PV204 Security technologies LABS

JavaCard programming, Secure Multiparty Computation

Petr Švenda *svenda@fi.muni.cz* *@rngsec*
Centre for Research on Cryptography and Security, Masaryk University

CROCS
Centre for Research on Cryptography and Security

The masterplan for this lab

- Threshold ECDSA signature (MeeSign tool)
  - Large group, smaller groups
- Brainstorm interesting usages of MPC
- Manage and update JavaCard applications on smartcard

Secure Multiparty Computation

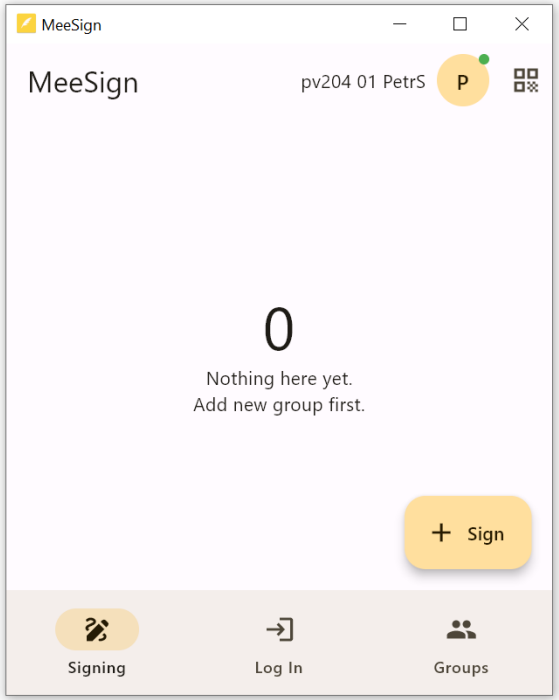
Preparation (every student)

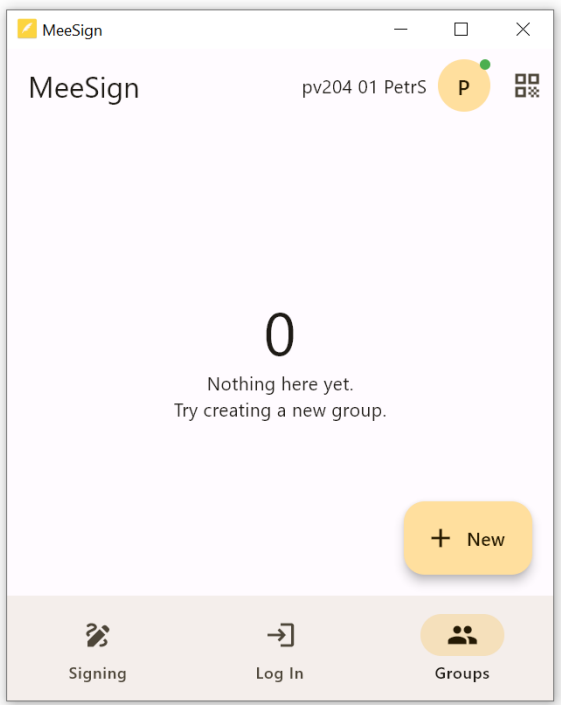
- Download and extract MeeSign client for your platform at https://meesign.crocs.fi.muni.cz/
- Connect to university network => wlan-fi or eduroam
- Start MeeSign application
- Check that server is set to `meesign.crocs.fi.muni.cz`
- ! Set your name as '`pv204 0x your_nick_here`' (replace x by number of your seminar group)
- Click Register

