nShield™ Family of Hardware Security Modules Firmware Version 2.33.60

nCIPHER™

Security Target

Version 1.9

17 March 2009

nCipher Thales
Jupiter House
Station Road
Cambridge, UK
CB1 2JD

http://www.ncipher.com

Prepared By:
Common Criteria Consulting LLC
15804 Laughlin Lane
Silver Spring, MD 20906

http://www.consulting-cc.com

Prepared For:
nCipher Corporation Ltd.
Jupiter House
Station Road
Cambridge, UK
CB1 2JD

http://www.ncipher.com

This document provides the basis for an evaluation of a specific Target of Evaluation (TOE), the nCipher nShield Family of Hardware Security Modules (HSMs) Firmware Version 2.33.60. This Security Target (ST) defines a set of assumptions about the aspects of the environment, a list of threats that the product intends to counter, a set of security objectives, a set of security requirements and the IT security functions provided by the TOE which meet the set of requirements.

Version	Date	Description
1.0	April 29, 2007	Initial version
1.1	April 30, 2007	Incorporated comments from nCipher
1.2	June 13, 2007	Addressed OR 01
1.3	June 18, 2007	Added user guide titles
1.4	July 2, 2007	nCipher Marketing Comments
1.4a	July 5, 2007	nCipher style sheet
1.5	July 9,2007	Updated algorithm certificates
1.6	August 22, 2008	Addressed CB OR, Updated firmware version, CMVP certificate number
1.7	September 18, 2008	Corrected incorrect firmware version
1.8	October 6, 2008	Addressed CB OR
1.9	March 17, 2009	Algorithm Certificates

Table of Contents

List of Figures

List of Tables

AES ..... Advanced Encryption Standard
ANSI ..... American National Standards Institute
API ..... Application Program Interface
CAVP ..... Cryptographic Algorithm Validation Program
CBC ..... Cipher-Block Chaining
CC ..... Common Criteria
CM ..... Configuration management
CMVP ..... Cryptographic Module Validation Program
DES ..... Data Encryption Standard
DSA ..... Digital Signature Algorithm
EAL4 ..... Evaluation Assurance Level 4
ECB ..... Electronic CodeBook
ECDSA ..... Elliptic Curve Digital Signature Algorithm
FIPS ..... Federal Information Processing Standard
HSM ..... Hardware Security Module
IT ..... Information Technology
MAC ..... Message Authentication Code
NIAP ..... National Information Assurance Partnership
NIST ..... National Institute of Standards and Technology
PCB ..... Printed Circuit Board
PCI ..... Peripheral Component Interconnect
PKI ..... Public Key Infrastructure
PP ..... Protection Profile
PRNG ..... Pseudo-Random Number Generator
RSA ..... Rivest, Shamir and Adleman
SF ..... Security Function
SFP ..... Security Function Policy
SHA ..... Secure Hash Algorithm
SHS ..... Secure Hash Standard
SOF ..... Strength of Function
ST ..... Security Target
TOE ..... Target of Evaluation
TSC ..... TSF Scope of Control
TSF ..... TOE Security Function
TSFI ..... TSF Interface
TSP ..... TOE Security Policy

This Security Target (ST) describes the objectives, requirements and rationale for the nCipher nShield Family of Hardware Security Modules (HSMs) Firmware Version 2.33.60. The language used in this Security Target is consistent with the *Common Criteria for Information Technology Security Evaluation, Version 2.3*, the ISO/IEC JTC 1/SC27, *Guide for the Production of PPs and STs, Version 0.9* and all international interpretations through May 31, 2007. As such, the spelling of terms is presented using the internationally accepted English.

1.1 Security Target Reference

nCipher nShield Family of Hardware Security Modules Firmware Version 2.33.60 Security Target, Version 1.9 17 March, 2009.

1.2 TOE Reference

nCipher nShield Family of Hardware Security Modules (HSMs) Firmware Version 2.33.60

Hereafter the TOE is referred to as the nShield HSM.

1.3 Evaluation Assurance Level

Assurance claims conform to EAL4 (Evaluation Assurance Level 4) from the *Common Criteria for Information Technology Security Evaluation, Version 2.3* augmented by ALC_FLR.1 (Basic flaw remediation).

1.4 Keywords

Hardware Security Module, Key Management, Secure Key Management

1.5 TOE Overview

This Security Target defines the requirements for the nShield HSM. The TOE is the firmware executing on the nShield family of HSMs, which protect keys within a commercial server platform in a highly secure, tamper-resistant hardware environment enabling them to be effectively managed and safely stored. nShield HSMs have received a FIPS 140-2 security validation at level 2 and level 3.

1.5.1 Security Target Organisation

Chapter 1 of this ST provides introductory and identifying information for the TOE.

Chapter 2 describes the TOE and provides some guidance on its use.

Chapter 3 provides a security environment description in terms of assumptions, threats and organisational security policies.

Chapter 4 identifies the security objectives of the TOE and of the Information Technology (IT) environment.

