Leveraging OP-TEE as a generic HSM via PKCS#11 for secure OTA

Ricardo Salveti <ricardo@foundries.io>

Principal Engineer

Introduction

Contents

1. IoT Security Needs
2. Hardware Security Module
3. PKCS#11
4. ARM TrustZone / OP-TEE
5. OP-TEE Secure Key Services
6. Secure OTA

IoT Security Needs

- Device security
  - Trusted and authentic software / firmware execution
  - Secure and verified boot

- Network security
  - Data integrity
  - Authentication via unique device identity
  - Data communication protection

Hardware Security Module

FOUNDRIES.IO

- Physical computing device for protecting and managing keys
- Key materials and cryptographic operations hard to tamper
- Allows importing, generating, deriving keys and cipher
- Encryption, decryption, sign and verify operations
- Used via platform-independent standards such as PKCS#11

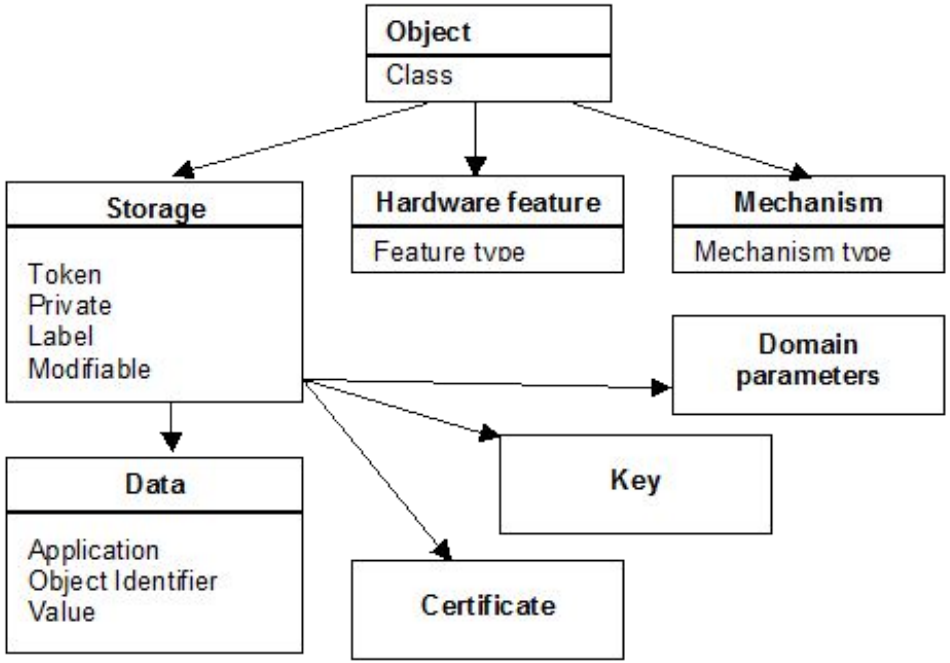
PKCS#11 - Cryptoki

- High-level platform-independent API to cryptographic devices
- The 'standard' API for smartcards and HSM
- Supported by OpenSSL, GnuTLS, wolfSSL and many others
- Great tooling support, including complete software implementations
- Simple API and data managed in an object based approach
- Latest specification release is v2.40, with v3.0 to be released soon

PKCS#11 - Cryptoki

FOUNDRIES.IO

```
C_Initialize()
C_GetInfo()
C_GenerateKey()
C_Encrypt()
C_Decrypt()
C_Digest()
C_Sign()
C_Verify()
C_Finalize()
```



Why not simply use HSM on IoT?