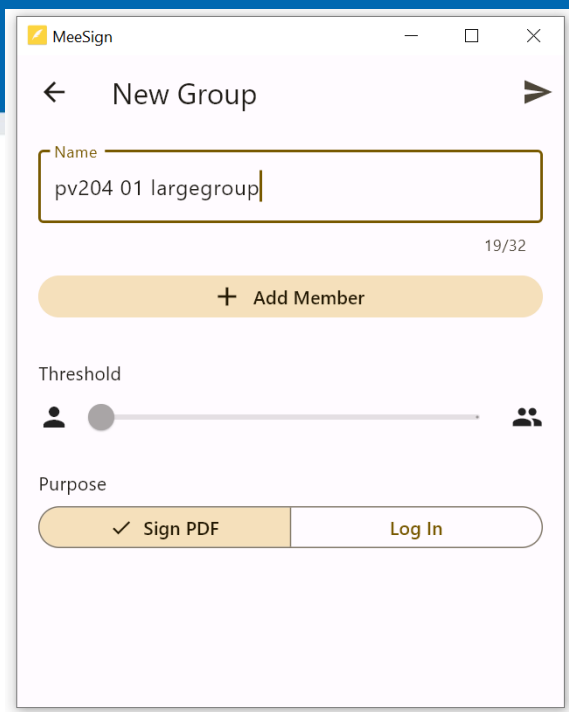
Task: Signing as a larger group

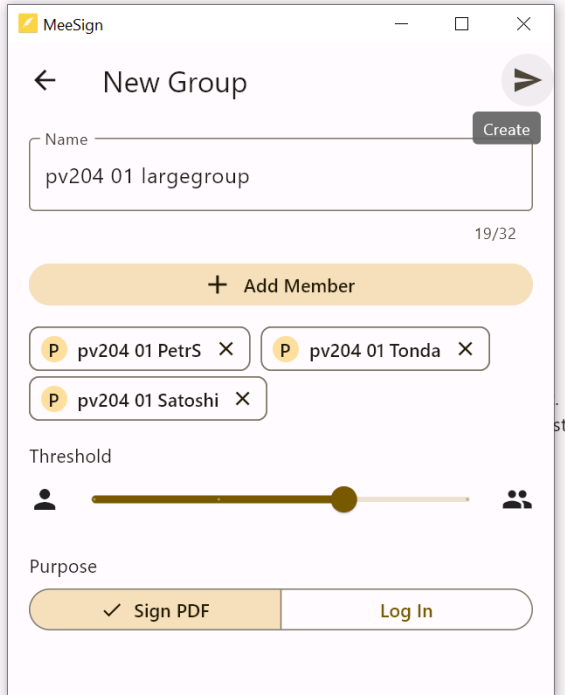
- New group ‘PV204_0x_large’ created by tutor
  - Threshold set to n-2
  - Students added by nickname (or QRCode)
- Confirm yourself in when prompted
- Tutor starts signing of document, wait for notification
- Open then sign pdf document shared, Sign afterwards
- Wait for the finalization (n-2 people needed)
- Check yourself properties of the resulting MPC signature
  - Adobe Acrobat Reader or https://ec.europa.eu/digital-building-blocks/DSS/webapp-demo/validation (upload signed file, Detailed report -> Basic Building Blocks SIGNATURE)

MeeSign
MeeSign
Name
pv204 01 PetrS
14/32
Server
meesign.crocs.fi.muni.cz
Register

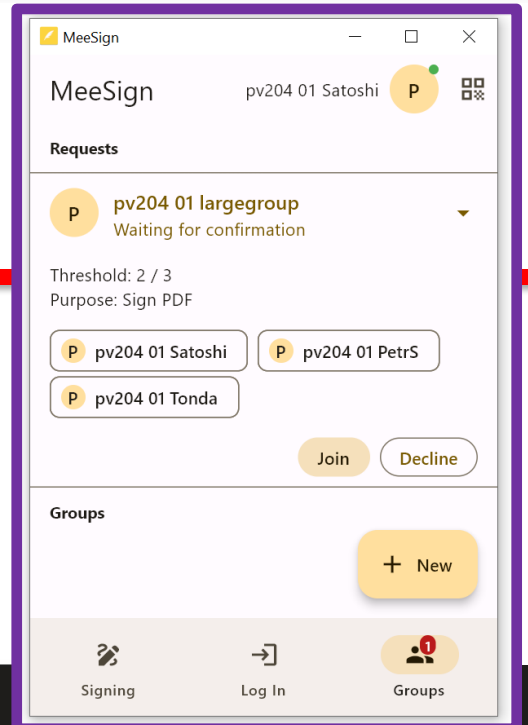

MeeSign
MeeSign
pv204 01 PetrS
P
0
Nothing here yet.
Add new group first.
Sign
Signing
Log In
Groups

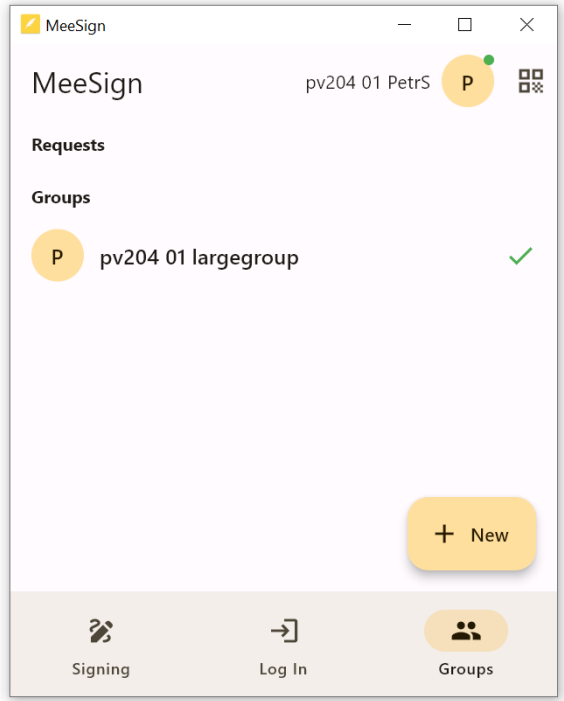

MeeSign
MeeSign
pv204 01 PetrS
P
0
Nothing here yet.
Try creating a new group.
New
Signing
Log In
Groups


