2.1 Generating a fast authentication PKI tree

Starting in HAB v4.1.2 users can use a single SRK key to authenticate the both CSF and IMG contents. This reduces the number of key pair authentications that must occur during the ROM/HAB boot stage, thus providing a faster boot process.

The script hab4_pki_tree.sh is also able to generate a Public Key Infrastructure (PKI) tree which only contains SRK Keys, users should not set the CA flag when generating the SRK certificates.

- Generating 2048-bit fast authentication PKI tree on CST (starting from v3.1.0):

```
  $ ./hab4_pki_tree.sh
  ...
  Do you want to use an existing CA key (y/n)?: n
  Do you want to use Elliptic Curve Cryptography (y/n)?: n
  Enter key length in bits for PKI tree: 2048
  Enter PKI tree duration (years): 5
  How many Super Root Keys should be generated? 4
  Do you want the SRK certificates to have the CA flag set? (y/n)?: n
```

The diagram below illustrate the PKI tree generated:

```
                             +----------+
                             |   CA     |
                             +----------+
                                  |
                                  |
         ---------------------------------------------------
         |               |                 |               |
         |               |                 |               |
         v               v                 v               v
    +---------+      +---------+      +---------+      +---------+
    |  SRK1   |      |  SRK2   |      |  SRK3   |      |  SRK4   |
    +---------+      +---------+      +---------+      +---------+
```

2.2 Generating a SRK Table and SRK Hash

The next step is to generated the SRK Table and its respective SRK Table Hash from the SRK public key certificates created in one of the steps above.

In the HAB architecture, the SRK Table is included in the CSF binary and the SRK Hash is programmed in the SoC SRK_HASH[255:0] fuses.

On the target device during the authentication process the HAB code verify the SRK Table against the SoC SRK_HASH fuses, in case the verification success the root of trust is established and the HAB code can progress with the image authentication.

The srktool can be used for generating the SRK Table and its respective SRK Table Hash.

- Generating SRK Table and SRK Hash in Linux 64-bit machines:

```
  $ ../linux64/bin/srktool -h 4 -t SRK_1_2_3_4_table.bin -e \
    SRK_1_2_3_4_fuse.bin -d sha256 -c \
    SRK1_sha256_2048_65537_v3_ca_crt.pem,\
    SRK2_sha256_2048_65537_v3_ca_crt.pem,\
    SRK3_sha256_2048_65537_v3_ca_crt.pem,\
    SRK4_sha256_2048_65537_v3_ca_crt.pem
```

The SRK_1_2_3_4_table.bin and SRK_1_2_3_4_fuse.bin files can be used in further steps as explained in HAB guides available under doc/imx/habv4/guides/ directory.

References:

[1] CST: i.MX High Assurance Boot Reference Code Signing Tool.
[2] AN4581: "i.MX Secure Boot on HABv4 Supported Devices"
[3] AN12056: "Encrypted Boot on HABv4 and CAAM Enabled Devices"