Hacking (with) a TPM

Don’t ask what you can do for TPMs,

ask what TPMs can do for you

AndreasFuchsSIT (@Github)
tpm2-software.github.io

$ whoami / Full Disclosure

- **Working on TPMs**
  - for Fraunhofer-SIT, some sponsored by Infineon
  - with contributions and maintainers from Intel, Infineon, etc and **hobbyist(s)**
- **TCG (Trusted Computing Group): TSS-WG chair**
- **tpm2-software project maintainer**
  - tpm2-tss
  - tpm2-tss-engine
  - tpm2-totp
- **TPM/TSS (1.2) consumer for 13 years (as a student)**
- **TSS 2.0 for 5 years now**
- **Opinions are mine, all typos are yours to keep**

Agenda

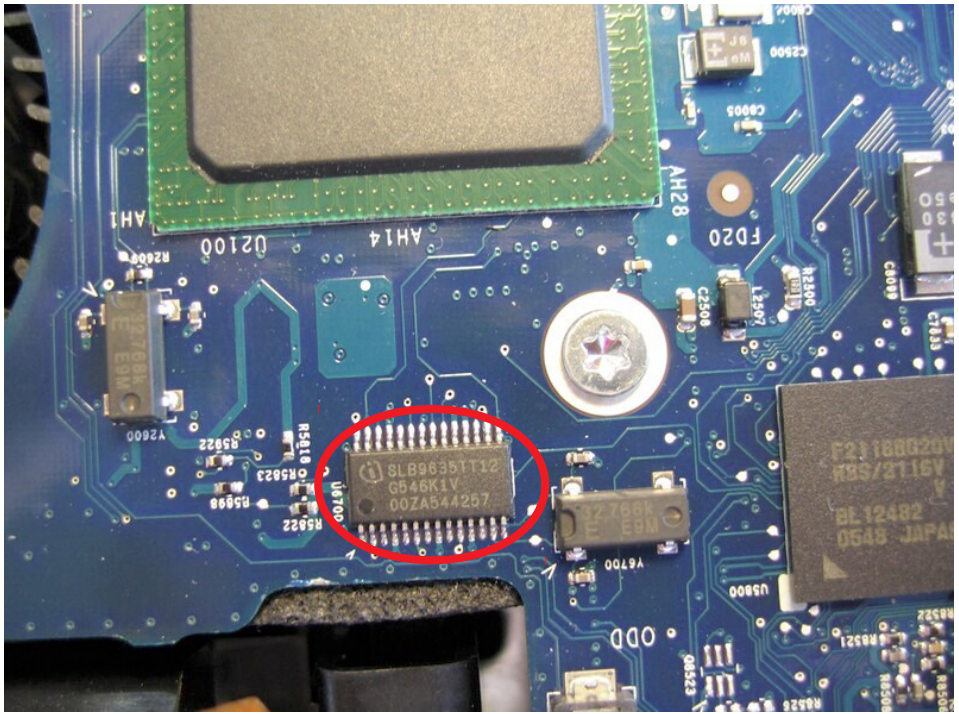
- **(Some) Introduction**
- **Credential protection**
  - TPMs for OpenSSL
  - TPMs as (virtual) SmartCards
- **(Early) Boot protections**
  - “Bitlocker for Linux”
  - Integrity Checking BIOS
- **Getting started yourself**

Demo preparations

```
sudo chmod go+rw /dev/tpmrm0
for i in tss tss-engine pkcs11 totp tools; do
       git clone --depth=1 \
https://github.com/tpm2-software/tpm2-${i}.git \
  && pushd tpm2-${i} \
  && ./bootstrap \
  && ./configure --enable-plymouth --sysconfdir=/etc \
  && sudo make -j install \
  && popd
done
tpm2-getcap properties-fixed
```

(Some) Introduction

- **Security Chip on Mainboard**
- **Thx @M$ for giving TPMs to all of us “for cheep”**
- **(Pretty) High security**
  - Common Criteria and such
  - except RSA-prime, tpm.fail, ...
- **Capable of crypto, (some) storage and recording boot’s hash values**
- **It’s passive !**



Are TPMs dangerous ?

