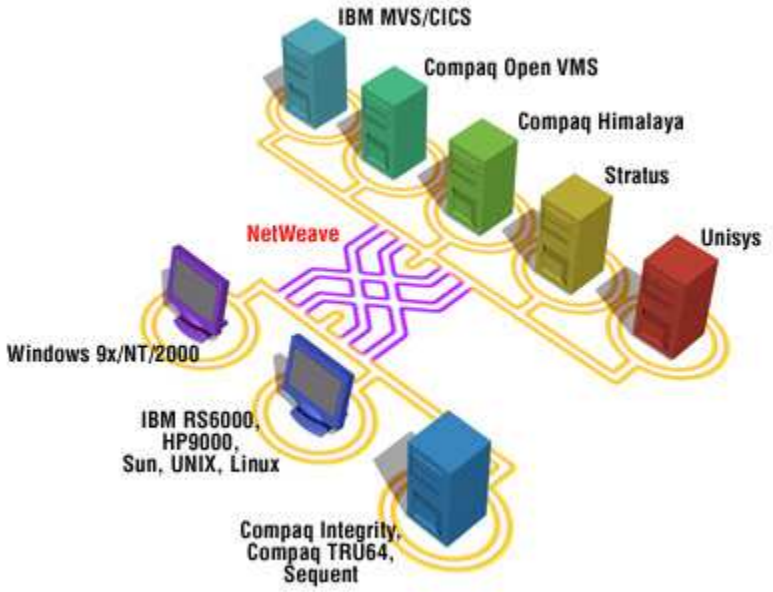
NetWeave NSK/D30
by NetWeave
Version 2.2v1



FIPS 140-2 Non-Proprietary
Security Policy
Revision 1.14

Level 1 Validation

June 2007

Table of Contents

Introduction

The sections that follow introduce this document, including the purpose for it, references used and the organizational structure of it.

Purpose

This is a non-proprietary Cryptographic Module Security Policy for the NetWeave NonStop Kernel (NSK)/D30 by NetWeave. This security policy describes how the NetWeave NSK/D30 by NetWeave meets the security requirements of FIPS 140-2 and how to run the module in a FIPS 140-2 approved mode of operation. This policy was prepared as part of the Level 1 FIPS 140-2 validation of the module.

FIPS 140-2 (Federal Information Processing Standards Publication 140-2 — *Security Requirements for Cryptographic Modules*) details the U.S. Government requirements for cryptographic modules. More information about the FIPS 140-2 standard and validation program is available on the National Institute of Standards and Technology (NIST) website at http://csrc.nist.gov/cryptval/.

References

This document deals only with the operations and capabilities of the module in the technical terms of a FIPS 140-2 cryptographic module security policy. More information is available on the module from the following sources:

- The NetWeave website (www.NetWeave.com) contains information on the full line of products from NetWeave.
- The NIST Validated Modules website (http://csrc.ncsl.nist.gov/cryptval/) contains contact information for answers to technical or sales-related questions for the module.

Document Organization

The Security Policy document is one document in a FIPS 140-2 Submission Package. In addition to this document, the Submission Package contains:

- Vendor Evidence document
- Finite State Machine
- Other supporting documentation as additional references

This Security Policy and the other validation submission documentation were produced by Corsec Security, Inc. under contract to NetWeave. With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 Validation Documentation is proprietary to NetWeave and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact NetWeave.

NSK/D30 BY NETWEAVE

Overview

The NetWeave NSK/D30 FIPS-Compliant Module application is packaged as a single binary executable file (HE2X) and the signature file containing the HMAC of the executable and the signature file itself (sigfile). All libraries that the NetWeave NSK/D30 FIPS-Compliant Module utilizes are statically linked. The NetWeave NSK/D30 FIPS-Compliant Module provides the following basic functionalities:

- Transparent Secure Sockets Layer (SSL) and Transport Layer Security (TLS) proxy server between a Hewlett-Packard (HP) Nonstop Server and arbitrary client platforms.
- Encryption of NetWeave Distributed Services sessions only.
- Employs the XYGATE /ESDK (Encryption Software Development Kit) to provide all cryptographic functionality
- Runs on HP NonStop Servers using the Guardian D39 version of the operation system

Module Interfaces

NetWeave NSK/D30 by NetWeave is classified as a Multi-chip standalone module for FIPS 140-2 purposes. The module is being tested on the HP Nonstop Guardian D39 server platform. The physical boundary of the NetWeave NSK/D30 is defined by the metal enclosure over the board.

