



# Integration Guide

## Apache2.2.x, OpenSSL 0.9.8x and nCipher Modules

- **AIX 5.3**
- **Red Hat Enterprise Linux 5**
- **Solaris 10 SPARC**
- **Solaris 10 X86**



These installation instructions are intended to provide step-by-step instructions for installing nCipher software with third-party software. These instructions do not cover all situations and are intended as a supplement to the nCipher documentation provided with nCipher products.

Disclaimer: nCipher Corporation Ltd disclaims all liabilities regarding third-party products and only provides warranties and liabilities with its own products as addressed in the Terms and Conditions for Sale. nCipher is a registered trademark of nCipher Corporation Limited. Any other trademarks referenced in this document are the property of the respective trademark owners. © Copyright 2008 nCipher Corporation Ltd, Cambridge, United Kingdom.

Version: 1.1



**TABLE OF CONTENTS**

**1.0 INTRODUCTION ..... 2**

**2.0 OVERVIEW ..... 2**

**3.0 SUPPORTED NCIPHER FUNCTIONALITY ..... 3**

**4.0 REQUIREMENTS ..... 3**

**5.0 PROCEDURES ..... 3**

    5.1. INSTALLING AND CONFIGURING THE NCIPHER MODULE ..... 3

    5.2. INSTALLING OPENSLL ..... 4

    5.3. INSTALLING THE APACHE HTTP SERVER ..... 7

    5.4. CONFIGURING THE APACHE HTTP SERVER TO USE THE NCIPHER MODULE ..... 1

    5.5. TROUBLESHOOTING ..... 2

**FURTHER INFORMATION ..... 3**



## 1.0 Introduction

This guide explains how to integrate an nCipher module (an nShield module or a netHSM) with the Apache HTTP Server. This guide assumes that you have read the *nShield Quick Start Guide* and are familiar with the documentation and setup process for the Apache HTTP Server.

---

*Note:* All nCipher documentation is available at: <http://www.ncipher.com/documentation>.

---

## 2.0 Overview

The nCipher Hardware Security Module (HSM) integrates with the Apache2 Server to provide significant performance improvements by off-loading cryptographic operations from the Apache2 Server to the nCipher module. In addition, the nCipher module provides extra security by protecting and managing the server's high value SSL private key within a FIPS 140-2 certified hardware security module. You can integrate the Apache HTTP Server with an nShield module by using nCipher's CHIL (Cryptographic Hardware Interface Library) interface.

The benefits of using an nCipher module with the Apache HTTP Server are:

- secure storage of the private key
- FIPS 140-2 level 3 validated hardware
- improved server performance by offloading the cryptographic processing
- enables full life cycle management of the keys
- provides failover support
- load balancing between modules.

The integration between the nShield module and the Apache HTTP Server has been tested for the following combinations:

Operating System	nCipher Version	Apache Version	PCI Support	Ethernet Support
AIX 5.3	11.00	2.2.6	Yes	Yes
Red Hat Enterprise Linux 5	11.00	2.2.8	Yes	Yes
Solaris 10 SPARC	11.00	2.2.8	Yes	Yes
Solaris 10 x86	11.00	2.2.6	Yes	Yes



### 3.0 Supported nCipher functionality

You can access the following nCipher functionality when you integrate an nCipher module with the Apache HTTP Server.

- |  |   |   |
|--|---|---|
| <input checked="" type="checkbox"/> Soft Cards     | <input checked="" type="checkbox"/> Key Management  | <input checked="" type="checkbox"/> Strict FIPS Support |
| <input checked="" type="checkbox"/> Key Recovery   | <input checked="" type="checkbox"/> Module Only Key | <input checked="" type="checkbox"/> K of N Card Set     |
| <input checked="" type="checkbox"/> Key Generation | <input checked="" type="checkbox"/> Key Import      | <input checked="" type="checkbox"/> Fail Over           |
| <input type="checkbox"/> Fall Back                 | <input checked="" type="checkbox"/> Load Balancing  | <input checked="" type="checkbox"/> Preload support     |

### 4.0 Requirements

Before you begin the integration process:

- read the *nShield Quick Start Guide* or the *netHSM Quick Start Guide* as appropriate.
- familiarize yourself with the documentation and setup process for the Apache2 HTTP server.