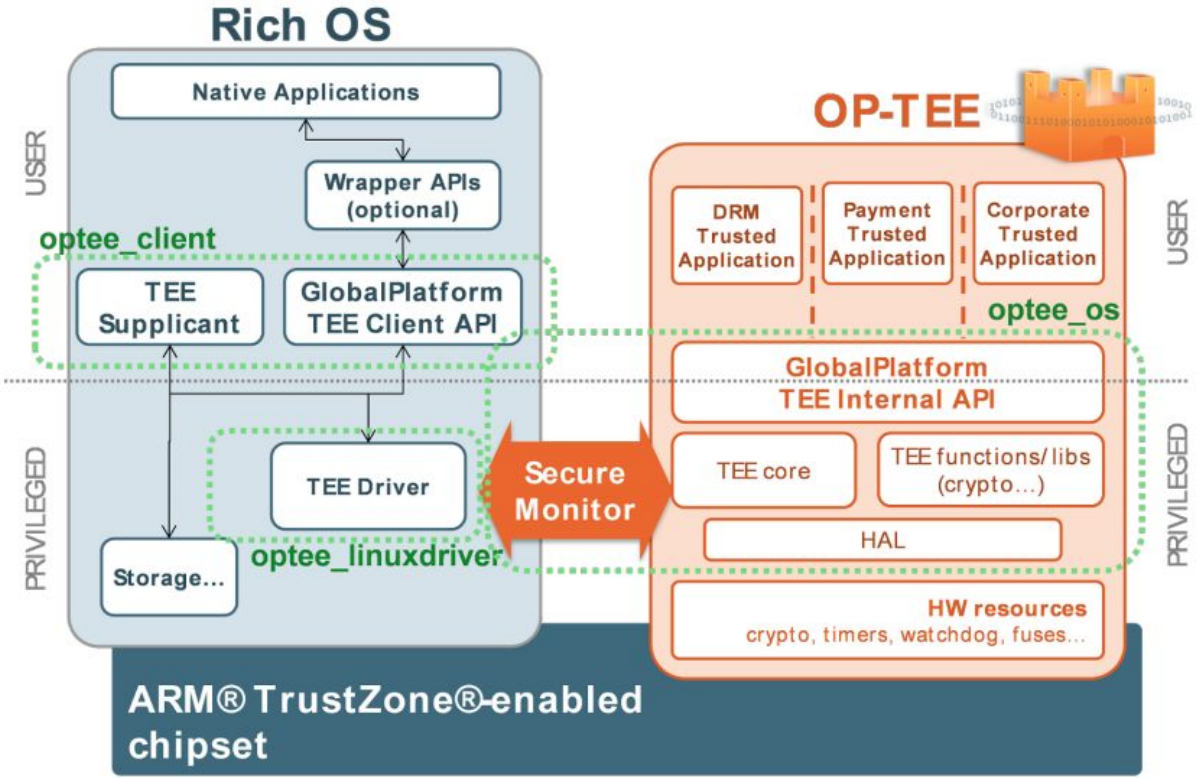
FOUNDRIES.IO

- Cost
  - Several IoT devices and applications are cost constrained
  - TPM as alternative, but known to be complex to manage
- Proprietary implementation
  - Unable to review the software implementation
  - Bug fixes can only be provided by the module vendor

- ARM-based devices are the majority of the IoT space
  - Possibility to use Trusted Execution Environment as a HSM
  - Open source secure OS available - OP-TEE

ARM TrustZone / OP-TEE

FOUNDRIES.IO



OP-TEE

- Open Source Trusted Execution Environment maintained by Linaro
- Uses GlobalPlatform APIs for the Client and Core
- TEE Client API
  - Used by Linux client to communicate to a Trusted Application
  - Functions calls exposed via commandID, up to 4 parameters
  - Shared memory
- TEE Internal Core API
  - Trusted Storage API for Data and Keys
  - Cryptographic, Time and Arithmetical Operations API
- TEE API satisfies the need of software-based HSM

OP-TEE Client API

FOUNDRIES.IO

```
TEEC_Result TEEC_InitializeContext(...)
void TEEC_FinalizeContext(...)
TEEC_Result TEEC_OpenSession (...)
void TEEC_CloseSession (...)
TEEC_Result TEEC_InvokeCommand(
        TEEC_Session* session,
        uint32_t commandID,
        TEEC_Operation* operation,
        uint32_t* returnOrigin)
```

OP-TEE Secure Key Services

- Open source implementation of PKCS#11 services as TA
- Started by Etienne Carriere <etienne.carriere@linaro.org>
- RFC available at https://github.com/OP-TEE/optee_os/pull/2732
- Libsks as PKCS#11 client library
- SKS TA implementing the HSM side of PKCS#11
  - Responsible for managing and operating keys
  - Uses TEE Internal Core API for secure storage
- Foundries.IO tree available at https://github.com/foundriesio/optee-sks
  - Upstreaming in process