Chapter 5 provides the TOE security and functional requirements, as well as requirements on the IT environment.

Chapter 6 is the TOE Summary Specification, a description of the functions provided by the TOE to satisfy the security functional and assurance requirements.

Chapter 7 identifies claims of conformance to a registered Protection Profile (PP).

Chapter 8 provides a rationale for the security objectives, requirements, TOE summary specification and PP claims.

1.6 Common Criteria Conformance

The nShield HSM-is compliant with the Common Criteria (CC) Version 2.3, functional requirements (Part 2) conformant and assurance requirements (Part 3) augmented for EAL4. The augmentation is ALC_FLR.1 (Basic flaw remediation).

1.7 Protection Profile Conformance

The nShield HSM does not claim conformance to any registered Protection Profile.

1.8 Conventions

The CC defines operations on security requirements. The font conventions listed below state the conventions used in this ST to identify the operations.

Assignment: indicated in underlined text

Selection: indicated in italics

Assignments within selections: indicated in italics and underlined text

Refinement: indicated with bold text

Iterations of security functional requirements may be included. If so, iterations are specified at the component level and all elements of the component are repeated. Iterations are identified by numbers in parentheses following the component or element (e.g., FAU_ARP.1(1)).

This section provides the context for the TOE evaluation by identifying the product type and describing the evaluated configuration.

2.1 Product Description

The nShield™ range of HSMs provide physical and logical protection for the cryptographic keys used within industry standard computing platforms. By providing a highly secure, tamper-resistant hardware environment sensitive keys and data are easily protected from a range of internal and external threats. nShield has been certified to FIPS 140-2 at level 2 and level 3, enabling organizations to comply with regulatory, industry and government best practice with existing applications.

In addition to providing a platform for implementing best practice security, nShield also boasts a choice of flexible key management solutions and hardware acceleration for a variety of applications that rely on cryptography – such as digital signature, data encryption and digital rights management . By storing, using and managing cryptographic keys entirely within nShield's highly secure hardware environment, organizations can protect keys, applications and data from a range of network, user and administrative threats.

The nShield product offers a range of performance options and security validations to best fit the individual business requirements. All members of the product family provide the security functionality described in this Security Target.

All nShield products can utilize nCipher's Security World™ key management system and offer a range of cryptographic APIs to integrate with applications on the server. nShield products can also be used as hardware endpoints with with keyAuthority, an enterprise key management system providing the automation and centralized control for organizations with large deployments of cryptographic services.

Cryptographically-secured keys are stored on the server disks thus eliminating any limitation on the number of keys that can be used by the TOE.

When a key is needed by the TOE, the user provides authorisation to use the key by inserting smart cards contain encrypted shares of the key used to encrypt the blob. Once this key is reassembled, the corresponding key blob is retrieved from the server disk and passed into the TOE. The TOE then verifies the integrity of the key blob, and the key itself is then decrypted within the TOE.

When used with keyAuthority, keys are security delivered to the nShield using a mutually authenticated and encrypted key delivery protocol from any of a number of nCipher key Provisioning Servers. This enables organizations to benefit from centrally administered automated key policy enforcement.

2.1.1 Physical Boundary

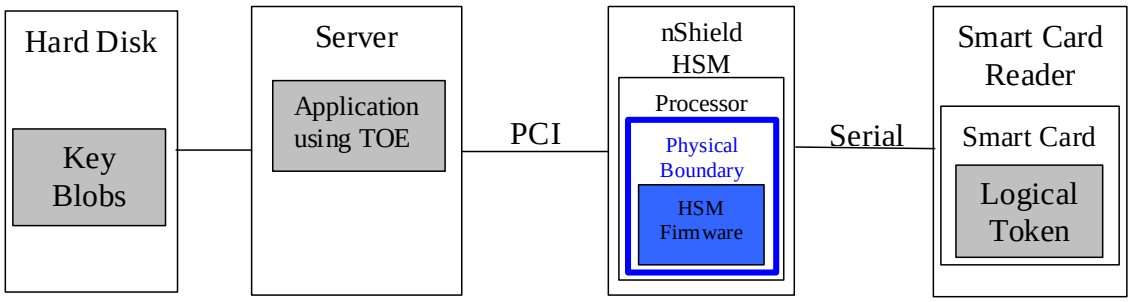
The HSM is a PCI or PCIe module that is integrated with a server. The HSM includes a processor, and the TOE executes on that processor. Protective potting covers the processor on the HSM to prevent any tampering without destroying the PCB. The following diagram shows one of the HSMs.

Figure 1 nShield PCI Module

In addition to the PCI connection to a server, the HSM also supports a connection to a smart card reader. Smart cards may be used to hold logical tokens used to encrypt key blobs. Possession of the correct smart card provides authorization to use the stored key.

The following block diagram illustrates the relationships between the components and delineates the physical boundary of the TOE. Hardware entities are shown with white fill while soft entities (software and files) are shown in grey. The physical boundary of the TOE (shown in blue) consists of the firmware executing on the processor of the HSM.