- **TPMs' reputation**

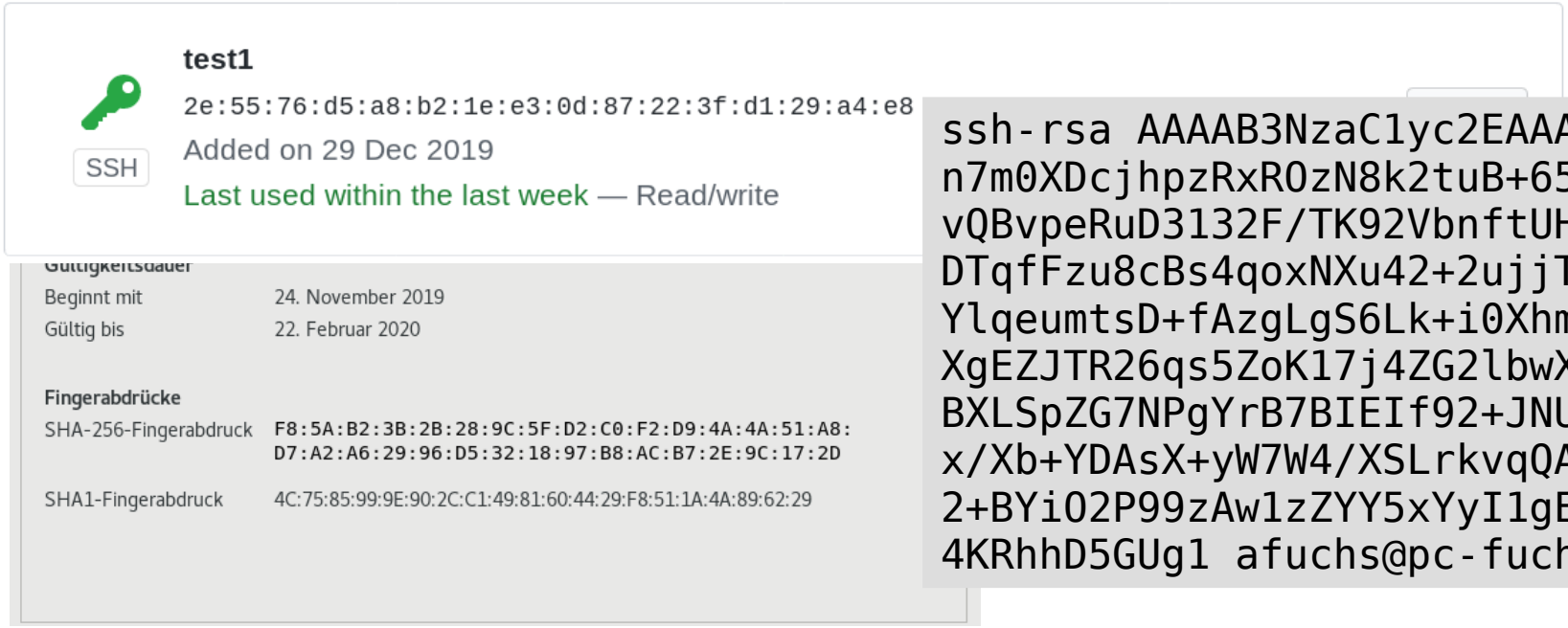
  "DRM devices that remote control our PCs"

- **TPMs in reality**
  - "Embedded SmartCards"
  - Integrity reporting / attestation capabilities
- **Stallman/GNU**

  [..] Therefore, we conclude that the "Trusted Platform Modules" available for PCs are not dangerous, and there is no reason not to include one in a computer or support it in system software. [..] https://www.gnu.org/philosophy/can-you-trust.en.html