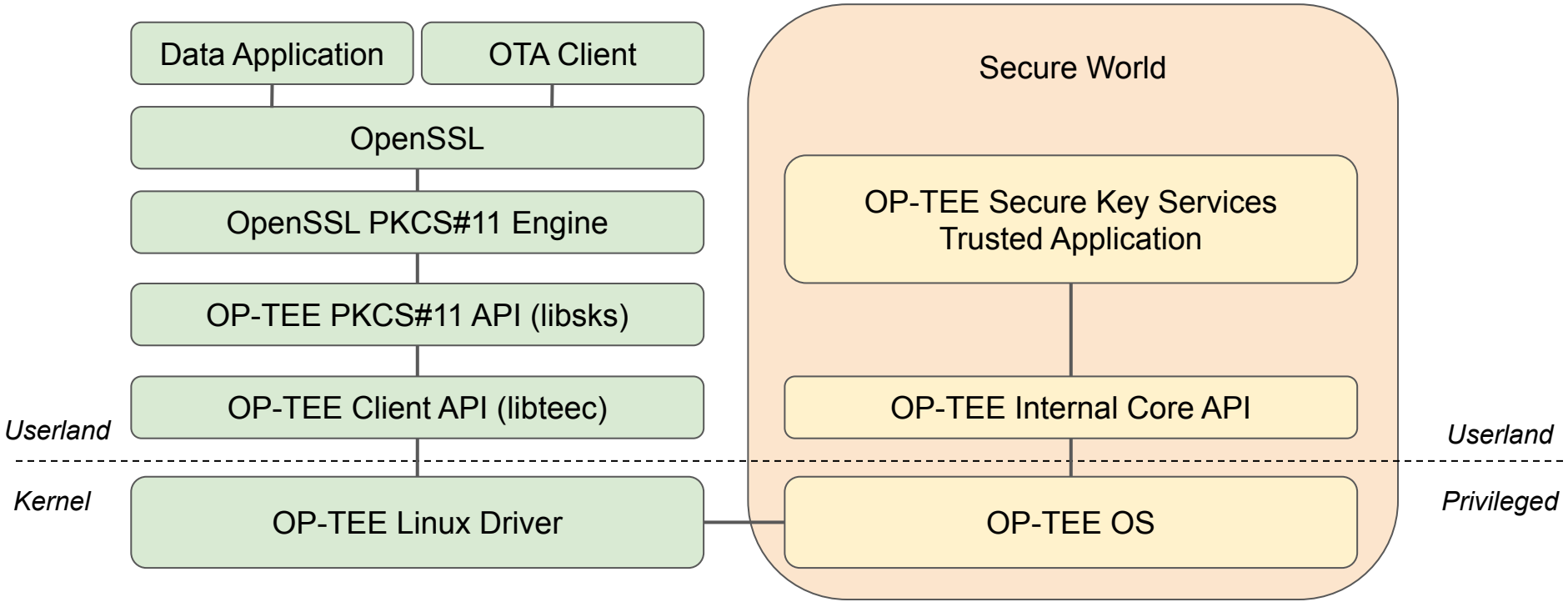
OP-TEE Secure Key Services API

- PKCS#11 function calls mapped to TEE commandID
- Function parameters mapped to TEE command parameters
- Shared memory for both data and serialization

`C_InitToken()`	`SKS_CMD_CK_INIT_TOKEN`
`C_EncryptInit()`	`SKS_CMD_ENCRYPT_INIT`
`C_EncryptUpdate()`	`SKS_CMD_ENCRYPT_UPDATE`
`C_EncryptFinal()`	`SKS_CMD_ENCRYPT_FINAL`
`C_CloseSession()`	`SKS_CMD_CK_CLOSE_SESSION`