Figure 2 Physical Boundary



2.1.2 Logical Boundary

The logical boundary of the TOE defines the security functionality provided by the TOE. The nShield HSM Firmware provides the following security functionality.

2.1.3 Key Management

The nShield HSM Firmware provides the ability to generate, archive and recover cryptographic keys. To support that functionality the TOE also provides cryptographic operations including encryption, decryption, hashes, digital signatures, message authentication codes, and random number generation.

2.1.4 Self Protection

The TOE protects itself from bypass and interference and tests the underlying abstract machine on which it operates.

2.1.5 Evaluated Configuration

The evaluated configuration of the TOE consists of the nShield HSM Firmware, version 2.33.60, executing on a single instance of any member of the nShield family of HSMs.

3.1 Introduction

This chapter defines the nature and scope of the security needs to be addressed by the TOE. Specifically this chapter identifies:

A assumptions about the environment,

B threats to the assets and

C organisational security policies.

This chapter identifies assumptions as A.*assumption*, threats as T.*threat* and policies as P.*policy.*

3.2 Assumptions

The specific conditions listed in the following table are assumed to exist in the TOE environment.

Table 1 Assumptions

A.Type	Description
A.ENVIRON	The TOE will be located in an environment that provides uninterruptible power, temperature control and necessary physical security required for reliable operation.
A.INSTALL	The Administrator will install and configure the TOE according to the administrator guidance.
A.NOEVILADMIN	Administrators are non-hostile and follow the administrator guidance when using the TOE. Administration is competent and on-going.
A.PLATFORM	The Administrator will ensure that the platforms used to host the TOE conform to the hardware and software outlined in the administrator guidance.

3.3 Threats

The threats identified in the following table are addressed by the TOE and/or the IT environment.

Table 2 Threats

T.Type	TOE Threats
T.KEY_ COMPROMISE	Keys used to protect sensitive data may be compromised, permitting unauthorized access to the protected data.
T.TSF_COMPROMISE	A user or process may cause, through an unsophisticated attack, TSF data or executable code to be inappropriately accessed (viewed, modified, or deleted).

3.4 Organisational Security Policies

The organizational security policies identified in the following table are addressed by the TOE and/or the IT environment.

Table 3 Organisational Security Policies

P.Type	Organisational Security Policies
P.CRYPTOGRAPHY	Only NIST FIPS 140-2 validated cryptography (methods and implementations) are acceptable for key management (i.e.; generation, access, distribution, destruction, handling, and storage of keys) and cryptographic services (i.e.; encryption, decryption, signature, hashing, key exchange, and random number generation services).

This section identifies the security objectives of the TOE, the TOE's IT environment and the TOE's non-IT environment. The security objectives identify the responsibilities of the TOE, the TOE's IT environment, and the TOE's non-IT environment in meeting the security needs. Objectives of the TOE are identified as *O.objective*. Objectives that apply to the IT environment are designated as *OE.objective*. Objectives that apply to the non-IT environment are designated with an *ON.objective*.

4.1 Security Objectives for the TOE

The TOE must satisfy the following objectives.

Table 4 Security Objectives for the TOE

O.Type	Security Objective
O.CRYPTOGRAPHY	The TOE shall use NIST FIPS 140-2 validated cryptographic services.
O.PROTECT_KEYS	The TOE will protect cryptographic keys from compromise.
O.SELF_PROTECT	The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure.

4.2 Security Objectives for the IT Environment

There are no objectives to be satisfied by the TOE's IT environment.

4.3 Security Objectives for the Non-IT Environment

The TOE's Non-IT environment must satisfy the following objectives.

Table 5 Security Objectives for the Non-IT Environment

O.N.Type	Security Objectives for the Non-IT Environment
ON.ENVIRON	The Administrator will install the TOE in an environment that provides physical security, uninterruptible power, and temperature control required for reliable operation.
ON.INSTALL	The Administrator will install and configure the TOE according to the administrator guidance.
ON.NOEVILADMIN	Administrators are non-hostile and follow the administrator guidance when using the TOE.
ON.PLATFORM	The Administrator will ensure that the platforms used to host the TOE conform to the hardware and software outlined in the administrator guidance.

This section contains the functional requirements that are provided by the TOE. These requirements consist of functional components from Part 2 of the CC.

5.1 TOE Security Functional Requirements

The functional requirements are described in detail in the following subsections. Additionally, these requirements are derived verbatim from Part 2 of the *Common Criteria for Information Technology Security Evaluation, Version 2.3* with the exception of completed operations.

5.1.1 Cryptographic Support (FCS)

FCS_CKM.1 Cryptographic Key Generation

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm as described below and specified cryptographic key sizes as described below that meet the following standards described below:

Table 6 Cryptographic Key Generation

Algorithm	Key Size in Bits	Standards
SHS (SHA-1) (CAVP cert #648)	112, 168, 256	FIPS 180-2
PRNG	Not applicable	FIPS 186-2 Change Notice 1 SHA-1 and FIPS 186-2 RNG General Purpose RNG

FCS_CKM.3 Cryptographic Key Access

FCS_CKM.3.1 The TSF shall perform key archive, key recovery in accordance with a specified cryptographic key access method key blobs that meets the following: FIPS 140-2.