MeeSign
New Group
Name
pv204 01 largegroup
19/32
Add Member
Threshold
Purpose
Sign PDF
Log In


MeeSign
New Group
Create
Name
pv204 01 largegroup
19/32
Add Member
P pv204 01 PetrS
P pv204 01 Tonda
P pv204 01 Satoshi
Threshold
Purpose
Sign PDF
Log In

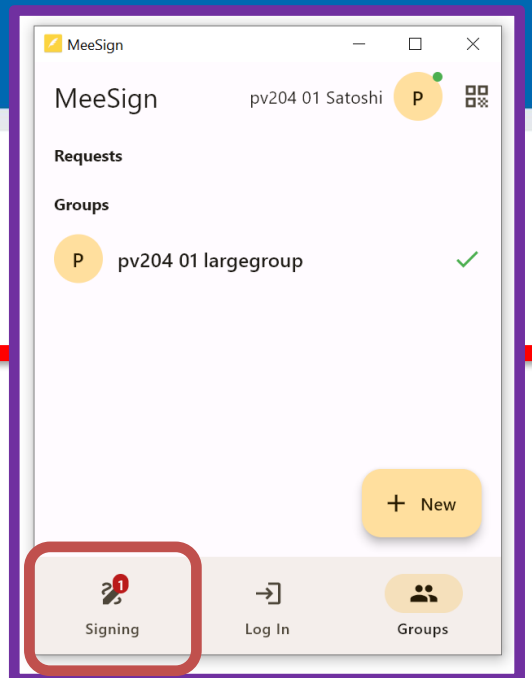
MeeSign
MeeSign
pv204 01 PetrS
P
Requests
P pv204 01 largegroup
Waiting for confirmation
Threshold: 2 / 3
Purpose: Sign PDF
P pv204 01 Satoshi
P pv204 01 PetrS
P pv204 01 Tonda
Join
Decline
Groups
New
Signing
Log In
Groups


MeeSign
MeeSign
pv204 01 Satoshi
P
Requests
P pv204 01 largegroup
Waiting for confirmation
Threshold: 2 / 3
Purpose: Sign PDF
P pv204 01 Satoshi
P pv204 01 PetrS
P pv204 01 Tonda
Join
Decline
Groups
New
Signing
Log In
Groups

