Using the TPM
It’s Not Rocket Science (Anymore)
Hello World!

Johannes Holland
Embedded Software Developer
@Infineon Technologies
github.com/joholl

Peter Huewe
Principal Engineer
@Infineon Technologies
github.com/peterhuewe
@peterhuewe
TPM – Why Even Bother?
TPM – Why Even Bother?
TPM – Why Even Bother?
TPM – Why Even Bother?
TPM – Why Even Bother?
TPM – Why Even Bother?

- Trusted Platform Module
  - **Open Standard**
  - Open Source Ecosystem
  - In virtually **every** consumer device!

**Trusted Platform Module**

- Powerful Authorization Model
- Platform Configuration Registers
- Secure Key Storage
- True Random Number Generator
Path to Power – TPM2 Software Stack

I want to wield the power of a TPM

Compatible with TCG Standards

Community driven

Multi-Platform

You name it

Easy to use (since FAPI)

Embedded ready

tpm2-software.github.io *

*no, we do not have a fancy name for the project and ecosystem
I want to use the TPM...
I want to use the TPM...

Uses OpenSSL

yes

tpm2-tss-engine
TPM – But How?

› Engine for OpenSSL: tpm2-tss-engine

› Create key

```
$ tpm2tss-genkey -a ecdsa mykey
$ openssl pkeyutl -engine tpm2tss -keyform engine -inkey mykey -sign -in mydata -out mysig
$ openssl pkeyutl -engine tpm2tss -keyform engine -inkey mykey -verify -in mydata -sigfile mysig
```

› Export the public key

```
$ openssl ec -engine tpm2tss -inform engine -in mykey -pubout -outform pem -out mykey.pub
```

› TLS Server? Sure!

```
$ ./tpm2tss-genkey -a rsa rsa.tss
$ openssl req -new -x509 -engine tpm2tss -key rsa.tss -keyform engine -out rsa.crt
$ openssl s_server -cert rsa.crt -key rsa.tss -keyform engine -engine tpm2tss -accept 8443
```
TPM – But How?

I want to use the TPM...

Uses OpenSSL

yes → tpm2-tss-engine

no
TPM – But How?

I want to use the TPM...

- Uses OpenSSL
  - yes: tpm2-tss-engine
  - no:
    - Uses PKCS #11
      - yes: tpm2-pkcs11
TPM – But How?

› PKCS #11 – Cryptoki
  – API for Smart Cards & other crypto tokens

› Supported by many widely-used applications
  – OpenSSH
  – OpenVPN
  – ...

› See the docs for examples: https://github.com/tpm2-software/tpm2-pkcs11/tree/master/docs

› If you integrate the TPM, tell us!
TPM – But How?

I want to use the TPM...

Uses OpenSSL

yes → tpm2-tss-engine

no

Uses PKCS #11

yes → tpm2-pkcs11

no

TPM2 Software Stack (TSS)
TPM – But How?

TPM2 Software Stack (TSS)

Language?

C

tpm2-tss
TPM – But How?

TPM2 Software Stack (TSS)

Language?

C

Rust bindings: tss-sapi, tss-esapi

Python bindings: tpm2-pytss

SWIG bindings: tpm2-swig

tpm2-tss
TPM – But How?

- **tpm2-tss**
  - GNU/Linux Filesystem? [yes | no]
    - yes: TSS Feature API: tpm2-fapi
    - no: Heap, Crypto Lib? [yes | no]
      - yes: TSS Enhanced System API: tpm2-esapi
      - no: Bare min. Resources? [yes | no]
        - yes: TSS System API: tpm2-sapi
        - no: Rocket Science!

Copyright © Infineon Technologies AG 2020. All rights reserved.
Install tpm2-tss (v2.4.0 or later)

Configure the TSS:

Downloads:

/etc/tpm2-tss/fapi-config.json

```json
{
    "profile_name": "P_ECCP256SHA256",
    "profile_dir": "/etc/tpm2-tss/fapi-profiles/",
    "user_dir": "~/tpm2-tss/user/keystore",
    "system_dir": "~/tpm2-tss/system/keystore",
    "tcti": "",
    "system_pcrs": [],
    "log_dir": "~/tpm2-tss/eventlog/"
}
```
FAPI – Hello World!

› A word about profiles...

