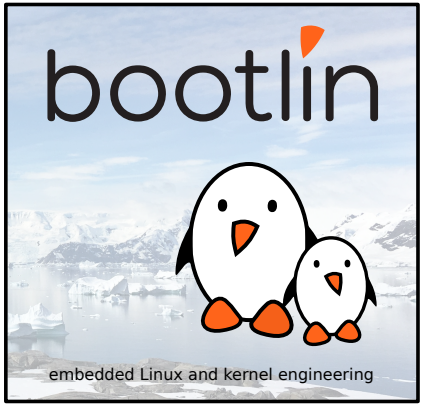
PKCS#11 with OP-TEE

Thomas Perrot
thomas.perrot@bootlin.com



Who is speaking ?

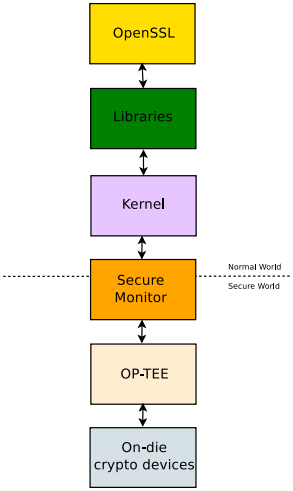
- Thomas Perrot
- Embedded Linux and kernel engineer at Bootlin
- Joined in 2020
- Embedded Linux engineer and trainer
- Open-source contributor
- Based in Toulouse, France

Agenda

- Introduction
  - The Goal
  - PKCS#11
  - OP-TEE
  - i.MX8
- Implementation
  - List required software parts
  - Customize the distribution
  - Perform functional tests
  - Generate a key

Introduction

The goal

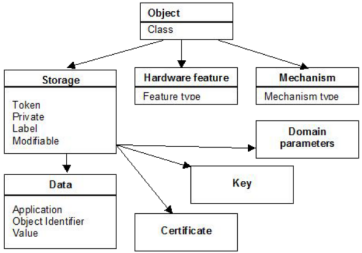


Perform cryptographic operations with:

- OpenSSL
- Isolate assets
- Control access
- On-die secure enclave
- Without HSM, TPM...

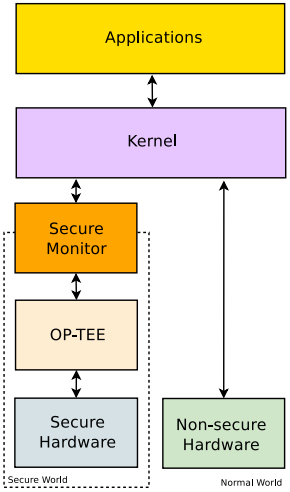
PKCS#11

- Public-Key Cryptographic Standards (PKCS)
- Defines a platform independent API (Cryptoki)
- Can be used with HSM, TPM, Secure element...
- Supported by OpenSSL, GnuTLS, OpenSSH...



C_Initialize() C_Finalize() C_GetInfo() C_GetFunctionList() C_GetSlotList() C_GetSlotInfo() C_GetTokenInfo() C_GetMechanismList() C_GetMechanismInfo() C_InitToken() C_InitPIN() C_SetPIN() C_OpenSession() C_CloseSession() C_CloseAllSessions() C_GetSessionInfo() CK_C_Login() C_Login() C_Logout() C_CreateObject() C_CopyObject() C_DestroyObject() C_GetObjectSize() C_GetAttributeValue() C_SetAttributeValue() C_FindObjectsInit() C_FindObjects() C_FindObjectsFinal() C_EncryptInit() C_Encrypt() C_EncryptUpdate() C_EncryptFinal() C_DecryptInit() C_Decrypt() C_DecryptUpdate() C_DecryptFinal() C_DigestInit() C_Digest() C_DigestUpdate() C_DigestKey() C_DigestFinal() C_SignInit() C_Sign() C_SignUpdate() C_SignFinal() C_VerifyInit() C_Verify() C_VerifyUpdate() C_VerifyFinal() C_GenerateKey() C_GenerateKeyPair() C_WrapKey() C_UnwrapKey() C_DeriveKey() C_SeedRandom() C_GenerateRandom()

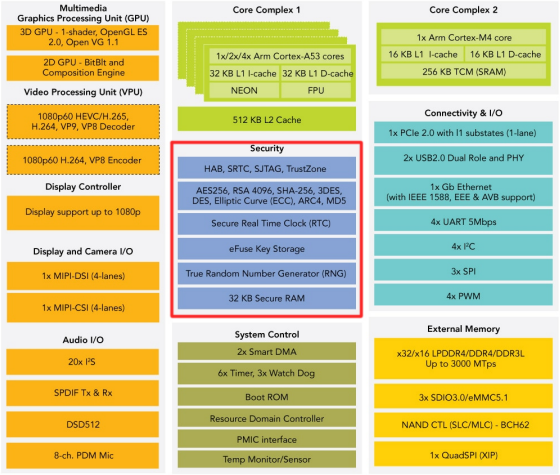
OP-TEE



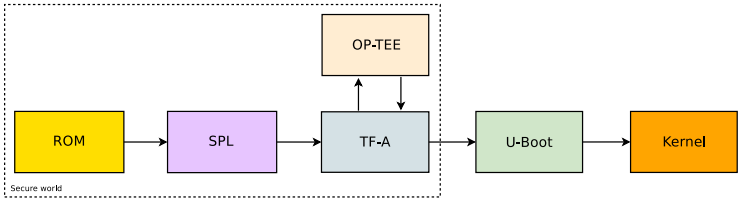
- Is a Trusted Execution Environment (TEE)
- Based on hardware isolation (ARM TrustZone)
- Protection against REE security vulnerabilities
- Communicates through Secure Monitor Call (SMC)
- Executes Trusted Applications (TAs)

i.MX8 features

- ARM TrustZone
- HABv4 or AHAB
- Hardware Tampering
- One Time Programmable (OTP)
- Secure Non-Volatile Storage (SNVS)
- Cryptographic Accelerator and Assurance Module (CAAM)
- Random Number Generator (RNG)



i.MX8 Boot sequence



- The boot sequence is platform dependent, here:
  - SPL loads and executes TF-A
  - TF-A and OP-TEE are executed from the **Secure world**
  - Then TF-A starts U-Boot from the **Normal world**
  - Bootloader stages can be signed