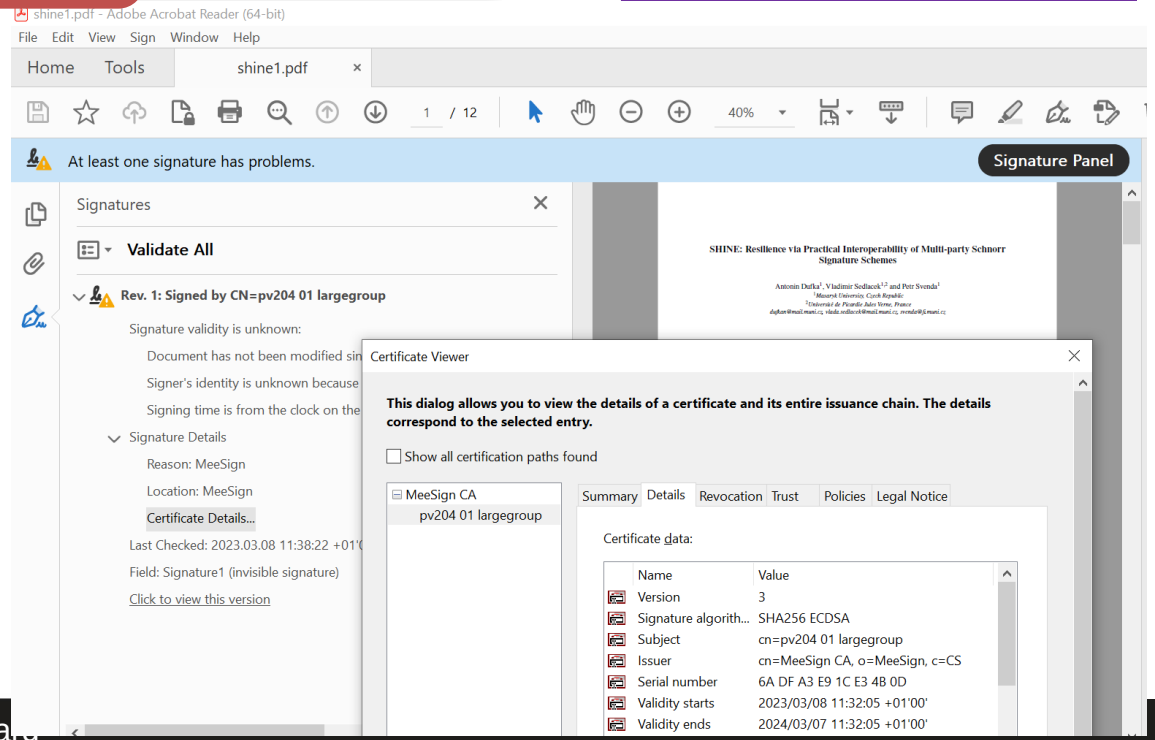

MeeSign
MeeSign
pv204 01 PetrS
P
Requests
Groups
P pv204 01 largegroup
New
Signing
Log In
Groups

MeeSign
MeeSign
pv204 01 PetrS
P
Select group
pv204 01 largegroup
+ Sign
Signing
Log In
Groups

MeeSign
MeeSign
pv204 01 PetrS
P
Requests
shine1.pdf
Waiting for confirmation
P pv204 01 largegroup
View
Sign
Decline
Signed files
+ Sign
Signing
Log In
Groups

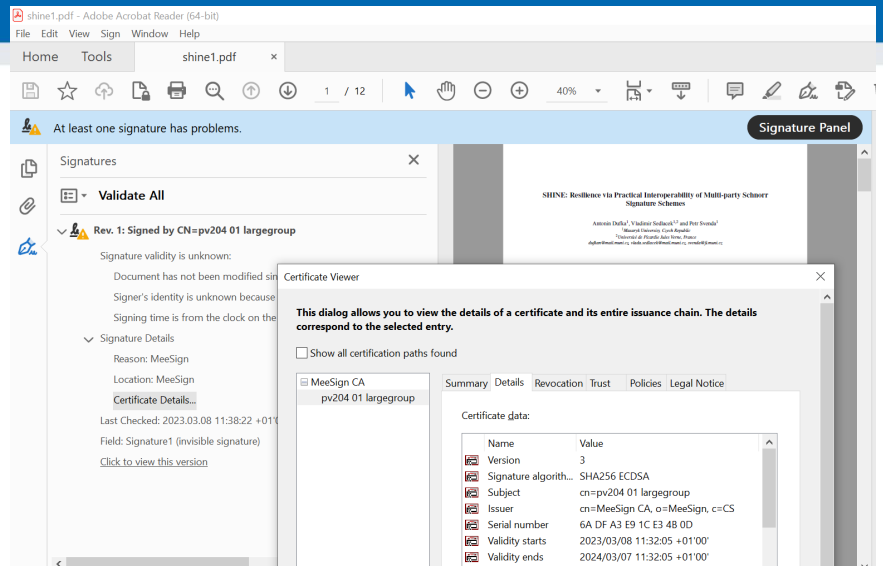

MeeSign
MeeSign
pv204 01 Satoshi
P
Requests
Groups
P pv204 01 largegroup
+ New
Signing
Log In
Groups

MeeSign
MeeSign
pv204 01 PetrS
P
Requests
Signed files
shine1.pdf
P pv204 01 largegroup
View
+ Sign
Signing
Log In
Groups


shine1.pdf - Adobe Acrobat Reader (64-bit)
File Edit View Sign Window Help
Home Tools
shine1.pdf
At least one signature has problems.
Signature Panel
Signatures
Validate All
Rev. 1: Signed by CN=pv204 01 largegroup
Signature validity is unknown:
Signature Details
Reason: MeeSign
Location: MeeSign
Certificate Details...
Field: Signature1 (invisible signature)
Click to view this version
SHINE: Resilience via Practical Interoperability of Multi-party Schnorr Signature Schemes
Certificate Viewer
This dialog allows you to view the details of a certificate and its entire issuance chain. The details correspond to the selected entry.
Show all certification paths found
MeeSign CA
pv204 01 largegroup
Summary Details Revocation Trust Policies Legal Notice
Certificate data:
Name Value
Version 3
Signature algorith... SHA256 ECDSA
Subject cn=pv204 01 largegroup
Issuer cn=MeeSign CA, o=MeeSign, c=CS
Serial number 6A DF A3 E9 1C E3 4B 0D
Validity starts 2023/03/08 11:32:05 +01'00'
Validity ends 2024/03/07 11:32:05 +01'00'