... don’t worry!

```
/etc/tpm2-tss/fapi-profiles/P_ECCP256SHA256.json
{
    "type": "TPM2_ALG_ECC",
    "nameAlg": "TPM2_ALG_SHA256",
    "srk_template": "system,restricted,decrypt,0x81000001",
    "srk_description": "Storage root key SRK",
    "srk_persistent": 0,
    "ek_template": "system,restricted,decrypt",
    "ek_description": "Endorsement key EK",
    "ecc_signing_scheme": {
        "scheme": "TPM2_ALG_ECDSA",
        "details": {
            "hashAlg": "TPM2_ALG_SHA256"
        }
    },
}
...```
>Hello World!

- Use your tools!
  
  ... seriously, they're great for debugging!

- Install tpm2-tools

- Provision the TPM and the host
  
  $ tss2_provision

  WARNING:fapi:[...]
  Directory /home/alarm/tpm2-tss/eventlog/ does not exist, creating

  WARNING:fapi:[...]
  Directory /home/alarm/tpm2-tss/user/keystore does not exist, creating

  WARNING:fapi:[...]
  Directory /home/alarm/tpm2-tss/system/keystore/policy does not exist, creating

  $ tss2_list

  /P_ECCP256SHA256/LOCKOUT:
  /P_ECCP256SHA256/HE:
  /P_ECCP256SHA256/HE/EK:
  /P_ECCP256SHA256/HN:
  /P_ECCP256SHA256/HS:
  /P_ECCP256SHA256/HS/SRK

- List all objects (keys, policies, ...)

Copyright © Infineon Technologies AG 2020. All rights reserved.
FAPI – Hello World!

› Create a key

```sh
$ tss2_createkey --path "/P_ECCP256SHA256/HS/SRK/mySigningKey" --type "noDa,sign"

New password:
Re-enter new password:
```

› Sign...

```sh
$ openssl dgst -sha256 -binary -out myDigest.bin firmware.img
$ tss2_sign --keyPath "/P_ECCP256SHA256/HS/SRK/mySigningKey" \
   --digest myDigest.bin \
   --publicKey mySigningKey_pub.pem \
   --signature mySignature.bin
```

› ... and verify