FCS_CKM.4 Cryptographic Key Destruction

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method zeroization that meets the following: FIPS 140-2 (CMVP cert #965, #966, #968, #970, #973 and #977).

FCS_COP.1 Cryptographic Operation

FCS_COP.1.1 The TSF shall perform the operations described below in accordance with a specified cryptographic algorithm multiple algorithms in the modes of operation described below and cryptographic key sizes multiple key sizes described below that meet the following multiple standards described below:

Table 7 Cryptographic Operations

Operation	Algorithm (mode)	Key Size in Bits	Standards
Encryption and decryption	Triple-DES (ECB, CBC) (CAVP cert #570))	112, 168	FIPS 46-3
	AES (CBC, ECB) (CAVP cert #599))	256	FIPS 197
Message authentication coding	HMAC (SHA-1) (CAVP cert #309))	128, 224, 256, 384, 512	FIPS 198
Hashing	SHS (CAVP cert #648)	128, 224, 256, 384, 512	FIPS 180-2
Random Number Generation	PRNG (CAVP cert #340)	Not Applicable	FIPS 186-2 Change Notice 1 SHA-1 and FIPS 186-2 RNG General Purpose RNG
Digital Signature	DSA (CAVP cert #233)	1024, 1536, 2048, 3072, 4096	FIPS 186-2
	ECDSA (CAVP cert #64)	1024, 1536, 2048, 3072, 4096	FIPS 186-2
	RSA (CAVP cert #274)	1024, 1536, 2048, 3072, 4096	ANSI X9.31

5.1.2 Protection of the TSF (FPT)

FPT_AMT.1 Abstract Machine Testing

FPT_AMT.1.1 The TSF shall run a suite of tests *during initial start-up* to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

FPT_RVM.1 Non-Bypassability of the TSP

FPT_RVM.1.1 The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

FPT_SEP.1 TSF Domain Separation

FPT_SEP.1.1 The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

5.2 TOE Security Assurance Requirements

The TOE meets the assurance requirements for EAL4 and is augmented by ALC_FLR.1. These requirements are summarised in the following table.

Table 8 EAL4 Assurance Requirements

Assurance Class	Component ID	Component Title
Configuration Management	ACM_AUT.1	Partial CM automation
	ACM_CAP.4	Generation support and acceptance procedures
	ACM_SCP.2	Problem tracking CM coverage
Delivery and Operation	ADO_DEL.2	Detection of modification
	ADO_IGS.1	Installation, Generation, and Start-Up Procedures
Development	ADV_FSP.2	Fully defined external interfaces
	ADV_HLD.2	Security enforcing high-level design
	ADV_IMP.1	Subset of the implementation of the TSF
	ADV_LLD.1	Descriptive low-level design
	ADV_RCR.1	Informal Correspondence Demonstration
	ADV_SPM.1	Informal TOE security policy model
Guidance Documents	AGD_ADM.1	Administrator Guidance
	AGD_USR.1	User Guidance
Life Cycle Support	ALC_DVS.1	Identification of security measures
	ALC_FLR.1	Basic flaw remediation
	ALC_LCD.1	Developer defined life-cycle model
	ALC_TAT.1	Well-defined development tools
Tests	ATE_COV.2	Analysis of coverage
	ATE_DPT.1	Testing: high-level design
	ATE_FUN.1	Functional Testing
	ATE_IND.2	Independent Testing - Sample

Table 8 EAL4 Assurance Requirements

Assurance Class	Component ID	Component Title
Vulnerability Assessment	AVA_MSU.2	Validation of analysis
	AVA_SOF.1	Strength of TOE Security Function Evaluation
	AVA_VLA.2	Independent vulnerability analysis

5.3 Strength of Function for the TOE

There are no non-cryptographic probabilistic or permutational mechanisms in the TOE. The overall SOF for the TOE is SOF-Basic.

5.4 CC Component Hierarchies and Dependencies

This section of the ST demonstrates that the identified SFRs include the appropriate hierarchy and dependencies. The following table lists the TOE SFRs and the SFRs each are hierarchical to, dependent upon and any necessary rationale.

Table 9 TOE SFR Dependency Rationale

SFR	Hierarchical To	Dependency	Rationale
FCS_CKM.1	No other components.	[FCS_CKM.2 or FCS_COP.1], FCS_CKM.4, FMT_MSA.2	Satisfied Satisfied Not satisfied – all key material (security attributes) is generated by the TOE so acceptance of secure values is not applicable.
FCS_CKM.3	No other components.	[FDP_ITC.1 or FDP_ITC.2, or FCS_CKM.1], FCS_CKM.4, FMT_MSA, .2	Satisfied Satisfied Not satisfied – all key material (security attributes) is generated by the TOE so acceptance of secure values is not applicable.
FCS_CKM.4	No other components.	[FDP_ITC.1 or FDP_ITC.2, or FCS_CKM.1], FMT_MSA.2	Satisfied Not satisfied – all key material (security attributes) is generated by the TOE so acceptance of secure values is not applicable.