Verify pdf signature

- Check resulting signature
  - Adobe Acrobat Reader
  - pdfsig (poppler-utils)
  - Online https://ec.europa.eu/digital-building-blocks/DSS/webapp-demo/validation

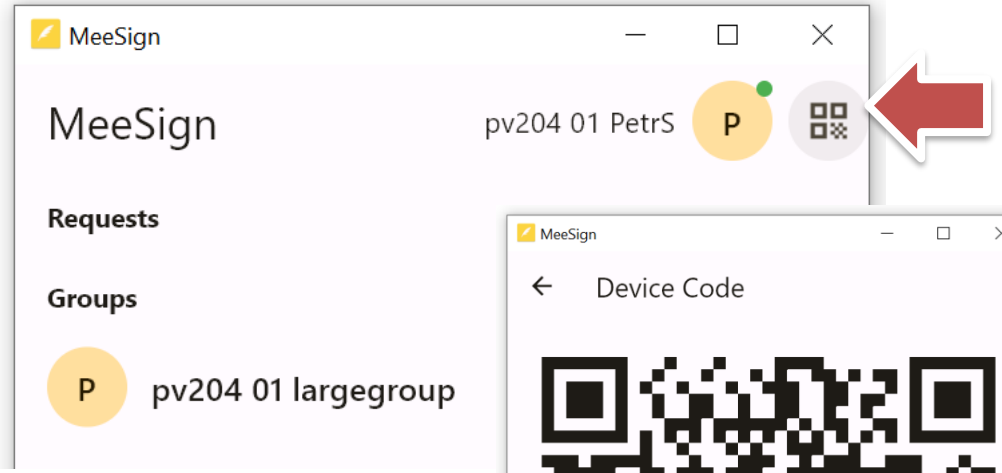



```
↳ pdfsig example.pdf
Digital Signature Info of: example.pdf
Signature #1:
  - Signer Certificate Common Name: Small Group (Jimmy & Joe)
  - Signer full Distinguished Name: CN=Small Group (Jimmy & Joe)
  - Signing Time: May 27 2022 09:05:26
  - Signing Hash Algorithm: SHA-256
  - Signature Type: adbe.pkcs7.detached
  - Signed Ranges: [0 - 106317], [125263 - 125849]
  - Total document signed
  - Signature Validation: Signature is Valid.
  - Certificate Validation: Certificate issuer isn't Trusted.
```

Cryptographic Verification : PASSED
Has the reference data object been found?
Is the reference data object intact?
Is the signature intact?
Signature Acceptance Validation : PASSED
Is the structure of the signature valid?
Is the signed attribute: 'signing-certificate' present?
Is the signed qualifying property: 'signing-time' present?
Is the signed qualifying property: 'message-digest' or 'SignedProperties' present?
Are cryptographic constraints met for the signature creation?
Are cryptographic constraints met for the message digest?

Task: Signing in smaller group

- Groups of 3 students
- Create new group with some unique name, add yourself and peers
  - Try to add peers via qrcode
    - Users display qrcode (upper right corner)
    - Group creator Add member -> Scan
  - Set threshold to 2-of-3
- Initiate MPC signing, sign
- View then sign pdf document
- Check yourself the resulting MPC signature
  - Adobe Acrobat Reader or https://ec.europa.eu/digital-building-blocks/DSS/webapp-demo/validation



Questions

- What is difference between group 2-of-3 and 3-of-3? What is security advantage of the first and second one respectively?
- What if two people from the group refuses to sign?
- How many devices needs an attacker to compromise to forge signatures?
- What is the reason why Adobe Acrobat Reader displays warning about resulting signature?
- What is a public key of your group?