Before running the setup program, you need to know:

- the number and quorum of Administrator Cards in the Administrator Card Set (ACS), and the policy for managing these cards
- whether the application keys are protected by the module or an Operator Card Set (OCS) with pass phrase
- the number and quorum of Operator Cards in the OCS, and the policy for managing these cards
- whether the security world is compliant with FIPS 140-2 at level 3.

For more information on administering an nCipher module, see the *nShield User Guide* or the *netHSM User Guide* as appropriate.

### 5.0 Procedures

The installation and configuration is performed in several steps:

1. Installing and configuring the nCipher module
2. Installing OpenSSL
3. Installing Apache HTTP Server
4. Configuring the Apache HTTP Server to use nCipher's CHIL engine

#### 5.1. Installing and configuring the nCipher module

To install and configure the nCipher module, complete the following steps:

1. Install the nShield hardware in the required computer. For instructions, see the *Hardware Installation Guide*.
2. Install the software and create the nCipher security world as described in the *nShield Quick Start Guide*.



*Note: nCipher always recommends uninstalling any existing nCipher software before installing the new software.*

## 5.2. Installing OpenSSL

To install OpenSSL, complete the following steps:

1. Create the directory in which OpenSSL is to be built by using the command:

---

```
mkdir openssl_dir
```

---

2. Download the **openssl-0.9.xx.tar.gz** file from the URL <http://www.openssl.org/source>.
3. Copy the **openssl-0.9.xx.tar.gz** file into the **openssl\_dir** directory.
4. Navigate to the **openssl\_dir** directory.
5. Decompress the **openssl-0.9.xx.tar.gz** file by running the command:

---

```
gzip -d openssl-0.9.xx.tar.gz
```

---

6. Untar the **openssl-0.9.xx.tar** file by running the command:

---

```
tar -xvf openssl-0.9.xx.tar
```

---

*Note: To build OpenSSL and Apache, you must log in as a root user with administrative privileges.*

7. Navigate to the **openssl\_dir/openssl-0.9.xx** directory by running the command:

---

```
cd openssl_dir/openssl-0.9.xx
```

---

8. Copy the patch file from the nCipher's **openssl** directory to the current directory:

---

```
cp /opt/nfast/toolkits/openssl/openssl098x-patch.txt  
/openssl_dir/openssl-0.9.xx/
```

---

9. Download the patch utility from the URL as given below:

- a For AIX 5.3, download **patch-2.5.4-4.aix4.3.ppc.rpm** from [http://www.bullfreeware.com/download/wpar\\_tt/listaixopensourcepms.html](http://www.bullfreeware.com/download/wpar_tt/listaixopensourcepms.html)
- b For Solaris 10 x86, download **patch-2.5.4-sol10-x86-local.gz** from <http://www.sunfreeware.com/indexintel10.html>
- c For Solaris 10 sparc, download **patch-2.5.4-sol9-sparc-local.gz** from <http://www.sunfreeware.com/indexsparc10.html>

*Note: The patch utility for AIX5.3 is not available.*

10. Install the patch utility by running the command:

On Aix 5.3:

---

```
rpm -i patch-2.5.4-4.aix4.3.ppc.rpm
```

---

The above command will install the patch utility in the **/opt/freeware/bin** directory.



On Solaris 10 x86:

---

```
gzip -d patch-2.5.4-sol10-x86-local.gz
pkgadd -d patch-2.5.4-sol10-x86-local
```

---

The above command will install the patch utility in the **/usr/local/bin** directory.

On Solaris 10 sparc:

---

```
gzip -d patch-2.5.4-sol9-sparc-local.gz
pkgadd -d patch-2.5.4-sol9-sparc-local
```

---

The above command will install the patch utility in the **/usr/local/bin** directory.

11. Apply the Openssl patch by running the command:

On Aix 5.3:

---

```
/opt/freeware/bin/patch -p1 < openssl098x-patch.txt
```

---

On Solaris 10 x86, Solaris 10 sparc and Red Hat Enterprise Linux 5:

---

```
/usr/local/bin/patch -p1 < openssl098x-patch.txt
```

---

12. Set the PATH environment variable as follows:

On Aix 5.3:

---

```
export PATH=$PATH:/usr/ccs/bin
export PATH=$PATH:/usr/local/ssl
export PATH=$PATH:/usr/local/ssl/bin
export PATH=$PATH:/opt/freeware/bin
```

---

On Solaris 10 x86 and Solaris 10 sparc:

