

KCA-EJBCA with nCipher HSM

Migration guide


2007-08-23

1 Document history

<i>Version</i>	<i>Datum</i>	<i>Name</i>	<i>Comment</i>
0.1	2007-07-27	Tomas Gustavsson	Migrating nCipher keys
1.0	2007-08-23	Tomas Gustavsson	Information and details of the complete migration
1.1	5/14/09	Tomas Gustavsson	Translated to english from swedish

Contents

1 Document history.....	2
2 Migrating KCA to EJBCA.....	3
2.1Keon CA.....	3
2.2EJBCA.....	3
3 Details of the migration.....	3
3.1Migrating KCAs signing keys.....	3
3.1.1PKCS11.....	3
3.1.2Signing keys.....	4
3.2Import CA.....	9
3.3Import user certificates.....	10

 PrimeKey Solutions	KCA-EJBCA-migration, nCipher HSM	Sidnr / Page no 3 (10)
Uppgjort / Author Tomas Gustavsson	Sekretess / Confidentiality OPEN	
Godkänd / Authorized	Datum / Date 15/05/09	Version 1.1

2 Migrating KCA to EJBCA

Migrating an installation of KCA to EJBCA includes several steps:

- Migrating the CA signing keys on an nCipher HSM so they can be used by EJBCA.
- Import CAs in EJBCA.
- Import user certificates in EJBCA.

2.1 Keon CA

Installation of a KCA was performed:

- Windows Server 2003
- KCA 6.5
- One root CA – TestKCARootCA
- One sub CA – TestKCASubCA
- CA-keys on nCipher nShield PCI kort
- 5 users signed by TestKCASubCA

After the installation of this environment a backup of the nCipher security workd, CA-certificates and user certificates was performed.

2.2 EJBCA

After the migration to EJBCA the same CAs are found in a new environment:

- Utuntu Linux 7.04 AMD64
- Jboss 4.2.0, MySQL 5.0
- EJBCA 3.5 beta
- One root CA – TestKCARootCA
- One sub CA – TestKCASubCA
- CA-nycklar on nCipher nShield PCI kort
- 5 users signed by TestKCASubCA

3 Details of the migration

3.1 Migrating KCAs signing keys


3.1.1 PKCS11

Configure PKCS#11 for nCipher.

For nCipher the slot number is the name of some virtual slot, which we don't know. Luckily we can use the default slot by not giving the slot number. We can also use the slotIndex instead. See the EJBCA HSM documentation for nCipher PKCS#11, as found in the EJBCA User Guide. When we want to use nCipher PKCS#11 you first have to configure a few parameters.

Create the file /opt/nfast/cknfastrc with the following contents:

```
CKNFAST_LOADSHARING=1
CKNFAST_NO_ACCELERATOR_SLOTS=1
CKNFAST_NO_UNWRAP=1
CKNFAST_OVERRIDE_SECURITY_ASSURANCES=import
```

 PrimeKey Solutions	KCA-EJBCA-migration, nCipher HSM	Sidnr/ Page no 4 (10)
Uppgjort/ Author Tomas Gustavsson	Sekretess/ Confidentiality OPEN	
Godkänd/ Authorized	Datum Date 15/05/09	Version 1.1

3.1.2 Signing keys

When KCA is installed it generates keys on the nCipher HSM. These keys are in “native” format and can only be used by KCA. No public key is associated in nCipher.

We start by listing some common information about the HSM:

```
C:\nfast\bin>enquiry
Server:
  enquiry reply flags  none
  enquiry reply level  Six
  serial number       B1BB-0EE9-3511
  mode                 operational
  version              2.23.6
  speed index          440
  rec. queue           422..622
  level one flags      Hardware HasTokens
  version string       2.23.6cam6, 2.22.43cam8 built on Oct
13 2006 16:16:12
  checked in           00000000431dca98 Tue Sep 06 18:58:00
2005
  level two flags      none
  max. write size      8192
  level three flags    KeyStorage
  level four flags     OrderlyClearUnit HasRTC HasNVRAM
HasNSOPermsCmd ServerHasPollCmds FastPollSlotList HasSEE
HasKLF HasShareACL HasFeatureEnable Ha sFileOp HasPCIPush
HasKernelInterface HasLongJobs ServerHasLongJobs
AESModuleKeys NTokenCmds LongJobsPreferred
  module type code     0
  product name         nFast server
  device name
  EnquirySix version   4
  impath kx groups
  feature ctrl flags   none
  features enabled     none
  version serial       0

Module #1:
  enquiry reply flags  none
  enquiry reply level  Six
  serial number       B1BB-0EE9-3511
  mode                 operational
  version              2.22.43
  speed index          440
  rec. queue           19..152
```

```

level one flags      Hardware HasTokens
version string      2.22.43cam8 built on Oct 13 2006
16:16:12
checked in          00000000452f6a4d Fri Oct 13 12:28:29
2006
level two flags     none
max. write size    8192
level three flags   KeyStorage
level four flags    OrderlyClearUnit HasRTC HasNVRAM
HasNSOPermsCmd ServerHasPollCmds FastPollSlotList HasSEE
HasKLF HasShareACL HasFeatureEnable Ha sFileOp HasPCIPush
HasKernelInterface HasLongJobs ServerHasLongJobs
AESModuleKeys NTokenCmds LongJobsPreferred
module type code    7
product name        nC1003P/nC3023P
device name         #1 PCI bus 7 slot 1
EnquirySix version  5
impath kx groups    DHPPrime1024
feature ctrl flags  LongTerm
features enabled    StandardKM
version serial      24
rec. LongJobs queue 18
SEE machine type    PowerPCSXF