Task: Brainstorm interesting usages for MPC

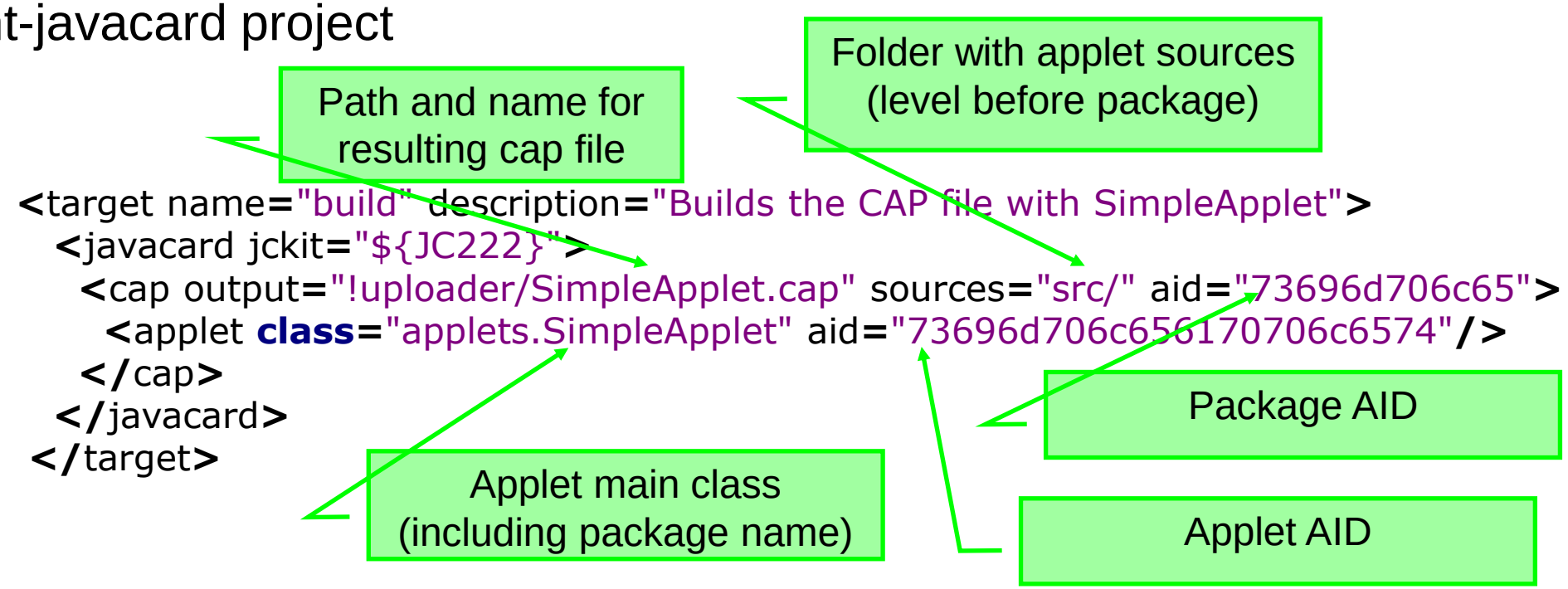
- Form groups of 3 students
- Brainstorm and write into Miro at least three concrete usage scenarios utilizing asymmetric cryptography where MPC can be used (be creative!)
  - https://miro.com/app/board/uXjVMf66usg=/?share_link_id=424368693160
  - Goals achieved, threshold configuration
- Pick the most interesting one and elaborate in more details
  - Describe process of group establishment, problems solved, comparison to single key scenario
- Some hints
  - RSA/ECDSA/Schnorr/EdDSA…
  - Document signing, authentication, collaborative decryption, key generation, PKI, single point of failure, unicorns, key distribution, …
  - k-of-n threshold, combination with secure hardware, temporary signers, cold-storage signers
  - Human participant, automated participant with policy, redundant participants, multiple shares by one participant…

CONVERSION AND UPLOAD TO REAL CARD

We will compile, convert and install SimpleApplet.cap

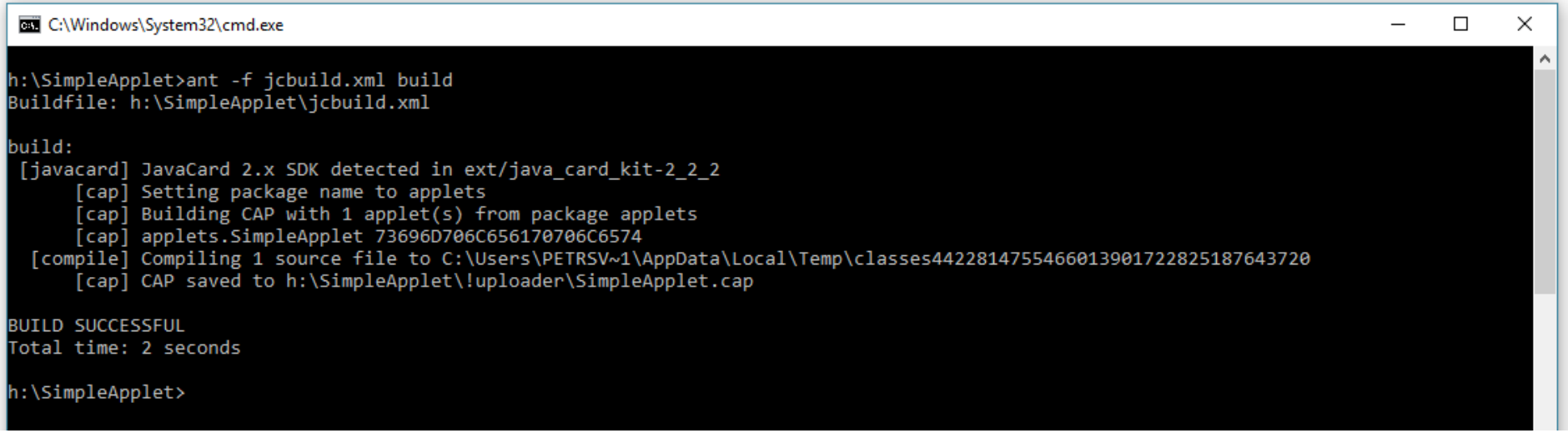
Task: Create cap file and upload to card