Credential Protection

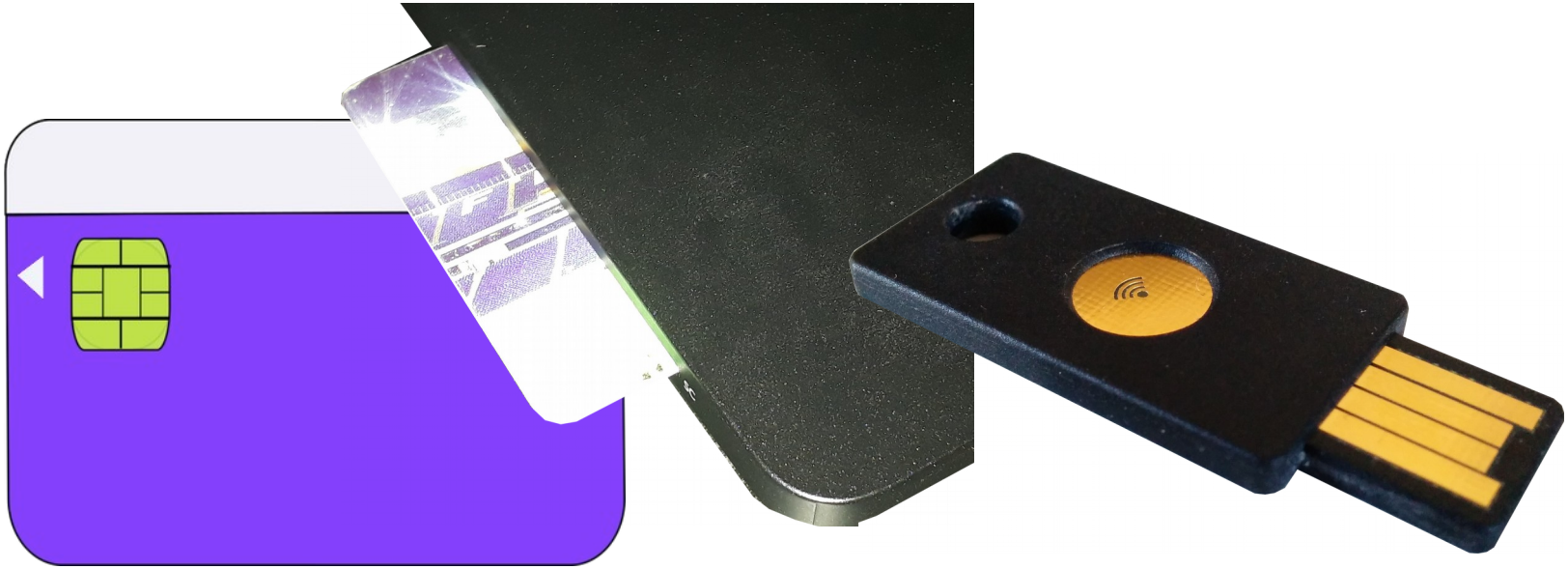
- **Who’s using public key crypto ?**




```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC1U
n7m0XDcjhpzRxROzN8k2tuB+6534SbdE7A83GX4nU
vQBvpeRuD3132F/TK92VbnftUHkJzV1gmlFi6A/M6
DTqfFzu8cBs4qoxNXu42+2ujjTOtbPwHPzbA3Lu+S
YlqeumtsD+fAzgLgS6Lk+i0XhmHBmjAWyJy4A3pn1
XgEZJTR26qs5ZoK17j4ZG2lbwXzgG466tkywmE/NO
BXLSpZG7NPgYrB7BIEIf92+JNUCD9tx9O2DKyzY+4
x/Xb+YDAsX+yW7W4/XSLrkvqQA4edveJvwv2W1KP0
2+BYiO2P99zAw1zZYY5xYyI1gEo+sNZVHfErTzpZi
4KRhhD5GUg1 afuchs@pc-fuchs
```

Credential Protection

- **Who’s using SmartCards / YubiKeys / … ?**



Credential Protection

- **Who’s using them like this ?**

Yubikey by Wikipedia-Commons / Autopilot CC-BY-SA

Credential Protection

- **What's the security idea**
  - Proof of possession
  - Proof of knowledge
- **What is proof of possession ?**
  - Something non-duplicable (unclonable)
  - Thus only 1 person can have possession
  - Proof of possession of my TPM-equiped laptop == Yubikey nano
- **But what if you're hacked ?**
  - General problem with all proof-of-possession means
  - Difference to soft tokens
    - Temporarily bound to time of hack (no copy)
    - No chance for Heartbleed

Credential Protection: OpenSSL demo (1.0.3)

- **Generate a key**

  tpm2tss-genkey mykey-engine.pem

- **Generate a (self-signed) certificate**

  openssl req -new -x509 -engine tpm2tss -key mykey-engine.pem -keyform engine -out mykey-engine.crt

- **Using curl**

  curl --insecure --engine tpm2tss --key-type ENG --key mykey-engine.pem --cert mykey-engine.crt https://192.168.122.1 (to nginx with ssl_client_certificate = optional_no_ca)

- **Using nginx**

  ssl_certificate = /home/andreas/mykey-engine.crt (in sites-enabled/default)
  ssl_certificate_key = engine:tpm2tss:/home/andreas/mykey-engine.pem

  ssl_engine = tpm2tss; (in nginx.conf; work around a bug in nginx on ENGINE_init())

Credential Protection: PKCS11 demo (1.0_rc0)

