Prep Notes for the SAFIRE Metadata Signing Key Generation Ceremony

Prerequisites

Install prerequisite software

apt-get install libccid libengine-pkcs11-openssl opensc opensc-pkcs11

Choose passwords

- Security officer PIN - eight byte random value
- User PIN
- Device key encryption key (backup/restore)

Create Device Key Encryption Key share

sc-hsm-tool --create-dkek-share dkek-share-a.pbe

HSM Initialization

These steps need to be completed on each physical HSM

Initialise HSM PKCS15 structure

sc-hsm-tool --initialize --so-pin 3537363231383830 --pin *userpin* --dkek-shares 1 --label SAFIRE-*n*

Where *n* is the HSM number.

Change the PINs

pkcs11-tool -l --login-type so --so-pin 3537363231383830 --change-pin --new-pin *sopin*

Import DKEK share

sc-hsm-tool --import-dkek-share dkek-share-a.pbe

Key generation

These steps only need to be performed on the first HSM, but must be repeated for each key we wish to generate (only the first key shown in most examples)

Generate a metadata signing key

pkcs11-tool -l --pin *userpin* --keypairgen --key-type rsa:2048 --id 1 --label metadata-rsa-1
pkcs11-tool -l --pin *userpin* --keypairgen --key-type EC:prime256v1 --id 3 --label metadata-ecc-1

Generate a self-signed certificate

OpenSSL> engine -t dynamic -pre SO_PATH:/usr/lib/engines/engine_pkcs11.so -pre ID:pkcs11 -pre LIST_ADD:1 -pre LOAD -pre MODULE_PATH:/usr/lib/x86_64-linux-gnu/pkcs11/opensc-pkcs11.so
OpenSSL> req -engine pkcs11 -new -key 1:10 -keyform engine -x509 -days 5479 -sha256 -subj '/C=ZA/O=South African Identity Federation/CN=SAFIRE Metadata Signing' -out metadata-rsa-1.crt

Import the cert onto the HSM

openssl x509 -in metadata-rsa-1.crt -out metadata-rsa-1.der -outform der
pkcs11-tool -l --pin userpin --write-object metadata-rsa-1.der --type cert --id 1 --label metadata-rsa-1

Check PKCS15/get the key reference

pkcs15-tool -D

Find the Key ref field for the metadata private key (or whatever one you're trying to export)

Export/wrap the key

sc-hsm-tool --wrap-key metadata-rsa-1-key.bin --key-reference 1 --pin *userpin*

Key duplication

These steps must be performed on each additional (initialised) HSM to clone the key. It needs to be repeated once per key, using the key-reference from the export.

Import/unwrap the key

sc-hsm-tool --unwrap-key metadata-rsa-1-key.bin --key-reference 1 --pin *userpin*

Backups

DKEK share

Onto a flash stick, write dkek-share-a.pbe
Print out a base64 encoded version of dkek-share-a.pbe

Storage

In safe A, place the additional Nitrokey HSMs
In safe B, place flash stick and Base64 encoded DKEK share
In a password safe, store the DKEK password and SO password