OP-TEE Secure Key Services Usage

FOUNDRIES.IO



rsalveti@evapro: ~ (mosh-client)

```
rsalveti@evaxps:/tmp/qemu$ qemu-system-aarch64 -device virtio-net-device,netdev=net0,mac=52:54:00:12:35:02 -netdev user,id=net0,hostfwd=tcp::2222-:22 -drive if=virtio
,file=/tmp/qemu/lmp-gateway-image-qemuarm64.img,format=raw -no-reboot -no-acpi -bios bl1.bin -d unimp -semihosting-config enable,target=native -nographic -machine vir
t,secure=on -cpu cortex-a57 -m 1057 -smp 2 -S -serial telnet:127.0.0.1:4444,server,nowait -serial telnet:127.0.0.1:4445,server,nowait
QEMU 4.0.0 monitor - type 'help' for more information
(qemu) c
(qemu)
```

rsalveti@evapro: ~ (mosh-client)

```
NOTICE:  Booting Trusted Firmware
NOTICE:  BL1: v2.1(release):v2.1-347-ge9e74aa4c
NOTICE:  BL1: Built : 00:00:00, Jan  1 1970
WARNING: Firmware Image Package header check failed.
NOTICE:  BL1: Booting BL2
NOTICE:  BL2: v2.1(release):v2.1-347-ge9e74aa4c
NOTICE:  BL2: Built : 00:00:00, Jan  1 1970
WARNING: Firmware Image Package header check failed.
WARNING: Firmware Image Package header check failed.
WARNING: Firmware Image Package header check failed.
WARNING: Firmware Image Package header check failed.
NOTICE:  BL1: Booting BL31
```

rsalveti@evapro: ~ (mosh-client)

```
I/TC: Switching console to device: /pl011@9040000
I/TC: OP-TEE version: 3.6.0-dev #1 Wed Jul 31 23:25:34 UTC 2019 aarch64
D/TC:0 0 check_ta_store:687 TA store: "Secure Storage TA"
D/TC:0 0 check_ta_store:687 TA store: "REE [buffered]"
D/TC:0 0 mobj_mapped_shm_init:446 Shared memory address range: f200000, 11200000
I/TC: Initialized
D/TC:0 0 init_primary_helper:1096 Primary CPU switching to normal world boot
D/TC:1   generic_boot_cpu_on_handler:1135 cpu 1: a0 0x0
D/TC:1   select_vector:1018 SMCCC_ARCH_WORKAROUND_1 (0x80008000) available
D/TC:1   select_vector:1020 SMC Workaround for CVE-2017-5715 used
D/TC:1   init_secondary_helper:1120 Secondary CPU Switching to normal world boot
D/TC:1   tee_entry_exchange_capabilities:101 Dynamic shared memory is enabled
D/TC:1 0 core_mmu_entry_to_finer_grained:794 xlat tables used 5 / 5
D/TC:? 0 tee_ta_init_pseudo_ta_session:280 Lookup pseudo TA fd02c9da-306c-48c7-a49c-bbd827ae86ee
D/TC:? 0 load_ldelf:744 ldelf load address 0x40006000
F/TC:? 0 trace_syscall:127 syscall #1 (syscall_log)
```

rsalveti@evapro: ~ (mosh-client)

```
Current directory: /sysroot/home/root
Seeding random number generator from /dev/urandom...
... seeding complete in 0.00114s
Use existing SQL storage: "/var/sota/sql.db"
meta with role root in repo director not present in db
meta with role root in repo image not present in db
Slot manufacturer......: Linaro
Slot description.......: 94e9ab89-4c43-56ea-8b35-45dc07226830
Slot token label.......: aktualizr
Slot token manufacturer: Linaro
Slot token model.......: OP-TEE SKS TA
Slot token serialnr....: 0000000000000000
Loading PKCS#11 engine library: /usr/lib/engines-1.1/pkcs11.so
Checking if device is provisioned...
* set default crypto engine 'pkcs11'
... provisioned OK
put request body:
{
        "capabilities" :
        {
                "smp" : "Symmetric Multi-Processing"
        },
        "claimed" : true,
        "class" : "system",
        "description" : "Computer",
        "id" : "qemuarm64",
        "width" : 64
}

*   Trying 35.194.54.53:8443...
```

rsalveti@evapro: ~ (mosh-client)

```
D/TA:  TA_InvokeCommandEntryPoint:357 SKS TA exit: SKS_CMD_SIGN_ONESHOT rc 0x00000000/OK
```

