# Entrust nShield Integration Guide

SHARDSECURE™

# Introduction

ShardSecure supports external key management by integration with Entrust nShield HSMs. This document describes how to configure the ShardSecure cluster to operate with Entrust nShield HSMs. The HSM can be used to manage the master key for TDE (encrypting data at rest).
***Note: The HSM integration requires that the HSM provider supports PKCS#11***

# TDE - Pointer data encryption

ShardSecure supports master key management via an external HSM.

Pointer data stored in the internal store is transparently encrypted with industry standard AES encryption using a set of internal keys. These keys are encrypted while stored and decrypted for use when needed.

The encryption and decryption of these keys can be offloaded to an external HSM system for secure access and storage.

## Configuration

To delegate the key management for encryption to an HSM, the following properties should be included in the application.properties file of each node.
Replace the values below to match the HSM configuration.

```
shardsecure.cluster.encryption=true #to enable TDE
shardsecure.cluster.encryption.hsm=true #to enable HSM TDE
shardsecure.cluster.encryption.hsm.password=secret
shardsecure.cluster.encryption.hsm.configfile=pkcs11_config.cfg
shardsecure.cluster.encryption.masterkeyname=cluster.master.key
```

## Key rotation

No matter if the encryption master key is supplied by an HSM or a regular Java Key Store, ShardSecure can be configured to rotate the keys automatically.
***Note: The new master key must be present in the relevant key store before ShardSecure can attempt to automatically rotate it into use.***

To enable master key rotation, an additional property should be added in the application.properties file as shown below (the upper property).

```
shardsecure.cluster.encryption.masterkeyname.format=cluster.master.key.%d
shardsecure.cluster.encryption.masterkeyname=cluster.master.key.0
```

The lower property is from the previous section. However, the two properties are related. The new property (with suffix “format”) must include the placeholder “%d” which must correspond to an incremental sequence number. The placeholder may be placed anywhere in the key name. In the other property, that placeholder should be replaced with the sequence number of the first master key name. ShardSecure will always increment the sequence number by one. If a master key name with the new sequence number does not exist, the rotation will fail. The actual rotation is performed by a janitor job which must be activated (manually or scheduled).

# PKCS#11

Prior to starting the configuration of the PKCS#11 integration, ensure that the PKCS#11 shared libraries are obtained from the Entrust nShield. In the example below the provider for Entrust nShield PKCS#11v 12.80.4 is installed.

## Installation

The installation must be repeated on each node that will be included in the ShardSecure cluster. Install the shard library file in a location that the ShardSecure engine can read and ensure that the access rights allow read rights for the ShardSecure engine user.

# HSM integration

This section describes how to integrate Entrust HSMs with ShardSecure.

## Entrust configuration

| nShield Model | Security World Client | Connect Image | Security World Version | FIPS Firmware | FIPS 140-2 L3 Tested | Key Protection Methods tested |
|---|---|---|---|---|---|---|
| Connect XC | 12.80.4 | 12.80.4 | v3 - DLf3072s256mAEScSP800131Ar1 | 12.50.11 | No | PKCS#11 |
| Solo XC | 12.80.4 | | v3 - DLf3072s256mAEScSP800131Ar1 | 12.50.11 | No | PKCS#11 |

### HSM Setup

Download the Security World Installation package ISO available from Entrust Support.

Install the Security World client on each cluster node, at least the HWSP (Hardware Support) and CTLS (Core) packages are required for ShardSecure integration.

Once the packages are installed follow the Entrust configuration and setup documentation for the HSM module that will be used. Ensure that the startup scripts for the local services are installed and started on each ShardSecure Cluster node.

Run the following command to test the installation and configuration.

```
/opt/nfast/bin/enquiry
```

If everything is operating correctly the command will report available HSM services.

It is now possible to generate the cluster master key using the following HSM command.

```
/opt/nfast/bin/generatekey --generate pkcs11 plainname=cluster.master.key
size=256 type=AES
```

The default path to the PKCS#11 library is /opt/nfast/toolkits/pkcs11/libcknfast.so (cknfast.dll on windows) If the Security World was installed in another location, change the location in the PKCS#11 configuration file below.

### Configuration file for PKCS#11

```
name=NFastJava
library=/opt/nfast/toolkits/pkcs11/libcknfast.so
```

```
slotListIndex=0
attributes(*,CKO_PUBLIC_KEY,*) = {
 CKA_TOKEN = false
}
attributes(*,CKO_PRIVATE_KEY,*) = {
 CKA_TOKEN = true
 CKA_PRIVATE = false
 CKA_SIGN = true
 CKA_DECRYPT = true
}
disabledMechanisms = {
 CKM_SHA1_RSA_PKCS
 CKM_SHA256_RSA_PKCS
 CKM_SHA384_RSA_PKCS
 CKM_SHA512_RSA_PKCS
 CKM_MD2_RSA_PKCS
 CKM_MD5_RSA_PKCS
 CKM_DSA_SHA1
 CKM_ECDSA_SHA1
}
```