The module supports the physical interfaces of a general purpose computer (GPC). The physical interfaces include the computer keyboard port, optical drives, floppy disk, mouse, serial ports, parallel ports, networks ports, monitor port and power plug. The functional module interface exists in the software. See Figure 1 for a standard GPC block diagram.

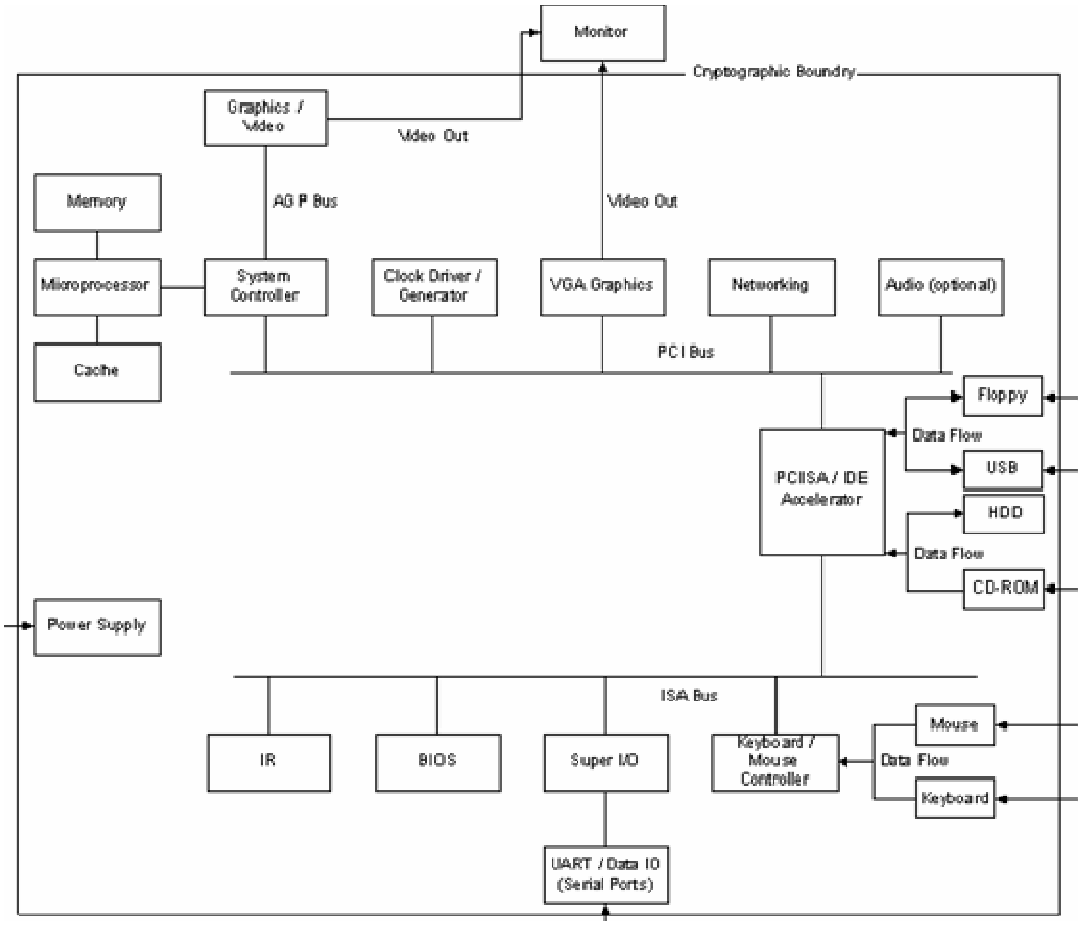


Figure 1 - Standard GPC Block Diagram

The NSK module employs the XYGATE /ESDK to provide all cryptographic functionality. The ESDK module provides scripts and graphical user interfaces to interact with the components. Figure 2 below shows the ESDK module's logical cryptographic boundary.