---

```
export PATH=$PATH:/usr/ccs/bin
export PATH=$PATH:/usr/local/ssl
export PATH=$PATH:/usr/local/ssl/bin
export PATH=$PATH:/usr/sfw/bin
export PATH=$PATH:/usr/local/bin
```

---

**Note:** For Red Hat Enterprise Linux 5 does not need to set above mentioned PATH environment Variables.

---

On Aix 5.3:

---

```
export LIBPATH=$LIBPATH:/usr/local/ssl/lib
```

---



On Solaris 10 x86 and Solaris 10 sparc:

---

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/ssl/lib
```

---

**Note:** Before proceeding further, ensure that the latest *apr*, *aprutil*, *gcc*, *libgcc*, *libiconv*, *libcrypt* utilities are installed in Solaris 10 x86, Solaris 10 sparc.

*You do not need to set the LD\_LIBRARY\_PATH variable for Red Hat Enterprise Linux 5 .*

---

13. Run the config script:

---

```
./config -ldl
```

---

Build OpenSSL by running the command:

---

```
make
```

```
make install
```

---

14. Change to the OpenSSL applications directory by running the command:

---

```
cd apps
```

---

15. For OpenSSL to use the nCipher module, the nCipher CHIL library must be on the path for any application that uses OpenSSL. Optionally, add this path to a temporary shell from the command line. For example, run the command:

For Aix 5.3:

---

```
export LIBPATH=$LIBPATH:/opt/nfast/toolkits/hwcrhk
```

---

On Solaris 10 x86, Solaris 10 sparc and Red Hat Enterprise Linux 5:

---

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/nfast/toolkits/hwcrhk
```

---

16. In order to check that OpenSSL is configured properly and working, run the following commands:

---

```
./openssl engine -t chil
```

---

For the above command, the output should be as follows:

---

```
(chil) CHIL hardware engine support
      [ available ]
```

---

**Note:** If the CHIL engine is unavailable, your LIBPATH/LD\_LIBRARY\_PATH is not set. Set the path again and check the availability of the CHIL engine.

---

```
/openssl speed rsa -engine chil -elapsed -multi 50
```

---



If the OpenSSL installation is successful, then the following program output is displayed:

---

```
engine "chil" set.
Forked child 0
+DTP:512:private:rsa:10
Forked child 1
+DTP:512:private:rsa:10
Forked child 2
+DTP:512:private:rsa:10
```

---

### 5.3. Installing the Apache HTTP Server

To install the Apache HTTP Server, complete the following steps:

1. Create the directory in which you will build the Apache2.2 server by using the command:

---

```
mkdir apache_dir
```

---

2. Download the **httpd-2.2\_xx.tar.gz** file from <http://www.openssl.org/source>.
3. Copy the **httpd-2.2\_xx.tar.gz** file into the **apache\_dir** directory.
4. Navigate to the **apache\_dir** directory.
5. Decompress the **httpd-2.2\_xx.tar.gz** file by running the command:

---

```
gzip -d httpd-2.2_xx.tar.gz
```

---

6. Untar the **httpd-2.2\_xx.tar** file by running the command:

---

```
tar -xvf httpd-2.2_xx.tar
```

---

7. Navigate to **apache\_dir/httpd-2.2.x** directory by running the command:

---

```
cd apache_dir/httpd-2.2.x
```

---

8. Configure the Apache server by running a command of the form:

---

```
./configure --enable-ssl --with-ssl=/openssl_dir/openssl-0.9.xx CFLAGS=-DSSL_EXPERIMENTAL_ENGINE
```

---

**Note:** For Solaris 10 Sparc (SunOS 10), configure Apache by running a command of the form:

---

```
# ./configure --enable-ssl --with-ssl=/usr/local/ssl/ --with-included-apr CFLAGS=-DSSL_EXPERIMENTAL_ENGINE --with-apr=/usr/local/apr
```

---

Build Apache by running the following commands:

---

```
make
make install
```

---



## 5.4. Configuring the Apache HTTP Server to use the nCipher module

To configure the Apache HTTP Server to use the nCipher module, complete the following steps:

1. Open the file `/usr/local/apache2/bin/apachectl` in a text editor. Locate the line `#!/bin/sh` and immediately following it add the line:

For Aix 5.3:

---

```
LIBPATH=/opt/nfast/toolkits/hwcrhk
```

---

On Solaris 10 x86, Solaris 10 sparc and Red Hat Enterprise Linux 5:

---

```
LD_LIBRARY_PATH=/opt/nfast/toolkits/hwcrhk
```

---

2. Open the file `/usr/local/apache2/conf/httpd.conf` in a text editor. Locate the line `ServerName www.example.com:80` and enter the proper machine name or IP address.
3. Create the `ssl.key` and `ssl.crt` directories in the `/usr/local/apache2/conf/` directory.
4. Open the file `/usr/local/apache2/conf/extra/httpd-ssl.conf` in a text editor, and edit the file as follows:
  - a. Comment out the `SSLMutex` path directive.
  - b. Locate the line `<IfDefine SSL>` and enter the following lines below:

---

```
<IfModule !mod_ssl.c>
    LoadModule ssl_module modules/mod_ssl.so
</IfModule>
SSLCryptoDevice chil
```

---

- c. Change `Listen 80` to `Listen 443`.

- d. Locate the following line:

---

```
SSLCertificateFile /usr/local/apache2/conf/ssl.crt/server-dsa.crt
```

---

Rename `server-dsa.crt` to the generated self-cert file.

- e. Locate the following line:

---

```
SSLCertificateKeyFile /usr/local/apache2/conf/ssl.key/server-dsa.key
```

---

Rename `server-dsa.key` to the generated key file.

- f. Locate the line `ServerName www.example.com:80` and edit this line to ensure that the values of the IP address and the port number (`<system ip address>:443`) are correct and then save the changes.

5. Use the `generatekey` command-line utility in order to generate a embed key:

---

```
/opt/nfast/bin/generatekey embed
```

---

6. Generating a key with the `generatekey` command-line utility stores the following information (in which `key_name` represents the name given to the key being generated):

- The key, in the file `key_name`



- The X.509 self-certificate, in the file **key\_name\_selfcert**
  - The X.509 (base 64 encoded PKCS #10) certificate request, in the file **key\_name\_req**
7. Copy the **key\_name** and **key\_name\_selfcert** files to **/usr/local/apache2/conf/ssl.key** and **/usr/local/apache2/conf/ssl.crt** directories respectively.
  8. Create a directory called **preload** with root as user and nfast as group at **/opt/nfast/kmdata** directory.
  9. In order to start the SSL-enabled nCipher-protected Apache HTTP Server successfully, run the following commands:

For token protected keys:

---

```
/opt/nfast/bin/preload -f /opt/nfast/kmdata/preload/apache --cardset-  
name=<token_name> /usr/local/apache2/bin/apachectl -DSSL
```

---

For Softcard protected keys:

---

```
/opt/nfast/bin/ppmk --preload -f /opt/nfast/kmdata/preload/apache  
<softcard_name> /usr/local/apache2/bin/apachectl -DSSL
```

---

For Module protected keys:

---

```
/opt/nfast/bin/preload -f /opt/nfast/kmdata/preload/apache -M  
/usr/local/apache2/bin/apachectl -DSSL
```

---

## 5.5. Troubleshooting

**Problem:** There is a problem compiling OpenSSL and Apache on Solaris 10 x86.

**Action/Solution:** Apply the Solaris x86 specific GCC run-time environment patch.

To apply the patch, complete the following steps:

1. Go to <http://www.openssl.org/~appro/values.c>
2. Copy the content displayed on the web page into a file and save it as **values.c**.
3. Navigate to the root directory.
4. Run the command:

---

```
ksh -f values.c
```

---

## Further information

This guide forms one part of the information and support provided by nCipher. You can find additional documentation, including *User Guides*, in the document directory of the CD-ROM for your nCipher product.

All nCipher product documentation is available from the nCipher web site at:

<http://active.ncipher.com/documentation/>

### **nCipher Corporation**

**Cambridge, UK**

Jupiter House

Station Road

Cambridge CB1 2JD

UK

**Tel: +44 (0) 1223 723666**

**Fax: +44 (0) 1223 723601**

### **nCipher Inc**

**Boston Metro Region, USA**

92 Montvale Avenue, Suite 4500

Stoneham

MA 02180

USA

**Tel: +1 (781) 994 4008**

**Fax: +1 (781) 994 4001**

### **Internet addresses**

**Web site:** <http://www.ncipher.com/>

**Support:** <http://www.ncipher.com/support>

**Online documentation:** <http://active.ncipher.com/documentation>