- Navigate to SimpleApplet folder
  - src folder contains applet's source code in SimpleApplet.java
  - **`jcbuild.xml`** contains configuration for conversion with ant-javacard project



Task: Create cap file and upload to card

- Compile & Convert
  - Execute on cmd line: **`ant -f jcbuild.xml build`**



- If OK, SimpleApplet.cap is created in !uploader folder

Task: Create cap file and upload to card

- http://github.com/martinpaljak/GlobalPlatformPro

1. List already loaded applets
   - `java -jar gp.jar -list -d`
2. Uninstall previous version of SimpleApplet
   - `java -jar gp.jar -uninstall SimpleApplet.cap -d`
3. Install SimpleApplet.cap
   - `java -jar gp.jar -install SimpleApplet.cap -d`
4. Use applet (commands in SimpleAPDU code)

Problem: what with other applets on card?

1. List already loaded applets
   - **`java -jar gp.jar -list -d`**
2. Find package_AID and run:
   - **`java -jar gp.jar -deletedeps -delete package_aid`**
   - The **`–deletedeps`** will also delete all applets from target package

- E.g., our SimpleApplet can be also removed by
  - **`gp -deletedeps -delete`** 73696d706c65

Be aware – real card can be blocked

- Too many unsuccessful authentication requests

```
>gp --list -debug
# Detected readers from SunPCSC
[*] Alcor Micro USB Smart Card Reader 0
SCardConnect("Alcor Micro USB Smart Card Reader 0", T=*) -> T=0, 3BF71800008031F
E45736674652D6E66C4
SCardBeginTransaction("Alcor Micro USB Smart Card Reader 0")
A>> T=0 (4+0000) 00A40400 00
A<< (0018+2) (56ms) 6F108408A0000000030000000A5049F6501FF 9000
A>> T=0 (4+0008) 80500000 08 6265E168FB2639C1
A<< (0028+2) (118ms) 00003126960097543174010200103595AC1420213D2969EA8B8C41F3 90
00
openkms.gp.GPException: STRICT WARNING: Card cryptogram invalid!
Card: 3D2969EA8B8C41F3
Host: DB1E6E1E71958A15
!!! DO NOT RE-TRY THE SAME COMMAND/KEYS OR YOU MAY BRICK YOUR CARD !!!
        at openkms.gp.GlobalPlatform.printStrictWarning(GlobalPlatform.java:156)

        at openkms.gp.GlobalPlatform.openSecureChannel(GlobalPlatform.java:471)
        at openkms.gp.GPTool.main(GPTool.java:348)
```

Be aware – real card can be blocked

- Don’t write script that executes many authentications at once (cycle, multiple commands)
- If unsuccessful one/two authentication is detected, then as for help, please!!!

Questions

- How can you list applets and packages available on card?
- How can you prevent people listing applets on your card?
- Why you need to remove applet first before installing updated version?