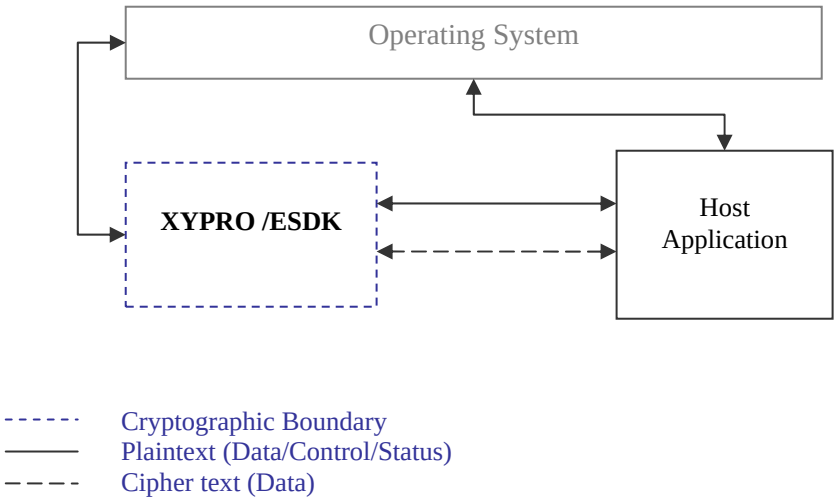


Figure 2 – Logical Cryptographic Boundary

Note that the NetWeave NSK is one possible Host Application, in which case the Cryptographic Boundary extends to the NetWeave NSK as well.

All of these physical interfaces are separated into logical interfaces defined by FIPS 140-2, as described in the following table:

Module Physical Interface	FIPS 140-2 Logical Interface	Module Mapping
PC Network port	Data Input Interface	Data input is provided through IP packets over a network port
PC Network Port	Data Output Interface	Data output is sent in the form of IP packets over a network port
PC Network Port, Serial Port, Keyboard port, Mouse port, PC Power Button	Control Input Interface	Control Input is provided through the keyboard and mouse in the form of specific commands and command line options that dictate the module's behavior.
Light Emitting Diodes, PC Monitor, PC Network Port	Status Output Interface	Status output is displayed in log files and display output.
PC Power Interface	Power Interface	Not Applicable.

Table 1 – FIPS 140-2 Logical Interfaces

Roles and Services

The module supports role-based operation. There are two main roles in the module (as required by FIPS 140-2) that operators may assume: a Crypto-Officer role and User role. The Crypto-Officer is responsible for initializing the module and configuring it to run in a FIPS approved mode. After the switch to the FIPS

approved mode of operation both the Crypto-Officer and User will be able to utilize the functionality of the module. The various services offered by the module are described below. Operators of the module implicitly assume a role based on the services of the module that they are using. Since all services offered by the module can only be used by either the Crypto-Officer or the User (never both) the roles are mutually exclusive.

Service	Description	Input	Output	CSP	Role
Installing the module	The Crypto-Officer has the ability to install the module.	Command	Result of Installation	None	Crypto-Officer
Uninstalling/ Removing the module	The Crypto-Officer has the ability to delete the module.	Command	Uninstalled Module	Integrity Check Key (delete)	Crypto-Officer
SSL/TLS Proxy Server	All users have the ability to operate the module	Command	SSL/TLS session established	Public/Private Keys (Read), Symmetric Key (Full), HMAC Key (Full)	User

Table 2 –Services, Descriptions, Inputs and Outputs

Authentication Mechanisms

The module does not support authentication.

Cryptographic Key Management

Symmetric Key Algorithms

The table below lists all the symmetric-key cryptographic algorithms that the NSK supports. The following modes of operation are defined: F – Cipher Feedback, OFB – Output Feedback, ECB – Electronic Code Book, CBC – Cipher Block Chaining.

Algorithm	Modes Implemented	Key Sizes	Certificate Number
AES	CBC, ECB, CFB, OFB	128, 192, 256 bits	505
Triple-DES	CBC, ECB, CFB, OFB	112, 168 bits	515

Hashing Algorithms

Algorithm	Certificate Number
SHA-1	576
SHA-256	576
HMAC SHA-1	258

Public Key and Signature Algorithms

Algorithm	Key Sizes	Certificate Number
RSA	1024, 2048, 4096 bits	220
DSA	1024 bits	209

Non-FIPS Approved Algorithms Allowed in FIPS mode

The module supports the following algorithms that are not FIPS approved but allowed in a FIPS mode of operation.