Implementation

Required software parts

- OpenSSL
- PKCS#11 library and engine (OpenSC)
- TEE Client library (libckteec)
- OP-TEE client
- OP-TEE PKCS#11 TA
- PKCS#11 tools

Customize the distribution

- Enable the support of OP-TEE

```
DISTRO_FEATURES:append = " optee"
```

- BSP is automatically customized, thanks to *COMBINED_FEATURES*
  - Build the SPL
  - Build OPTEE-OS wit the right platform flavor
  - Update TF-A variables

```
NEED_BL32=yes BL32_BASE=0xbe000000 SPD=opteed
```

- Install rootfs packages

```
IMAGE_INSTALL:append = " optee-test libp11 opensc openssl-bin"
```

Customize the i.MX8 BSP (optional)

- To use NXP BSP instead of the mainline

```
IMX_DEFAULT_BSP = "nxp"
```

- Enable the PKCS#11 support in OP-TEE

```
CFG_PKCS11_TA=y
```

- Customize the OP-TEE features

```
CFG_TEE_TA_LOG_LEVEL=4 CFG_TEE_CORE_LOG_LEVEL=4
CFG_UNWIND=y
TA_PUBLIC_KEY=ta_pub_key.pem
```

- Modify the default platform configuration

```
CFG_UART_BASE=UART2_BASE [...]
```

Check that OP-TEE is functioning

```
dmesg|grep optee
[    1.807123] optee: probing for conduit method.
[    1.811603] optee: revision 3.10 (87956c34)
[    1.813114] optee: dynamic shared memory is enabled
[    1.822546] optee: initialized driver
```

```
ls -lh /dev/tee*
crw------- 1 root root 246,  0 Sep 20 10:44 /dev/tee0
crw------- 1 root root 246, 16 Sep 20 10:44 /dev/teepriv0
```

```
ls -lh /lib/optee_armtz/|grep fd02c9da-306c-48c7-a49c-bbd827ae86ee
-r--r--r--    1 root     root      155.1K Mar  9 12:34 fd02c9da-306c-48c7-a49c-bbd827ae86ee.ta
```

```
xtest
[...]
35546 subtests of which 0 failed
126 test cases of which 0 failed
0 test cases were skipped
TEE test application done!
```

Check that OP-TEE provides PKCS#11

```
pkcs11-tool --show-info --module /usr/lib/libckteec.so.0
Cryptoki version 2.40
Manufacturer     Linaro
Library          OP-TEE PKCS11 Cryptoki library (ver 0.1)
Using slot 0 with a present token (0x0)
```

Create a PKCS#11 token

```
pkcs11-tool --module /usr/lib/libckteec.so.0 --init-token --label testtoken --so-pin 12341234
pkcs11-tool --module /usr/lib/libckteec.so.0 --label testtoken --login --so-pin 12341234 \
    --init-pin --pin 12341234
pkcs11-tool --module /usr/lib/libckteec.so.0 --label testtoken --login --pin \
    12341234 --keypairgen --label testkey --key-type rsa:2048
```

Show PKCS#11 information

```
pkcs11-tool --list-slots --module /usr/lib/libckteec.so.0
Available slots:
Slot 0 (0x0): OP-TEE PKCS11 TA - TEE UUID 2b9f2e53-bff5-5239-986c-52530dda62db
  token state:   uninitialized
Slot 1 (0x1): OP-TEE PKCS11 TA - TEE UUID 2b9f2e53-bff5-5239-986c-52530dda62db
  token state:   uninitialized
Slot 2 (0x2): OP-TEE PKCS11 TA - TEE UUID 2b9f2e53-bff5-5239-986c-52530dda62db
  token label        : PKCS11 TA test token
  token manufacturer : Linaro
  token model        : OP-TEE TA
  token flags        : login required, rng, token initialized, PIN initialized
  hardware version   : 0.0
  firmware version   : 0.1
  serial num         : 0000000000000002
  pin min/max        : 4/128
```

Generate an OpenSSL key

```
openssl
OpenSSL> engine dynamic -pre SO_PATH:/usr/lib/engines-1.1/pkcs11.so -pre ID:pkcs11 \
    -pre LIST_ADD:1 -pre LOAD -pre MODULE_PATH:/usr/lib/engines-1.1/libckteec.so.0.1.0
(dynamic) Dynamic engine loading support
[Success]: SO_PATH:/usr/lib/engines-1.1/pkcs11.so
[Success]: ID:pkcs11
[Success]: LIST_ADD:1
[Success]: LOAD
[Success]: MODULE_PATH:/usr/lib/engines-1.1/libckteec.so.0.1.0
Loaded: (pkcs11) pkcs11 engine
OpenSSL> genrsa -engine pkcs11 -out priv_key.pem 2048
engine "pkcs11" set.
Generating RSA private key, 2048 bit long modulus (2 primes)
.....................................................................+++++
.....................................+++++
e is 65537 (0x010001)
```

Questions? Suggestions? Comments?

Thomas Perrot
thomas.perrot@bootlin.com

https://bootlin.com/pub/conferences/2022/lee/perrot-optee-pkcs11/