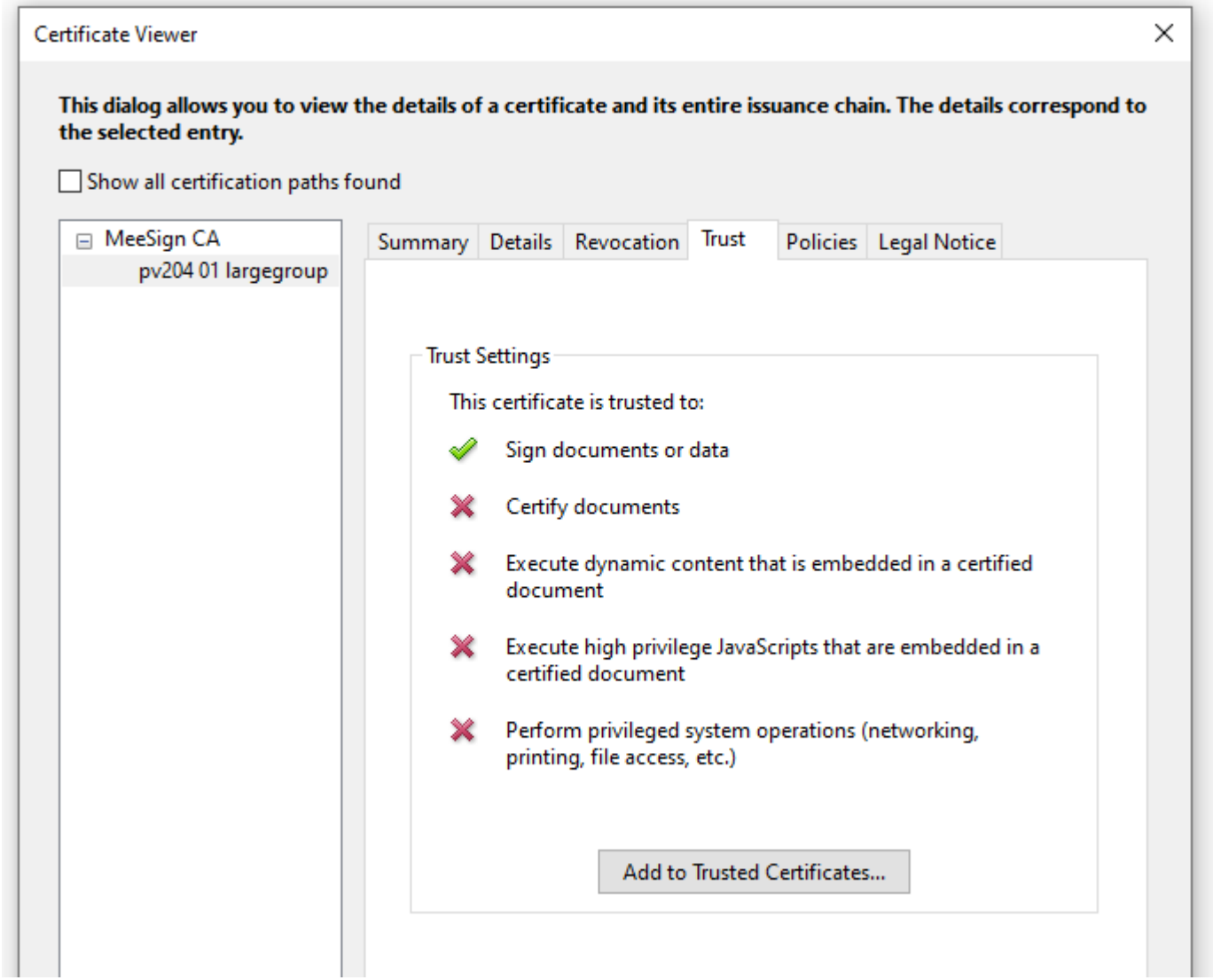
ADDING NEW JAVACARD FUNCTIONALITY

We will update, compile, convert and install SimpleApplet.cap

Tasks: add new “increment” method to applet

- Implement on-card Increment() method
  - All payload bytes from incoming apdu are incremented by one (separately)
  - Resulting array is returned back to host
- Add new constant for instruction INS_INC
- Add new method void Increment(APDU apdu) and its implementation
  - setIncomingAndReceive(), for loop over array, setOutgoingAndSend()
- Add method call into switch inside process() method
- Debug functionality with simulated card
- Compile, convert and upload updated applet to real card
- Change from simulator to real card
  - runCfg.setTestCardType(RunConfig.**CARD_TYPE.PHYSICAL**);
- Test functionality using real card

NO ASSIGNMENT THIS WEEK ☺


Certificate Viewer
This dialog allows you to view the details of a certificate and its entire issuance chain. The details correspond to the selected entry.
Show all certification paths found
MeeSign CA
pv204 01 largegroup
Summary
Details
Revocation
Trust
Policies
Legal Notice
Trust Settings
This certificate is trusted to:
Sign documents or data
Certify documents
Execute dynamic content that is embedded in a certified document
Execute high privilege JavaScripts that are embedded in a certified document
Perform privileged system operations (networking, printing, file access, etc.)
Add to Trusted Certificates...