Public Key Algorithms

Algorithm
Diffie-Hellman (provides 80 to 152 bits of encryption strength)
RSA key wrapping (provides 80 to 152 bits of encryption strength)

Non-FIPS Approved Algorithms

The module supports the following non-approved cryptographic algorithms:

Symmetric Key Algorithms

Algorithm	Modes Implemented	Key Sizes
Data Encryption Standard (DES)	CBC, ECB, CFB, OFB	40 – 56 bits
International Data Encryption Algorithm (IDEA)	CBC, ECB, CFB, OFB	128 bits
Rivest Cipher 2 (RC2)	CBC, ECB, CFB, OFB	40-1024 bits
RC4	CBC, ECB, CFB, OFB	40-2048 bits

Hashing Algorithms

Algorithm
Message Digest 2 (MD2)
MD4
MD5
HMAC MD5

Public Key Algorithms

Algorithm
RSA 512-bit and 768-bit
Diffie-Hellman (non-compliant less than 80-bits provided down to 56-bits)
RSA key wrapping (non-compliant less than 80-bits provided down to 56-bits)

The module supports the following critical security parameters:

Cryptographic Key	Description	Key Type	Storage and Zeroization
Private/public keys	These keys are either in public key certificate form or stored as private key files. These keys also included generated Diffie-Hellman keys.	RSA, DSA, or Diffie-Hellman	Public keys are stored in certificates. Private keys are stored as files on the hard drive. Keys can be deleted by deleting the file that stores them. Diffie-Hellman keys exist only in memory as part of an SSL/TLS session and are zeroized when a session is ended.
Symmetric keys	These keys are used for bulk encryption/decryption during SSL/TLS sessions	AES or Triple-DES	Symmetric keys are ephemeral and stored in RAM. They are used for the duration of an SSL/TLS session and then are zeroized.
HMAC keys	These keys are used for authentication and integrity checking during SSL/TLS sessions	HMAC-SHA-1	HMAC keys follow the same storage/zeroization rules as Symmetric keys
PRNG Seed	Random entropy data collected from various sources that is used to seed the ANSI X9.31 PRNG.	Random entropy data	The PRNG seed is stored in volatile memory and is zeroized when a new seed is generated.
Integrity Check Key	Externally generated key used to verify integrity of the module.	HMAC-SHA-1	The key is hard-coded into the source code of the module and is deleted when the module is deleted.

Table 3 – Keys and CSPs

Key Generation

All keys are either generated by the module using the FIPS approved pseudo-random number generator included in the underlying library or provided as part of X.509 certificates in the case of Public keys.

Key Establishment

Diffie-Hellman keys are generated internally by the module, while all other public/private keys are generated externally or reside in X.509 public key certificates. Symmetric keys are established using either RSA or DH as part of a SSL/TLS session. Both RSA key transport and DH key agreement provide 56-bits up to 152-bits of key strength. In a FIPS approved mode of operation, the key a minimum key strength of 80-bits must be used for RSA key transport and Diffie-Hellman key agreement.

Key Storage

Public keys are stored in certificates. Private keys are stored in PKCS #15 formatted files within the file system, which have been properly encrypted to prevent access or modification. Diffie-Hellman, symmetric and HMAC keys are stored ephemerally in RAM for the duration of an SSL/TLS session then zeroized when the session ends.

Key Usage

Users have complete access to symmetric and HMAC keys through the SSL/TLS session services. Public key certificates and encrypted private key files are read from as part of SSL/TLS negotiations. Crypto-Officers uninstalling the module zeroize and delete the Integrity Check Key as part of uninstallation.

Key Zeroization

Diffie-Hellman, symmetric and HMAC keys are zeroized when their associated SSL/TLS session is terminated. Public/Private keys (except for Diffie-Hellman keys) are zeroized by deleting the file in which they reside. The Integrity Check Key is zeroized upon uninstallation of the module.

Self-Tests

The NSK/D30 does not perform its own self-tests; instead it relies on the self-testing functionality of the XYGATE /ESDK. The NSK/D30 statically links the ESDK library so all self-testing requirements are met.

