PV204 Security technologies

Trusted boot, TPM

Petr Švenda svenda@fi.muni.cz
Faculty of Informatics, Masaryk University

CS
Centre for Research on Cryptography and Security

Laboratory

1. Presentation of projects
2. Basic operations with TPM chip

Homework – no new homework ☺

- But work on project!
- The deadline for memory analysis homework moved
  - New date is 4.5. 23:59
  - (the lab materials were not uploaded to IS)

Projects - timeline

1. Identify target scenario, design of applications: 3 points ~~(6.4.2017)~~
   - Report (max. 2 pages A4)
   - Deadline moved to 11.4.
2. Write code (GitHub): 10 points (before 4.5.2017)
   - JavaCard application, PC-based application
   - Design, code + presentation ~~(4.5.2017~~, your seminar group, random team member) – presentation moved to 27.4.
3. Review and attack implementations: 7 points (before 18.5.2017)
   - Up to 4 points assigned by reviewers, up to 3 points by me
   - Review and attack implementations of other teams
   - Report + presentations (18.5.2017, random team member)

- At least 10 points (total) from project are required

Organizational

- Not every computer have TPM chip
  - Make group with at least one TPM-enabled computer
  - Use own computer (if TPM-enabled) or provided ones
- Use of TPM differs between Windows and Linux
  - We will focus on Windows, but you may try Linux as well
- Prepared software (Windows, Linux)
  - Preconfigured binaries and cheetsheet in IS 10_TPM.zip
  - Or https://www.fi.muni.cz/~xsvenda/tpm.zip
  - Use printed cheetsheet

Questions to answer

1. Figure out maker and version of TPM chip
2. Obtain number of OS boot counts
3. Obtain list of PCR registers
4. Generate new RSA keypair and export public key
5. Seal (encrypt) data so only your machine will be able to decrypt
6. (optional) Try to create Remote Attestation report

Fill the table on the whiteboard ☺

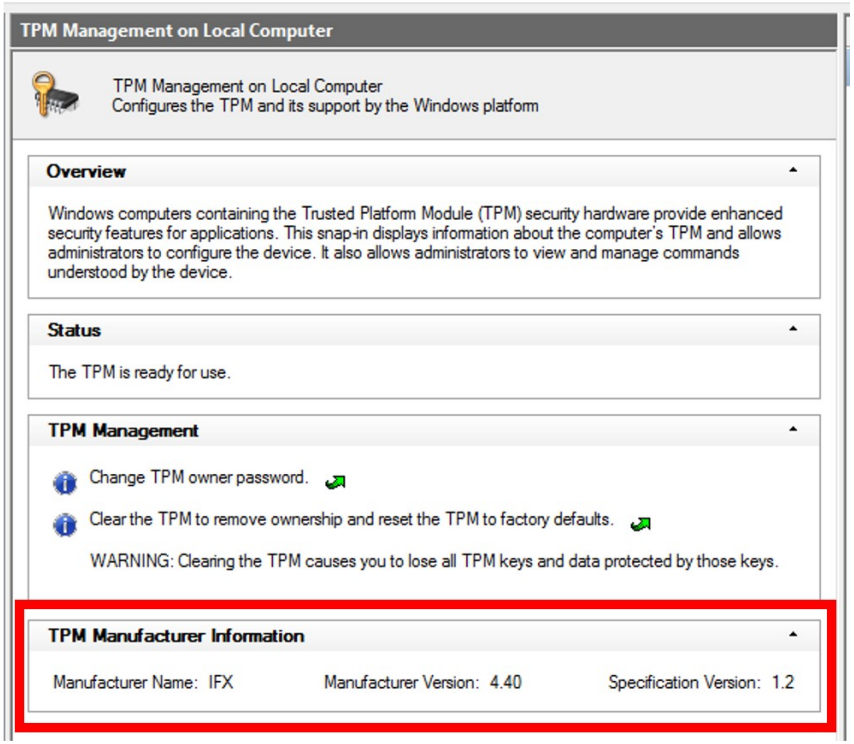
- Name of your computer (e.g., HP ProBook 6470b)
- Supported version of TPM specification
- TPM manufacturer and manuf. version
- Number of OS Boot Counts
- Platform Configuration Registers #used / #unused
- Generate on-TPM RSA key (tick OK when done)
- Seal data (tick OK when done)
- Remote Attestation (tick OK when done)

Sending TPM commands, tools

- ISO/IEC 11889 standard for secure crypto-processor
- Versions published by Trusted Computing Group
  - TPM 1.2 (2011), TPM 2.0 (2016, not compatible with 1.2)
  - https://trustedcomputinggroup.org
- Windows: Microsoft PCPTool, TSS.MSR
- Linux: tpm_tools, GUI TPMManager
- (All tools and scripts available in single package)
  - https://www.fi.muni.cz/~xsvenda/tpm.zip

Is TPM chip inside my computer?