```sh
$ openssl dgst -verify mySigningKey_pub.pem -signature mySignature.bin firmware.img
Verified OK
```
#include <tss2/tss2_fapi.h>
#include <string.h>
#include <stdio.h>

int main(){
    TSS2_RC r = 0;
    size_t signatureSize = 0;
    uint8_t *signature;
    char *publicKey;
    size_t digestSize = 32;
    uint8_t digest[32] = {0x67, ..., 0x3e};

    FAPI_CONTEXT *fapi_context;
    r = Fapi_Initialize(&fapi_context, NULL);
    if (r != TSS2_RC_SUCCESS)
        goto error;

    r = Fapi_CreateKey(fapi_context, "HS/SRK/mySigningKey", "noDa, sign", ":", ":");
    if (r != TSS2_RC_SUCCESS)
        goto error;

    ...
... 

r = Fapi_Sign(fapi_context, "HS/SRK/mySigningKey", NULL, digest, digestSize, &signature, 
    &signatureSize, &publicKey, NULL);
if (r != TSS2_RC_SUCCESS)
goto error;

Fapi_Finalize(&fapi_context);

for (size_t i = 0; i < signatureSize; i++)
    printf("%02x", signature[i]);
printf("\n");
return 0;

error:
    Fapi_Finalize(&fapi_context);
    return 1;
}
Enhanced Authorizations (EA) & Policies

- Fine grained access control mechanism to TPM entities

- A policy consists of one or more sub-policies
  - sub-policies can be combined via AND / OR

- “EA policies can become complex very quickly.” *(A Practical Guide to TPM 2.0)*

Yet another complex TPM feature
Policies by Example

- chmod 446 nvindex (more or less)
  - Read for everyone
  - Write with correct password
Policies – TSS JSON Policy Editor

TSS Json Policy Editor

- TSS FAPI Policy Editor

Policy root element

- description

  This is my first policy

  The human readable description of the policy

- policy

  List of policy elements, each element is AND combined with the other policies in the list

  Item 1: Policy OR

  Combines two policy entries by an OR

  type

  or

- branches

  List of policy elements, each element is AND combined with the other policies in the list

  Item 1: Policy

  name

  NVRead

  description

  NV Read is always allowed

- policy

  List of policy elements, each element is AND combined with the other policies in the list

  Item 2: Policy

  name

  NVWrite

  description

  For NV Write supply a password

JSON Output

Demo Time!
$ cat mynvpolicy.json
{
  "description": "A simple read write nv policy",
  "policy": [
    {
      "type": "or",
      "branches": [
        {
          "name": "NVRead",
          "description": "this is the read policy",
          "policy": [
            {
              "type": "commandCode",
              "code": "NV_READ"
            }
          ]
        },
        {
          "name": "NVWrite",
          "description": "NV Write part - please supply a password",
          "policy": [
            {
              "type": "commandCode",
              "code": "NV_WRITE"
            },
            {
              "type": "password"
            }
          ]
        }
      ]
    }
  ]
}
Policies – Usage with FAPI Tools

$ tss2_import -p /policy/nvtest -i mynvpolicy.json

$ tss2_createnv -p /nv/Owner/mynvtest -P /policy/nvtest -s 64
New password: password
Re-enter new password: password

$ tss2_nvwrite -p /nv/Owner/mynvtest -i -
Hello OSSEU & ELCE!
Select a branch for /nv/Owner/mynvtest "PolicyOR"
  1 NVRead
  2 NVWrite
Your choice: 2
Authorize /nv/Owner/mynvtest "": password

$ tss2_nvread -p /nv/Owner/mynvtest -o -
Select a branch for /nv/Owner/mynvtest "PolicyOR"
  1 NVRead
  2 NVWrite
Your choice: 1
Hello OSSEU & ELCE!
Policies – Where to Start?

› TCG TSS JSON Policy Language Specification:
  https://trustedcomputinggroup.org/resource/tcg-tss-json

› Policies used in the TSS Repository:
  https://github.com/tpm2-software/tpm2-tss/tree/master/test/data/fapi/policy

› JSON Schema (WIP)
  https://github.com/PeterHuewe/TSS_JSON_Policy_Schema
  - Can be used by a JSON UI Editor
  - E.g. https://json-editor.github.io/json-editor
  - Great to explore TPM Policies

Looking for more fun with NV Indices and Policies?
Introducing TPM NV Storage with E/A Policies and TSS-FAPI - Andreas Fuchs, Fraunhofer SIT
October 29 • 14:40 - 15:25 @ LSSEU2020
What about Disk Encryption?

› TPM support on its way!
  – Initiated by Andreas Fuchs
  – [PR #98](https://github.com/infineon/linux-tpm2/commit/123456) (by Ondrej Kozina, maintainer)
  – Will ship with release 2.4.0!

› Support for **loadable plugins**
  – TPM support built-in

› Bonus:
  – Completely new plugin system
  – Ease of integration for smart cards etc.
LUKS – A Sneak Peak

NAME
cryptsetup-tpm2 - tool for activating LUKS2 encrypted volumes using TPM

SYNOPSIS
cryptsetup-tpm2 <options> <action> <action args>

DESCRIPTION
cryptsetup-tpm2 is used to conveniently lock and activate LUKS2 volumes using a passphrase stored in a TPM 2.0. TPM is a secure cryptoprocessor (similar to a smartcard) present in most of the recent laptops (and some desktops). cryptsetup-tpm2 stores metadata in a tpm2 token inside the tokens section of the LUKS2 header (see LUKS2 On-Disk Format Specification).

This utility supports sealing the TPM key to a current HW/SW state. To fully utilize this feature, it is required that the state is measured to the PCR banks by BIOS/UEFI, bootloader (e.g. TrustedGrub) and kernel (IMA subsystem). If anything in the boot chain deviates from the pre-measured state, the passphrase won’t be released from the TPM.

You can seal the passphrase to a specified PCR. They have the following meaning:

PCRA-7 Measured by BIOS/EFI. includes measurement of items like boot options and order, microcode or secure boot status.

PCRA16 Debug register

In TrustedGrub2 PCRs contain the following measurements:

PCRA8 First sector of TrustedGRUB2 kernel (diskboot.img)

PCRA9 TrustedGRUB2 kernel (core.img)

PCRA10 Loader measurements - currently linux-kernel, initrd, nt1dr, chainloader, multiboot, module

PCRA11 Contains all command line arguments from scripts (e.g. grub.cfg) and those entered in the shell

PCRA12 LUKS-header
Overview over the TPM Open Source Ecosystem

Cryptsetup
LUKS
Disk-Encryption

Keylime
Remote Attestation

OpenConnect
VPN

StrongSwan
IPSec

Your Application

tpm2-tss-engine
OpenSSL Engine

tpm2-pkcs11
PKCS#11 Provider

tpm2-tools
CLI Tools

tpm2-tss
Python Bindings

Keylime
Remote Attestation

OpenConnect
VPN

StrongSwan
IPSec

tpm2-software.github.io

Cryptsetup
LUKS
Disk-Encryption

Keylime
Remote Attestation

OpenConnect
VPN

StrongSwan
IPSec

tpm2-software.github.io
A Look into the Future...

I want to use the TPM...
A Look into the Future...

I want to use the TPM...

It's not rocket science

TPM Support Everywhere

› Ubiquitous usage
  - Transparent
  - Easy to use
  - Secure keys for everyone
  - Secure platforms for everyone

› Similar to HTTPS / TLS
  - Nobody talks about this anymore

› For this we need your help!
Need Help? Want to Contribute?

- Start at our GitHub Pages!
  https://tpm2-software.github.io/

- Mailing List
  https://lists.01.org/postorius/lists/tpm2.lists.01.org/

- Chat on Gitter
  https://gitter.im/tpm2-software/community

- IRC @ FreeNode
  #tpm2.0-tss

- Slack
  https://tpm2-tss.slack.com

Copyright © Infineon Technologies AG 2020. All rights reserved.
Let’s Start Hacking!

On your Laptop...

```bash
$ ls /dev/tpm*
/dev/tpm0 /dev/tpmrm0
```

... or a TPM simulator (development only)

- SWTPM Simulator
  [github.com/stefanberger/swtpm](https://github.com/stefanberger/swtpm)
- IBM TPM Simulator
  [https://sourceforge.net/projects/ibmswtpm2/](https://sourceforge.net/projects/ibmswtpm2/)

... the Raspberry Pi...

IRIDIUM9670 by Infineon

[Let's Trust TPM](https://sourceforge.net/)
Questions?
Part of your life. Part of tomorrow.