Table 9 TOE SFR Dependency Rationale

SFR	Hierarchical To	Dependency	Rationale
FCS_COP.1	No other components.	[FDP_ITC.1 or FDP_ITC.2, or FCS_CKM.1], FCS_CKM.4, FMT_MSA.2	 Satisfied Satisfied Not satisfied – all key material (security attributes) is generated by the TOE so acceptance of secure values is not applicable.
FPT_AMT.1	No other components.	None	n/a
FPT_RVM.1	No other components.	None	n/a
FPT_SEP.1	No other components.	None	n/a

6.1 Security Functions

6.1.1 Key Management

The TOE provides key generation, key archive and key recovery functions. FIPS-validated cryptographic operations are used in these functions, ensuring the keys remain protected at all times. The TOE also makes the cryptographic operations available to users of the TOE, enabling the operations to be performed without exposing the keys in clear text outside the TOE. The operations made available are defined in FCS_COP.1 in chapter 5.

Key generation is performed by calculating a SHA-1 hash of the output of the pseudo-random number generator. Keys are generated for internal use within the TOE as well as for user-controlled sessions. The TOE also supports key fragmentation, which enables keys to be broken into multiple pieces (and stored separately). The fragments must be recombined before the key is again available for use by the TOE.

Keys are archived (or exported) as key blobs, which are cryptographically protected so that the keys can be securely recovered (or imported) later for use by the TOE. Key archive and key recovery are used by the TOE to avoid any limitations on the number of keys it can use. Key blobs may be encrypted with a key always accessible to the module (Module Key) or with a logical token, with the logical token in turn securely stored on a smart card. If the smart card with the appropriate logical token is not presented to the TOE, the key blobs protected by that logical token can't be used by the TOE. The key blobs can be securely stored on the server's hard disk until the key is needed by the TOE, when it is recovered. The key recovery process includes verification that the key has not been modified while stored as a key blob.

Key blobs are created by:

The target key (or fragment) is encrypted using strong (Triple-DES or AES) encryption.

That result is signed with a wrapper key (module key or logical token), to form the key blob.

A Message Authentication Code (MAC) is stored with the key blob, ensuring that tampering is detectable.

Once a key is no longer needed in the TOE, it is zeroized.

The following keys are generated by the TOE:

Table 10 Keys Generated by the TOE

Key Type	Description
Long Term Signing Key	The TOE stores a 160 bit random number in the HSM. This data is combined with a discrete log group stored in the TOE firmware to produce a DSA key. This key is not used to encrypt any other data. It only serves to provide a cryptographic identity for a module that can be included in a PKI certificate chain. This is the only key that is not zeroized when the module is initialized.
Module Signing Key	When the TOE is initialized it automatically generates a DSA key pair that it uses to sign certificates. The private half of this pair is stored internally in the HSM and never released. The public half is revealed in plaintext, or encrypted as a key blob under some other key. This key is only ever used to verify that a certificate was generated by a specified module.
Module Keys	Module keys are AES or Triple DES keys used to protect tokens. The TOE generates the first module key when it is initialized. This module key is an AES key guaranteed never to have been known outside this module. Setting a key as a module key stores the key in the HSM. Module keys can not be exported once they have been assigned as module keys.
Logical Tokens	A logical token is an AES or Triple DES key used to protect key blobs. Logical tokens are associated with module keys. The key type depends on the key type of the module key. When a module key is exported the logical token - the Triple DES key plus the token parameters - is first encrypted with a module key. Then the encrypted token is split into shares using the Shamir Threshold Sharing algorithm, even if the total number of shares is one. Each share is then encrypted using a share key and written to a physical token (smart card) or software token. Logical tokens can be shared between one or more physical tokens.
Share Keys	A share key is used to protect a logical token share when they are written to a smart card or software token that is used for authentication. The share key is created by creating a message comprised of an nCipher secret prefix, Module key, Share number, smart card unique id and an optional 20 bytes supplied by the user (expected to be the SHA-1 hash of a pass phrase entered into the application), and using this as the input to the approved pRNG function to form a unique key used to encrypt the share - this is either an AES or Triple DES key depending on the key type of the logical token which is itself determined by the key type of the module key. This key is not stored on the module. It is recalculated every time share is loaded. The share data includes a MAC; if the MAC does not verify correctly the share is rejected.
Administrator Keys	The administrator keys must be set as part of the initialisation process. This is a public / private key pair that the administrator uses to sign certificates to authorize key management and other secure operations. The SHA-1 hash of the public half of this key pair is stored in the HSM. The public half of this key is included as plain text in certificates.