- Windows
  - WinButton+R
  - tpm.msc (requires admin)
- Linux
  - sudo apt-get install tpm-tools
  - tpm_setactive -s
  - tpm_setactive -a
  - tpm_version
  - (systemctl restart tcsd)
- Check BIOS settings (TPM or Security chip…)



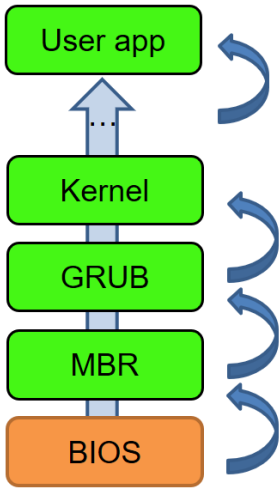
TPM platform info

- Provides information about your platform state
- **W:** PCPTool.exe GetPlatformCounters
- **L:** not readily available, try
  - sudo cat /sys/kernel/security/tpm0/ascii_bios_measurements
  - sudo cat /sys/kernel/security/ima/ascii_runtime_measurements

```
<PlatformCounters>
  <OsBootCount>44</OsBootCount>
  <OsResumeCount>2</OsResumeCount>
  <CurrentBootCount>0</CurrentBootCount>
  <CurrentEventCount>66</CurrentEventCount>
  <CurrentCounterId>179136858</CurrentCounterId>
  <InitialBootCount>0</InitialBootCount>
  <InitialEventCount>64</InitialEventCount>
  <InitialCounterId>179136858</InitialCounterId>
</PlatformAttestation>
```

Reboot =>

```
<PlatformCounters>
  <OsBootCount>45</OsBootCo
  <OsResumeCount>0</OsResu
  <CurrentBootCount>0</Curre
  <CurrentEventCount>67</Cur
  <CurrentCounterId>17913685
  <InitialBootCount>0</InitialBo
  <InitialEventCount>67</Initial
  <InitialCounterId>179136858<
</PlatformAttestation>
```



Platform attestation – PCR registers

- **W:** PCPTool.exe GetPCRs
- **L:** cat `find /sys/class/ -name "tpm0"`/device/pcrs

Table 12-1. Example PCR Allocation

PCR Number	Allocation
0	BIOS
1	BIOS configuration
2	Option ROMs
3	Option ROM configuration
4	MBR (master boot record)
5	MBR configuration
6	State transitions and wake events
7	Platform manufacturer specific measurements
8–15	Static operating system
16	Debug
23	Application support

```
bug>PCPTool.exe GetPCRs
<PCRs>
  <PCR Index="00">8cb1a2e093cf41c1a726bab3e10bc1750
  <PCR Index="01">b2a83b0ebf2f8374299a5b2bdfc31ea95
  <PCR Index="02">b2a83b0ebf2f8374299a5b2bdfc31ea95
  <PCR Index="03">b2a83b0ebf2f8374299a5b2bdfc31ea95
  <PCR Index="04">1e3c5e15b5f023765147535e092d22d7c
  <PCR Index="05">75acbe8a48ba02a85d6301b33005d0867
  <PCR Index="06">b2a83b0ebf2f8374299a5b2bdfc31ea95
  <PCR Index="07">b2a83b0ebf2f8374299a5b2bdfc31ea95
  <PCR Index="08">0000000000000000000000000000000000
  <PCR Index="09">0000000000000000000000000000000000
  <PCR Index="10">0000000000000000000000000000000000
  <PCR Index="11">ebb98df76613280f20dc38221143a9e72
  <PCR Index="12">67afac5ca0fc6c9a3d881d681121f7d43
  <PCR Index="13">be1d9bd7318a9140b26f00a5283f37a61
  <PCR Index="14">7f599cd09efefc7422085a0f490f8f1cb
  <PCR Index="15">0000000000000000000000000000000000
  <PCR Index="16">0000000000000000000000000000000000
  <PCR Index="17">ffffffffffffffffffffffffffffffffff
  <PCR Index="18">ffffffffffffffffffffffffffffffffff
  <PCR Index="19">ffffffffffffffffffffffffffffffffff
  <PCR Index="20">ffffffffffffffffffffffffffffffffff
  <PCR Index="21">ffffffffffffffffffffffffffffffffff
  <PCR Index="22">ffffffffffffffffffffffffffffffffff
  <PCR Index="23">0000000000000000000000000000000000
</PCRs>
```