```

C:\nfast\bin>nfkminfo.exe

```

World
generation 2
state      0x17270000 Initialised Usable Recovery !
PINRecovery !ExistingClient RTC NVRAM FTO SEEDebug
n_modules  1
hkns0      057731d6c635560900003fed89eba88c5082f2a1
hkm        b08e66b01777fcf34dceb77c0684c8d1a071144e (type
DES3)
hkmwk      1d572201be533ebc89f30fdd8f3fac6ca3395bf0
hkre       b0f5dbebedc703c6acd7685b261f6748bc4a9535
hkra       7368a945efc76505a19092c5115f493716de3172
hkmc       1d1e3fb769837be06e028bc038d1789ced2ac9e1
hkrtc      55f17755cae9ae49a588f154260306deae1f5cc
hkvn       82674f0623407fbaac5a3aaa0fc08346dff34f37
hkdsee     6df83c268c25939c307f61245235a2ac44e487ae
hkfto      cae32a48bd1b9e68e606ae723f8dcb93eb4ee9ab
hknull     1d572201be533ebc89f30fdd8f3fac6ca3395bf0
ex.client  none
k-out-of-n 1/1
other quora m=1 r=1 nv=1 rtc=1 dsee=1 fto=1

```



```
createtime 2007-07-16 14:58:41
nso timeout 10 min
```

Module #1

```
generation 2
state      0x2 Usable
flags      0x0 !ShareTarget
n_slots    2
esn        B1BB-0EE9-3511
hkml       5e43facc6aa39068092762d80a1954bde193b4b
```

Module #1 Slot #0 IC 25

```
generation 1
phystype   SmartCard
slotlistflags 0x2 SupportsAuthentication
state      0x5 Operator
flags      0x10000 Passphrase
shareno    1
shares     LTU(PIN)
error      OK
```

Cardset

```
name       "oper"
k-out-of-n 1/1
flags      NotPersistent
```

PINRecoveryForbidden(disabled) !RemoteEnabled

```
timeout    none
card names ""
hkltu      3126f2b3cf7d9d53e3ba278a081ef471644298f8
gentime    2007-07-16 15:00:07
```

Module #1 Slot #1 IC 0

```
generation 1
phystype   SoftToken
slotlistflags 0x0
state      0x2 Empty
flags      0x0
shareno    0
shares
error      OK
```

No Cardset

No Pre-Loaded Objects

You can figure out which keys exist with the command:

```
C:\nfast\bin>nfkminfo -k
```

```
Key list - 19 keys
```

```

AppName rsa-keon-ca-65      Ident 1184599865281000
AppName rsa-keon-ca-65      Ident 1184599867765000
AppName rsa-keon-ca-65      Ident 1184599888484000
AppName rsa-keon-ca-65      Ident 11845998900
AppName rsa-keon-ca-65      Ident 1184599947937000
AppName rsa-keon-ca-65      Ident 1184599969937000
AppName rsa-keon-ca-65      Ident 1184599971843000
AppName rsa-keon-ca-65      Ident 1184599974609000
AppName rsa-keon-ca-65      Ident 1184599977718000
AppName rsa-keon-ca-65      Ident 1184599980515000
AppName rsa-keon-ca-65      Ident 1184599982765000
AppName rsa-keon-ca-65      Ident 1184599985656000
AppName rsa-keon-ca-65      Ident 1184599987625000
AppName rsa-keon-ca-65      Ident 1184599989140000
AppName rsa-keon-ca-65      Ident 1184599991234000
AppName rsa-keon-ca-65      Ident 1184600151640000
AppName rsa-keon-ca-65      Ident 1184600153609000
AppName rsa-keon-ca-65      Ident 1184659106609000
AppName rsa-keon-ca-65      Ident 1184659187875000

```

It is a lot of keys, and we don't know which key is associated with the CAs signing certificate. We can figure this out with:

```

C:\nfast\bin>pubkey-find.exe c:\kca.pem
input format cert
nCore hash d0196ea2e070315e9e162c28dc5a524bf6380d

unnamed
appname rsa-keon-ca-65
ident 1184659106609000

```

After this we can add the public key:

```
C:\nfast\bin>pubkey-find.exe --augment c:\kca.pem
```

Now we can move the key to be accessible through PKCS#11 instead:

```

C:\nfast\bin>generatekey --retarget --no-verify pkcs11
13:40:30 WARNING: nfgk_debug_output is now deprecated (see
manual).
from-application: Source application? (jcecs, pkcs11, rsa-
keon-ca-65)