6.1.2 Self Protection

The TOE provides for self protection and non-bypassability of functions within the TOE's scope of control (TSC). By maintaining and controlling user interactions, the TOE ensures that no security functions within the TSC are bypassed. Since the TOE is a stand-alone system with no support for general purpose users to introduce code into the TOE, it inherently maintains a separate domain for its own execution that prevents the TOE from being interfered with or tampered with. The TOE supports multiple users; each interaction is processed separately so that separate domains are maintained for each of the user sessions. No mechanism is available to modify the TOE firmware via the TSFIs. On every power up, a set of tests are executed against the HSM hardware to ensure the underlying abstract machine is operating properly.

6.2 Assurance Measures

The following table provides a high-level description of the documents that satisfy each of the security assurance requirements.

Table 11 EAL4 Assurance Measures

Assurance Class	Component ID	Documentation Satisfying Component
Configuration Management	ACM_AUT.1	nShield V11 Configuration Management Plan nCipher performs configuration management on configuration items of the TOE. Automated processes are utilized to ensure that only authorized changes are made.
	ACM_CAP.4	nShield V11 Configuration Management Plan nCipher performs configuration management on configuration items of the TOE. Configuration management is performed on the TOE, the implementation representation of the TOE, security flaws pertinent to the TOE, and all documentation submitted as evidence for the CC evaluation. The configuration items are uniquely identified and each release of the TOE has a unique reference. The processes include an acceptance plan that describes the procedures used to accept modified or newly created configuration items as part of the TOE.
	ACM_SCP.2	nShield V11 Configuration Item List The Configuration Items include the TOE, the implementation representation of the TOE, security flaws pertinent to the TOE, and all documentation submitted as evidence for the CC evaluation.

Table 11 EAL4 Assurance Measures

Assurance Class	Component ID	Documentation Satisfying Component
Delivery and Operation	ADO_DEL.2	nShield V11 Delivery Processes and Procedures nCipher documents the delivery procedure for the TOE to include how components of the TOE are delivered to the user. The delivery procedure detail how the end-user may determine if they have the TOE and if the integrity of the TOE has been maintained.
	ADO_IGS.1	Hardware Installation Guide nCipher documents the installation, generation, and startup procedures so that the users of the TOE can put the components of the TOE in the evaluated configuration.

Table 11 EAL4 Assurance Measures

Assurance Class	Component ID	Documentation Satisfying Component
Development	ADV_FSP.2	nShield V11 Functional Specification nCore Developers Refernce The external TSFIs are fully documented along with the description of the security functions and a correspondence between the interfaces and the security functions.
	ADV_HLD.2	nShield V11 High Level Design The subsystems of the TOE are documented in the High Level Design. The TOE identifies the TSP-enforcing subsystems and provides interface details for those subsystems.
	ADV_IMP.1	Source Code The source code for the TOE provides the implementation representation.
	ADV_LLD.1	nShield V11 Low Level Design The Low Level Design provides a description of the internal workings of the TSF in terms of modules and their interrelationships and dependencies. For each module of the TSF, the Low Level Design describes its purpose, function, interfaces, dependencies, and the implementation of any TSP-enforcing functions.
	ADV_RCR.1	nShield V11 Correspondence Mapping The Correspondence Mapping between the various TSF representations addresses the correct and complete instantiation of the requirements starting with the ST and continuing to the implementation representation.
	ADV_SPM.1	nShield V11 Security Policy Model The Security Policy Model describes the correspondence between the functional specification, the security policy model, and the policies of the TSP.

Table 11 EAL4 Assurance Measures

Assurance Class	Component ID	Documentation Satisfying Component
Guidance Documents	AGD_ADM.1	nShield User Guide The administrative guidance is detailed to provide descriptions of how administrative users of the TOE can securely administer the TOE using those functions and interfaces detailed in the guidance.
	AGD_USR.1	nShield User Guide The user guidance provides the information necessary to users to safely and responsibly use the security functionality of the TOE.
Life Cycle Support	ALC_DVS.1	nShield V11 Life Cycle nCipher implements development security mechanisms during the development and maintenance of the TOE.
	ALC_FLR.1	nShield V11 Flaw Remediation nCipher implements process and procedures to collect information regarding security flaws in the TOE, identify corrections for flaws, and communicate flaw information to customers.
	ALC_LCD.1	nShield V11 Life Cycle The Life Cycle Document defines the procedures, tools and techniques used to develop and maintain the TOE.
	ALC_TAT.1	nShield V11 Life Cycle The Life Cycle Document identifies all tools used in the development of the TOE. The documentation for those tools defines all statements and options used.