Platform info

- Obtain information about your platform
- Version info: pcptool GetVersion
- Get platform counters: pcptool GetPlatformCounters

```
<PlatformCounters>
  <OsBootCount>44</OsBootCount>
  <OsResumeCount>2</OsResumeCount>
  <CurrentBootCount>0</CurrentBootCount>
  <CurrentEventCount>66</CurrentEventCount>
  <CurrentCounterId>179136858</CurrentCounterId>
  <InitialBootCount>0</InitialBootCount>
  <InitialEventCount>64</InitialEventCount>
  <InitialCounterId>179136858</InitialCounterId>
</PlatformAttestation>
```

Reboot =>

```
<PlatformCounters>
  <OsBootCount>45</OsBootCo
  <OsResumeCount>0</OsResu
  <CurrentBootCount>0</Curre
  <CurrentEventCount>67</Cur
  <CurrentCounterId>17913685
  <InitialBootCount>0</InitialBo
  <InitialEventCount>67</Initial
  <InitialCounterId>179136858<
</PlatformAttestation>
```

Encrypt data only for your TPM (Windows)

- (RSA key with name *openlab* already generated)

1. Export public key
   - PCPTool.exe GetPubKey openlab openlab.pub
2. Encrypt data by public key
   - PCPTool.exe Encrypt openlab.pub Hello msg_enc.bin
3. Decrypt only on your machine
   - PCPTool.exe Decrypt openlab msg_enc.bin

Encrypt data only for your TPM (Linux)

- Sealed storage using Root Storage Key

```
echo "Hello World!" > cleartext.txt
tpm_sealdata --well-known --infile cleartext.txt > encrypted.txt
tpm_sealdata -z -i cleartext.txt > encrypted.txt
cat encrypted.txt
tpm_unsealdata --srk-well-known --infile encrypted.txt
tpm_unsealdata -z -i encrypted.txt
```

- The proper way for TPM encryption is installing the openssl_tpm_engine,however the repository is not maintained anymore and does not build on my system (not even with OpenSSL 0.9.8) https://sourceforge.net/p/trousers/openssl_tpm_engine/ci/master/tree/
- There is a newer patched version, but too complicated: https://blog.hansenpartnership.com/using-your-tpm-as-a-secure-key-store/

Holy grail: Remote attestation

- Apps running on your computer measured in PCRs
- Your TPM contains unique Endorsement key
- You can generate Attestation key inside TPM (AIK)
  - And sign AIK by Endorsement key (inside TPM)
- You can sign your PCRs by AIK (inside TPM)
- Remote party can verify signature on AIK key
  - Using public key of Endorsement key
- Remote party can verify signature on PCRs
  - Using public key of AIK key
- Remote party now knows what you are running

Attestation keys

1. Create attestation identity key (AIK)
   - CreateAIK AIK_NAME filename aikNonce
2. Get public part of attestation key
   - GetPubAIK
3. Authentication of generated AIK to remote entity
   - Omitted (challenge-response and endorsement key used)
4. Get platform attestation signed by AIK
   - GetPlatformAttestation

1. Create attestation key

- PCPTool.exe CreateAIK myAIK test.tmp 1234

```
<AIK>
 <RSAKey size="283" keyName="myAIK">
  <Magic>RSA1<!-- 0x31415352 --></Magic>
  <BitLength>2048</BitLength>
  <PublicExp size="3">
   010001
  </PublicExp>
  <Modulus size="256" digest="520aabc242eddb488d1c3da30f56b4268222982a">
   9ddc3bb99eab0d9…d0fb46a48224cf15e9
  </Modulus>
  <Prime1/>
  <Prime2/>
 </RSAKey>
 <IdentityBinding size="568">01010000000000079139f69c93c042496a8e958ec5930662c6c
    ccafbf000000010…093873f194ce7b68ef667f00eca2090adad3
 </IdentityBinding>
</AIK>
```