```

```

[default jceesp] > rsa-keon-ca-65
from-ident: Source key identifier? (1184599865281000,
1184599867765000,
                                     1184599888484000,
11845998900,
                                     1184599947937000,
1184599969937000,
                                     1184599971843000,
1184599974609000,
                                     1184599977718000,
1184599980515000,
                                     1184599982765000,
1184599985656000,
                                     1184599987625000,
1184599989140000,
                                     1184599991234000,
1184600151640000,
                                     1184600153609000,
1184659106609000,
                                     1184659187875000)
[1184599865281000]
> 1184659106609000
plainname: Key name? [] >
ERROR: plainname: key name unspecified
plainname: Key name? [] > kcaSign
key generation parameters:
  operation      Operation to perform      retarget
  application    Application                  pkcs11
  slot          Slot to read cards from    0
  verify        Verify security of key     no
  from-application Source application        rsa-keon-ca-65
  from-ident    Source key identifier      1184659106609000
  plainname     Key name                   kcaSign
*****
* WARNING: will not verify the security of the key *
*****

Loading `oper':
  Module 1: 0 cards of 1 read
  Module 1 slot 0: `oper' #1
  Module 1 slot 0:- passphrase supplied - reading card
Card reading complete.

Key successfully retargetted.
    
```

 PrimeKey Solutions Uppgjort/ Author Tomas Gustavsson Godkänd/ Authorized	KCA-EJBCA-migration, nCipher HSM	Sidnr/ Page no 9 (10)
		Sekretess/ Confidentiality OPEN
	Datum Date 15/05/09	Version 1.1

Path to key:

```
c:\nfast\kmdata\local\key_pkcs11_uc3126f2b3cf7d9d53e3ba278a081ef471644298f8-628484d430c6c0502fdb1a520fb84b9dc73c8372
```

We must also associate a certificate with the key to be able to access it through the Java PKCS#11 provider. We will use the real certificate:

```
C:\nfast\bin>ckcerttool.exe -c oper -f c:\ca\kca.pem -k uc3126f2b3cf7d9d53e3ba278a081ef471644298f8-628484d430c6c0502fdb1a520fb84b9dc73c8372 -L kcaSign
```

```
Certificate found, processing...
```

```
Please enter the passphrase for "oper" token (No echo set).
```

```
Passphrase:
```

```
Certificate successfully imported.
```

```
Run ckslist to view your certificate object.
```

```
OK
```

Now we want to generate two additional keys for EJBCA. One for encryption (internal encryption for key recovery etc) and one for HSM tests. This generation should be done as described in the EJBCA User Guide though, using the regular EJBCA tools.

Generate one RSA key of length 2048 bits with alias kcaDefault and one RSA key of length 1024 with alias kceTest.

We use the EJBCA tools to get automatic association between the private key and a (dummy) certificate, so it can be used through the Java PKCS#11 provider.

3.2 Import CA

After we have done *retarget* of the keys for a Root CA and a Sub CA we can import the CA certificate in EJBCA.

KCA does not use the same DN order as EJBCA does by default, so if you want issued certificate to look exactly the same you should uncheck the checkbox *Use LDAP DN order* in EJBCA after the installation.


First install EJBCA as normal with an AdminCA. After this the KCA CAs can be imported using command line tools.

```
bin/ejbca.sh ca importca TestKCARootCA
org.ejbca.core.model.ca.catoken.PKCS11CAToken foo123
rootca.properties TestKCARootCA.pem
```

rootca.properties contains:

```
defaultKey rootDefault
```

```
certSignKey rootSign
```

 PrimeKey Solutions	KCA-EJBCA-migration, nCipher HSM	Sidnr/ Page no 10 (10)
Uppgjort/ Author Tomas Gustavsson	Sekretess/ Confidentiality OPEN	
Godkänd/ Authorized	Datum Date 15/05/09	Version 1.1

```
crlSignKey rootSign
testKey rootTest
pin foo123
sharedLibrary /opt/nfast/toolkits/pkcs11/libcknfast.so
```

Do the same thing with SubCA:

```
bin/ejbca.sh ca importca TestKCASubCA
org.ejbca.core.model.ca.catoken.PKCS11CAToken foo123
subca.properties TestKCASubCA-chain.pem
```

subca.properties contains:

```
defaultKey subDefault
certSignKey subSign
crlSignKey subSign
testKey subTest
pin foo123
sharedLibrary /opt/nfast/toolkits/pkcs11/libcknfast.so
```

TestKCASubCA-chain.pem is created with:

```
cat TestKCASubCA.pem TestKCARootCA.pem > TestKCASubCA-chain.pem
```

After this step we now have a Root CA and a Sub CA using the same signature keys as the KCA.

3.3 Import user certificates

In EJBCA there is a command to import user certificates:

```
bin/ejbca.sh ca importcert kcal foo123 TestKCASubCA ACTIVE
user1.pem
```

The command does not import revocation information, so this should be added in a migration tool.

You can easily write your own migration tool, speeding up the process by using the command line code as a template. It can be found in src/java/org/ejbca/ui/cli/CAImportCertCommand.java.