, such as module keys.	*Zeroizes* objects.	All
Create Buffer	No	cert [handle]	cert [handle]	Allocates an area of memory to load data. If the data is encrypted, this service specifies the encryption key and IV used. This service is feature enabled. The decrypt operation is performed by **LoadBuffer**	*Uses* a key object	AES, DES*, Triple DES, Arc Four*, Aria*, Camellia*, SEED*
Create SEE World	No	handle cert	handle cert	Creates a **SEE World**, passing in the initialization data stored in a buffer. This command checks the DSA signatures on the buffer using the public key provided. It also specifies whether debugging is allowed or not. Enabling debugging requires a certificate from the nCipher Security Officer.	No access to keys or CSPs	
Decrypt	No	handle, ACL	handle, ACL	Decrypts a cipher text with a stored key returning the plain text.	Uses a key object	AES, DES*, Triple DES, Arc Four*, Aria*, Camellia*, SEED*, Diffie-Hellman, ECDH, RSA*, ElGamal*, KCDSA*