2. Get public part of attestation key

- PCPTool.exe GetPubAIK test.tmp AIKPub.key

```
<RSAKey size="283" keyName="AIK">
  <Magic>RSA1<!-- 0x31415352 --></Magic>
  <BitLength>2048</BitLength>
  <PublicExp size="3">
   010001
  </PublicExp>
  <Modulus size="256"
   digest="520aabc242eddb488d1c3da30f56b4268222982a">
   9ddc3bb99eab0d913cd…0a40de6d62424b9a311
  </Modulus>
  <Prime1/>
  <Prime2/>
</RSAKey>
```

3. Get platform attestation

- PCPTool.exe GetPlatformAttestation myAIK attestation.tmp 4321
  - TpmAttGeneratePlatformAttestation() called internally
  - Large XML file is produced

- Why AIK is relevant for platform attestation?
- Why makes sense to have multiple AIKs?
- Why nonce 4321 is included?

4. Platform attestation – PCR registers

```
<PlatformAttestation size="30591">
  <Magic>PADS<!-- 0x53444150 --></Magic>
  <Platform>TPM_VERSION_12</Platform>
  <HeaderSize>28</HeaderSize>
  <PcrValues size="480">
    <PCR Index="0">8cb1a2e093cf41c1a726bab3e10bc1750180bbc5</PCR>
    <PCR Index="1">b2a83b0ebf2f8374299a5b2bdfc31ea955ad7236</PCR>
    <PCR Index="2">b2a83b0ebf2f8374299a5b2bdfc31ea955ad7236</PCR>
    <PCR Index="3">b2a83b0ebf2f8374299a5b2bdfc31ea955ad7236</PCR>
    <PCR Index="4">68fffb7e5c5f6e6461b3527a0694f41ebd07e4e1</PCR>
    <PCR Index="5">8e33d52190def152c9939e9dd9b0ea84da25d29b</PCR>
    <PCR Index="6">b2a83b0ebf2f8374299a5b2bdfc31ea955ad7236</PCR>
    <PCR Index="7">b2a83b0ebf2f8374299a5b2bdfc31ea955ad7236</PCR>
    <PCR Index="8">0000000000000000000000000000000000000000</PCR>
    <PCR Index="9">0000000000000000000000000000000000000000</PCR>
    <PCR Index="10">0000000000000000000000000000000000000000</PCR>
    <PCR Index="11">b2a83b0ebf2f8374299a5b2bdfc31ea955ad7236</PCR>
    <PCR Index="12">7c84e69cd581eefd7ebe1406666711fd4fda8aa8</PCR>
    <PCR Index="13">01788a8a31f2dafcd9fe58c5a11701e187687d49</PCR>
    <PCR Index="14">26cda47f1db41bedc2c2b1e6c91311c98b4e2246</PCR>
    <PCR Index="15">0000000000000000000000000000000000000000</PCR>
    <PCR Index="16">0000000000000000000000000000000000000000</PCR>
    <PCR Index="17">ffffffffffffffffffffffffffffffffffffffff</PCR>
    <PCR Index="18">ffffffffffffffffffffffffffffffffffffffff</PCR>
    <PCR Index="19">ffffffffffffffffffffffffffffffffffffffff</PCR>
    <PCR Index="20">ffffffffffffffffffffffffffffffffffffffff</PCR>
    <PCR Index="21">ffffffffffffffffffffffffffffffffffffffff</PCR>
    <PCR Index="22">ffffffffffffffffffffffffffffffffffffffff</PCR>
    <PCR Index="23">0000000000000000000000000000000000000000</PCR>
  </PcrValues>
...
```

4. Platform attestation – EFI boot info

```
<EV_EFI_GPT_Event PCR="05" EventDigest="46bd9620d741812ca1c2bd82c
  4546492050415254000001005c00000015ff46140000000010000000000
  <!-- EFI.PART.....F.....2.....2.....4...N.2.Eg.....
 </EV_EFI_GPT_Event>
 <EV_EFI_Boot_Services_Application PCR="04" Digest="65bb31b71a030a3fe
  18e086a40000000058ad180000000000000010000000034000000000
  <!-- .....X.....4.....0...E.F.I...B.o.o.t...B.o.o.t.x.6.4...e.f.i..... --
 </EV_EFI_Boot_Services_Application>
```