rsalveti@evapro: ~ (mosh-client)

```
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Server: nginx/1.13.7
< Date: Sat, 03 Aug 2019 15:14:48 GMT
< Content-Type: application/json
< Content-Length: 994
< Connection: keep-alive
< x-ats-version: director/0.7.0-7-g8cbbcc3
<
* Connection #0 to host ota-ce.foundries.io left intact
response http code: 200
response: {"signatures":[{"keyid":"a045d63a8e943c50b50bf790ff0d5e75a398b48dd07285e1630c9cc827ff3c98","method":"rsassa-pss-sha256","sig":"UxN8FEL1VAl1wKcrDiECtJjSCXwDSa6EZN3Fh5E+Qwu3IfB0pWtnHXynf4pGKrbuCUZ/fgPyz687OQzlnZegoRPUBVbSB541/AiktHcx9efHGDJ5YxjVKEbrXTXoxKe8uNsxopUD194pQ0o+kbgcJwUgJTuiBEhGsH7knZi4SNbHPQntib7RzjuV7XhmyQ5djpApbm+NtDsjKv17wcQGLduqMyyuBFTpMNv/BCG8z/GpxeSt2wrCoKVXYlH3pQNstFGU0XzbkVc7GFruuuhwj+m3MxvWJXkiXv6m07XoaB+3DTpG6mfckSTmEtfcshSk600/7uNheSLkUYhR8I4rkg==","type":"Targets","expires":"2019-09-01T15:16:09Z","targets":{"qemuarm64-lmp-6":{"hashes":{"sha256":"2eb4ce30b764d3e5400c7200a9dbe9b93484c2eec61ad4600de481dd125ca668"},"length":0,"custom":{"ecuIdentifier":"3c31d2f610f49c507ca13afaaecedfb05fefed1631d0391198f5e9e7c859375e","hardwareId":"qemuarm64","ecuIdentifiers":{"3c31d2f610f49c507ca13afaaecedfb05fefed1631d0391198f5e9e7c859375e":{"hardwareId":"qemuarm64"}}}}},"version":1,"custom":{"correlationId":"urn:here-ota:mtu:3e24ad10-ad9c-4f56-96fe-258635f84ed0"}}}
No new updates found in Uptane metadata.
```

rsalveti@evapro: ~ (mosh-client)

```
F/TC:? 0 trace_syscall:127 syscall #31 (syscall_cryp_obj_copy)
F/TC:? 0 trace_syscall:127 syscall #1 (syscall_log)
I/TA: SKSs1: init processing ECDSA SIGN
F/TC:? 0 trace_syscall:127 syscall #1 (syscall_log)
D/TA:  TA_InvokeCommandEntryPoint:357 SKS TA exit: SKS_CMD_SIGN_INIT rc 0x00000000/OK
F/TC:? 0 trace_syscall:127 syscall #1 (syscall_log)
D/TA:  TA_InvokeCommandEntryPoint:160 SKS_CMD_SIGN_ONESHOT ctrl 4@0x40060fb0, in 64@0x4005fdc0, out 72@0x4005e420
F/TC:? 0 trace_syscall:127 syscall #39 (syscall_asymm_operate)
F/TC:? 0 trace_syscall:127 syscall #1 (syscall_log)
I/TA: SKSs1: processing ECDSA SIGN
F/TC:? 0 trace_syscall:127 syscall #17 (syscall_cryp_state_free)
F/TC:? 0 trace_syscall:127 syscall #1 (syscall_log)
D/TA:  TA_InvokeCommandEntryPoint:357 SKS TA exit: SKS_CMD_SIGN_ONESHOT rc 0x00000000/OK
```

OP-TEE SKS Next Steps

- Improve testing coverage via optee_test
- Extend support for additional mechanisms
- Upstream into OP-TEE (available as part of the main project)
- Add support to PKCS#11 v3.0

References

- OP-TEE official documentation: https://optee.readthedocs.io
  - Several useful presentations available
- OASIS PKCS#11 TC: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=pkcs11
- OP-TEE SKS RFC: https://github.com/OP-TEE/optee_os/pull/2732
- Aktualizr: https://github.com/advancedtelematic/aktualizr/tree/master/docs

Thank You!

FOUNDRIES.IO