Table 11 EAL4 Assurance Measures

Assurance Class	Component ID	Documentation Satisfying Component
Tests	ATE_COV.2	nShield V11 Test Plan nShield V11 Test Coverage nCipher demonstrates the external interfaces tested during functional testing using a coverage analysis. The analysis includes information describing how the interfaces are tested.
	ATE_DPT.1	nShield V11 Test Plan nCipher demonstrates the internal subsystem interfaces tested during functional testing using a depth analysis. The analysis includes information describing how the interfaces are tested.
	ATE_FUN.1	nShield V11 Test Plan, nShield V11 Test Procedures, nShield V11 Test Results nCipher functional testing documentation contains a test plan, a description of the tests, along with the expected and actual results of the test conducted against the functions specified in the ST.
	ATE_IND.2	nShield V11 Test Plan, nShield V11 Test Procedures, nShield V11 Functional Specification, nShield User Guide The nCipher documentation provides the necessary information for the evaluators to develop independent tests.

Table 11 EAL4 Assurance Measures

Assurance Class	Component ID	Documentation Satisfying Component
Vulnerability Assessment	AVA_MSU.2	nShield User Guide Hardware Installation Guide nShield V11 Functional Specification The documentation provides descriptions of how administrators of the TOE can correctly administer the TOE.
	AVA_SOF.1	n/a The TOE does not contain any non-cryptographic probabilistic or permutational mechanisms and does not include any security mechanisms with strength of function claims. Therefore, no strength of function analysis is required.
	AVA_VLA.2	nShield V11 Vulnerability Analysis nShield Vulnerability Review nCipher documents their vulnerability analysis search for flaws and weaknesses in the TOE.

6.3 Strength of Function Claim

There are no non-cryptographic probabilistic or permutational mechanisms in the TOE. The overall SOF for the TOE is SOF-Basic.

This chapter provides detailed information in reference to the Protection Profile conformance identification that appears in Chapter 1.

7.1 Protection Profile Reference

This Security Target does not claim conformance to any registered Protection Profile.

7.2 Protection Profile Refinements

This Security Target does not claim conformance to any registered Protection Profile.

7.3 Protection Profile Additions

This Security Target does not claim conformance to any registered Protection Profile.

7.4 Protection Profile Rationale

This Security Target does not claim conformance to any registered Protection Profile.

This chapter provides the rationale for the selection of the IT security requirements, objectives, assumptions and threats. It shows that the IT security requirements are suitable to meet the security objectives, Security Requirements, and TOE security functional requirements.

8.1 Rationale for IT Security Objectives

This section of the ST demonstrates that the identified security objectives are covering all aspects of the security needs. This includes showing that each threat and assumption is addressed by a security objective.

The following table identifies for each threat and assumption, the security objective(s) that address it.

Table 12 Threats and Assumptions to Security Objectives Mapping

	O.CRYPTOGRAPHY	O.PROTECT_KEYS	O.SELF_PROTECT	ON.ENVIRON	ON.INSTALL	ON.NOEVILADMIN	ON.PLATFORM
T.KEY_COMPROMISE	X	X					
T.TSF_COMPROMISE			X				
A.ENVIRON				X			
A.INSTALL					X		
A.NOEVILADMIN						X	
A.PLATFORM							X
P.CRYPTOGRAPHY	X						

8.1.1 Rationale Showing Threats to Security Objectives

The following table describes the rationale for the threat to security objectives mapping.

Table 13 Threats to Security Objectives Rationale

T.TYPE	Security Objectives Rationale
T.KEY_COMPROMISE	O.CRYPTOGRAPHY mitigates this threat by providing cryptographically strong and validated mechanisms that can be used to protect the keys. O.PROTECT_KEYS counters the threat by requiring the TOE to provide protection for the keys that is uses.
T.TSF_COMPROMISE	O.SELF_PROTECT counters this threat by ensuring that the TSF can protect itself from users within the TSC. If the TSF could not maintain and control its domain of execution, it could not be trusted to control access to the resources under its control. Ensuring that the TSF is always invoked is also critical to the mitigation of this threat.

8.1.2 Rationale Showing Assumptions to Environment Security Objectives

The following table describes the rationale for the assumption to security objectives mapping.

Table 14 Assumptions to Security Objectives Rationale

A.TYPE	Environment Security Objective Rationale
A.ENVIRON	ON.ENVIRON addresses this assumption by restating it as an objective for the Administrator to satisfy.
A.INSTALL	ON.INSTALL addresses this assumption by restating it as an objective for the Administrator to satisfy.
A.NOEVILADMIN	ON.NOEVILADMIN addresses this assumption by restating it as an objective for the Administrator to satisfy.
A.PLATFORM	ON.PLATFORM addresses this assumption by restating it as an objective for the Administrator to satisfy.

8.1.3 Rationale Showing OSPs to Environment Security Objectives

The following table describes the rationale for the organisational security policies to security objectives mapping.

Table 15 OSPs to Security Objectives Rationale

P.TYPE	Environment Security Objective Rationale
P.CRYPTOGRAPHY	**O.CRYPTOGRAPHY** satisfies this policy by requiring the TOE to implement NIST FIPS 140-2 validated cryptographic operations.

8.2 Security Requirements Rationale

8.2.1 Rationale for Security Functional Requirements of the TOE Objectives

This section provides rationale for the Security Functional Requirements demonstrating that the SFRs are suitable to address the security objectives.