- **Generate Token (weirdly not in make install)**
  - export PYTHONPATH=$PWD/tpm2-pkcs11/tools
    export TPM2_PKCS11_STORE=$HOME/
    tpm2-pkcs11/tools/tpm2_ptool init --path=$TPM2_PKCS11_STORE
    tpm2-pkcs11/tools/tpm2_ptool addtoken --pid=1 --label=label \
    --sopin=123456 --userpin=1234 --path=$TPM2_PKCS11_STORE
    tpm2-pkcs11/tools/tpm2_ptool addkey --algorithm=rsa2048 --label="label" \
    --userpin=1234 --path=$TPM2_PKCS11_STORE
- **ssh-genkey/-copy-id**

  ssh-keygen -D /usr/local/lib/libtpm2_pkcs11.so → authorized_keys

  ssh -I /usr/local/lib/libtpm2_pkcs11.so afuchs@192.168.122.1
- **git+ssh+pkcs11**

  echo -e '#!/bin/sh\nssh -I /usr/local/lib/libtpm2_pkcs11.so $@' >ssh-pkcs11
  chmod +x ssh-pkcs11
  export GIT_SSH=$PWD/ssh-pkcs11
  git clone --depth=1 git@github.com:AndreasFuchsSIT/tpm2-tss.git tpm2-tss-ssh

“Bitlocker for Linux”

- **HDD-crypto on Linux**
  - LUKS / (lib)cryptsetup
  - VolumeKey encrypted with kdf’d password
  - multiple “keyslots” for key encryption keys
- **How it works**
  - Store VolumeKey inside TPM (nv space)
  - Store meta-data (tpm nv index, etc) in LUKS header
- **Now wip-tokens @upstream**


```
{
  "keyslots": {
    "0": {
      "type": "luks2",
      "key_size": 32,
      "kdf": {
        ...
      },
      "af": {
        "type": "luks1",
        "hash": "sha256",
        "stripes": 4000
      },
      "area": {
        "type": "raw",
        "encryption": "aes-xts-plain64",
        "key_size": 32,
        "offset": "32768",
        "size": "131072"
      }
    }
```

```
{
  "keyslots": {
    "1": {
      "type": "tpm2",
      "key_size": 32,
      "area": {
        "type": "tpm2nv",
        "nvindex": 29294593,
        "pcrselection": 0,
        "pcrbanks": 1,
        "noda": true
      },
```

Demo Time ! cryptsetup (PoC in MR !51)

- **Ubuntu Install with LUKS & LVM chosen during partitioning**

```
./autogen.sh && \
./configure --prefix=/usr --libdir=/lib/x86_64-linux-gnu \
        --sbindir=/sbin --mandir=/usr/share/man \
        --enable-libargon2 --enable-shared \
        --enable-cryptsetup-reencrypt --enable-tpm2 && \
sudo make -j install
sudo update-initramfs -u
sudo cryptsetup luksAddKey --tpm /dev/vda5
sudo cryptsetup luksDump /dev/vda5
```

- **From Install USB-Stick (or similar)**

```
sudo cryptsetup convert /dev/vda5 --type luks2 (from bootmedia)
```

- **See you at reboot…**

  (Caution PoC code; completely WIP @upstream)

Integrity Checking

- **tpm2-totp**
  - Based on tpm-totp by Matthew Garret @32c3
  - Detail on TPM based attestation capabilities: https://media.ccc.de/v/32c3-7343-beyond_anti_evil_maid
- **The idea**
  - The TPM records hashes of BIOS, Kernel and Initrd
  - Share a secret between TPM and your phone
  - Restrict the usage of secret to recorded hashes
  - Calculate time-based OTPs on boot
  - Thus verify that PC BIOS and Kernel were not altered

Demo Time ! (0.2.1 / feature-gtk)

- **Install**

  ./configure --enable-plymouth --sysconfdir=/etc

  sudo update-initramfs -u

- **tpm2-totp / gtpm2-totp**

  tpm2-totp / gtpm2-totp

- **Let’s reboot both demos**

How to hack (with) TPMs yourself

→ https://tpm2-software.github.io ←

- **Look at tss2_fapi.h or tss2_esys.h and existing code**
- **Read the TPM- and TSS specs (tpm2-software → External)**
- **Need inspiration ? (tpm2-software → software → scroll down)**
- **Look at tpm2-tools:**
  - tpm2_* is (mostly) 1:1 tss2_esys.h
  - tss2_* is 1:1 tss2_fapi.h
- **Talk, mail, gitter: @AndreasFuchsSIT / andreas.fuchs@sit.fraunhofer.de**

- **Bonus tip: Random Fails ? → TPM Resource Exhaustion**

  tpm2_flushcontext -t / -l / -s

Question time