Power-Up Self-Tests

The power-up self-tests implemented in the ESDK library and used by NSK include known answer tests (KAT) for Approved algorithms (Triple-DES, AES, SHA-1, SHA-256, HMAC-SHA-1, and RSA) and non-Approved algorithms (DES, IDEA, RC2, RC4, MD2, MD4, MD5, HMAC-MD5, and Diffie-Hellman). Also executed at power-up is a PRNG KAT, pairwise consistency tests for DSA (FIPS-Approved) and a software integrity check with HMAC SHA-1.

Because the ESDK module is a software library, the power-up self-tests are run when a host application (such as the NSK/D30 module) performs the call to the cryptInit() function. Linking the ESDK module invokes certain API function calls, which triggers the self tests. If these self tests fail, then the API calls fail and the API will not be functional.

Conditional Self-Tests

The module performs two conditional self-tests: a pairwise consistency test each time the module generates a Diffie-Hellman or RSA public/private key and a

continuous random number generator test each time the module produces random data. The seed for the PRNG is also tested. If any error occurs during the test, the cryptographic module will be unloaded from memory

Design Assurance

NetWeave stores their source code within a Concurrent Versions System (CVS) repository stored in NetWeave's data center. Code must be checked out to edit and then checked in to become part of the final source tree for the NSK/D30. Corsec Security, Inc. stores documents in Visual Source Safe to ensure that documents are not tampered with and are only edited by those who have the proper permissions.

Mitigation of Other Attacks

In a FIPS mode of operation the module does not claim to mitigate any additional attacks.

SECURE OPERATION

The NSK/D30 meets Level 1 requirements for FIPS 140-2. The sections below describe how to place and keep the module in FIPS-approved mode of operation.

Initial Setup

The module has to be installed on the HP Nonstop Guardian D39 server platform. The FIPS mode of operation for the NSK involves use of the ESDK to provide a software integrity test and power up cryptographic algorithm known answer tests to ensure proper functioning of the module. Through execution of the ESDK function cryptInit(), all power up self tests are performed. The module will operate in a FIPS mode when started.

Crypto-Officer Guidance

The Crypto-Officer is responsible for performing the steps detailed in the Initial Setup section above.

User Guidance

The user is responsible for ensuring that the configuration file is properly configured by setting the line ENCRYPTION_REQUIRED = 1 and the line ENCRYPTION_MODE = SSL. These two lines will ensure that the module only uses TLS and only uses TLS cipher suites that make use of FIPS approved algorithms. Please note that while ENCRYPTION_MODE is set to "SSL", use of the XYPRO /ESDK also operating in FIPS mode ensures that the protocol used is actually TLS. The ESDK also ensures that TLS cipher suites employing only FIPS-approved algorithms are used. Included below is a sample configuration file

with the relevant lines in red text. Status output is done on a log file that is also configured in the configuration file. The line that stipulates which log file to use is in blue.

```
[LICENSE_GROUP]
* 20071231 -- 192.168.17.16
LICENSE_KEY = DRLHBDGBGGFCGFQFPMCHBP

[K202]
PROTOCOL = TCPIP
TCP_PROCESS_NAME = $ZTC0
TCPIP_ADDRESS = 192.168.17.16
TCPIP_PORT = 8420
LM_PERMITTED = 1
LM_BUFFER_SIZE = 65000
ENCRYPTION_REQUIRED = 1
ENCRYPTION_TYPE = SSL
NEWCALL_PUBLIC = 1

[SSL]
PUBLICKEY_FILE = $lib15.xygatehe.f2srvcrt
PRIVATEKEY_FILE = $lib15.xygatehe.f2srvkey
CA_KEY_FILE = $lib15.xygatehe.rootcrt
SSL_DEBUG = 0
SIGNATURE_FILE = $lib11.testmpx.sigfile

[HE2]
@TRACE_FILE@ = $s.#he2x
```

ACRONYMS

Acronym	Definition
AES	Advanced Encryption Standard
CVS	Concurrent Versions System
DES	Data Encryption Standard
ESDK	Encryption Software Development Kit
FIPS	Federal Information Processing Standard
GPC	General Purpose Computer
HMAC	Hashed Message Authentication Code
HP	Hewlett-Packard
IDEA	International Data Encryption Algorithm
IP	Internet Protocol
NIST	National Institute of Standards and Technology
NSK	NonStop Kernel
SHA	Secure Hash Algorithm
SSL	Secure Sockets Layer
TLS	Transport Layer Security