The following table identifies for each TOE security objective, the SFR(s) that address it.

Table 16 SFRs to Security Objectives Mapping

	O.CRYPTOGRAPHY	O.PROTECT_KEYS	O.SELF_PROTECT
FCS_CKM.1		X	
FCS_CKM.3		X	
FCS_CKM.4		X	
FCS_COP.1	X	X	
FPT_AMT.1			X
FPT_RVM.1			X
FPT_SEP.1			X

The following table provides the detail of TOE security objective(s).

Table 17 Security Objectives to SFR Rationale

Security Objective	SFR and Rationale
O.CRYPTOGRAPHY	**FCS_COP.1** ensures that all cryptographic operations are FIPS 140-2 validated.
O.PROTECT_KEYS	**FCS_CKM.1** requires the TOE to generate keys. They are generated by a cryptographic hash function, ensuring they are initially secure. **FCS_CKM.3** addresses archive and recovery of keys as they exit and enter the TOE. The algorithm for performing these operations ensures that the keys remain secure when they outside the TSC. **FCS_CKM.4** ensures the keys are securely destroyed when they are no longer needed. **FCS_COP.1** details the cryptographic operations used to create the keys, protect the keys when they are archived, and validate the keys when they are recovered.
O.SELF_PROTECT	**FPT_AMT.1** ensures that the underlying abstract machine is operating as expected. Without this assurance the correct operation of the TOE would not be assured. **FPT_SEP.1** ensures the TSF provides a domain that protects itself from untrusted users. If the TSF cannot protect itself it cannot be relied upon to enforce its security policies. **FPT_RVM.1** ensures that the TSF makes policy decisions on all interfaces that perform operations on subjects and objects that are within the TSC. Without this non-bypassability requirement, the TSF could not be relied upon to completely enforce the security policies, since an interface(s) may otherwise exist that would provide a user with access to TOE resources (including TSF data and executable code) regardless of the defined policies.

8.2.2 Rationale for Security Functional Requirements of the IT Environment Objectives

No objectives for the IT Environment are included in the ST.

8.2.3 Security Assurance Requirements Rationale

8.2.3.1 TOE Security Assurance Requirements Rationale

The TOE meets the assurance requirements for EAL4 and is augmented by ALC_FLR.1.

The TOE stresses assurance through vendor actions that are within the bounds of current best commercial practice. The TOE provides, primarily via review of vendor-supplied evidence, independent confirmation that these actions have been competently performed.

The general level of assurance for the TOE is:

A Consistent with current best commercial practice for IT development and provides a product that is competitive against non-evaluated products with respect to functionality, performance, cost, and time-to-market.

B The TOE assurance also meets current constraints on widespread acceptance, by expressing its claims against part 3 of the Common Criteria.

8.3 TOE Summary Specification Rationale

This section demonstrates that the TOE's Security Functions completely and accurately meet the TOE SFRs.

The following tables provide a mapping between the TOE's Security Functions and the SFRs and the rationale.

Table 18 SFRs to TOE Security Functions Mapping

	Key Management	Self Protection
FCS_CKM.1	X	
FCS_CKM.3	X	
FCS_CKM.4	X	
FCS_COP.1	X	
FPT_AMT.1		X
FPT_RVM.1		X
FPT_SEP.1		X

Table 19 SFR to SF Rationale

SFR	SF and Rationale
FCS_CKM.1	**Key Management** – The SF uses the keys generated by the TOE.
FCS_CKM.3	**Key Management** – The SF uses key archive and key recovery to store key blobs on the server's hard disk and retrieve them as needed.
FCS_CKM.4	**Key Management** – When the TOE is done using a key it is zeroized.
FCS_COP.1	**Key Management** – The operations specified in the SFR are used internally by the TOE to generate keys or for key blobs. They are also made available to applications using the TOE.

Table 19 SFR to SF Rationale

SFR	SF and Rationale
FPT_AMT.1	**Self Protection** – The TOE executes on the HSM hardware. Assurance of the TOE's proper operation is dependent on the HSM hardware operating properly.
FPT_RVM.1	**Self Protection** – The TOE encompasses all code executing on the HSM processor. Since the TOE strictly controls all accesses through its interfaces, it is not possible to bypass the TSF.
FPT_SEP.1	**Self Protection** – The TOE encompasses all code executing on the HSM processor and external users are not permitted to introduce new code to the HSM. The TOE executes on behalf of multiple users, but tracks each interaction separately to ensure separation between the sessions.

8.4 PP Claims Rationale

The rationale for the Protection Profile conformance claims is defined in Chapter 7, Section 7.4 Protection Profile Rationale.

8.5 Strength of Function Rationale

SOF-basic is defined in CC Part 1 section 2.3 as: "A level of the TOE strength of function where analysis shows that the function provides adequate protection against casual breach of TOE security by attackers possessing a low attack potential." Because this ST identifies threat agents with low attack potential, SOF-basic was chosen.