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/decryption (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/
  - Alternatively use the provided VM http://is.muni.cz/go/meesign-vm
- Connect to university network => wlan-fi, eduroam, vpn
- 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

Troubleshooting

- Missing link to libssl.so 1.1 - https://stackoverflow.com/questions/72133316/libssl-so-1-1-cannot-open-shared-object-file-no-such-file-or-directory

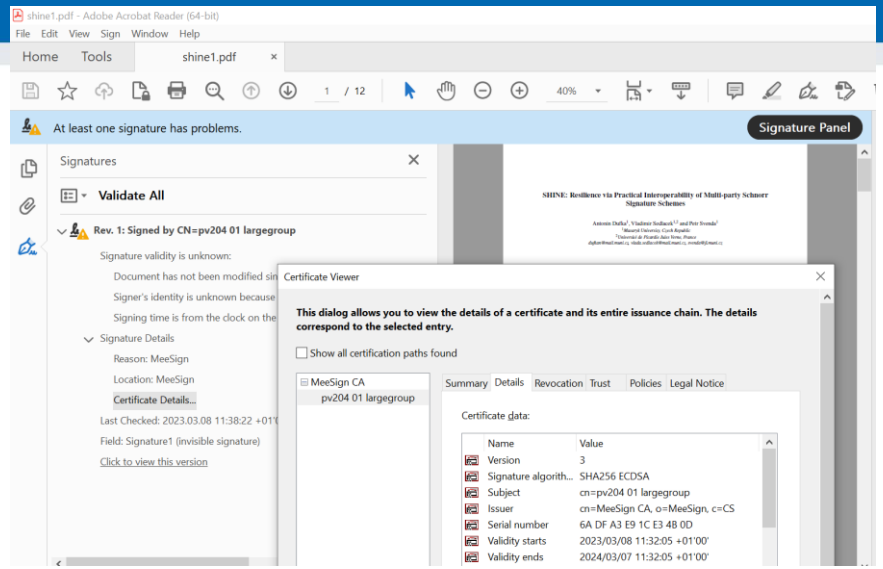
- You can run multiple clients on single machine
  - Download binary from https://meesign.crocs.fi.muni.cz/
  - Create first, second, third… new folder, extract binary there
  - Run meesign_client from each folder

Task1: 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)

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.
```

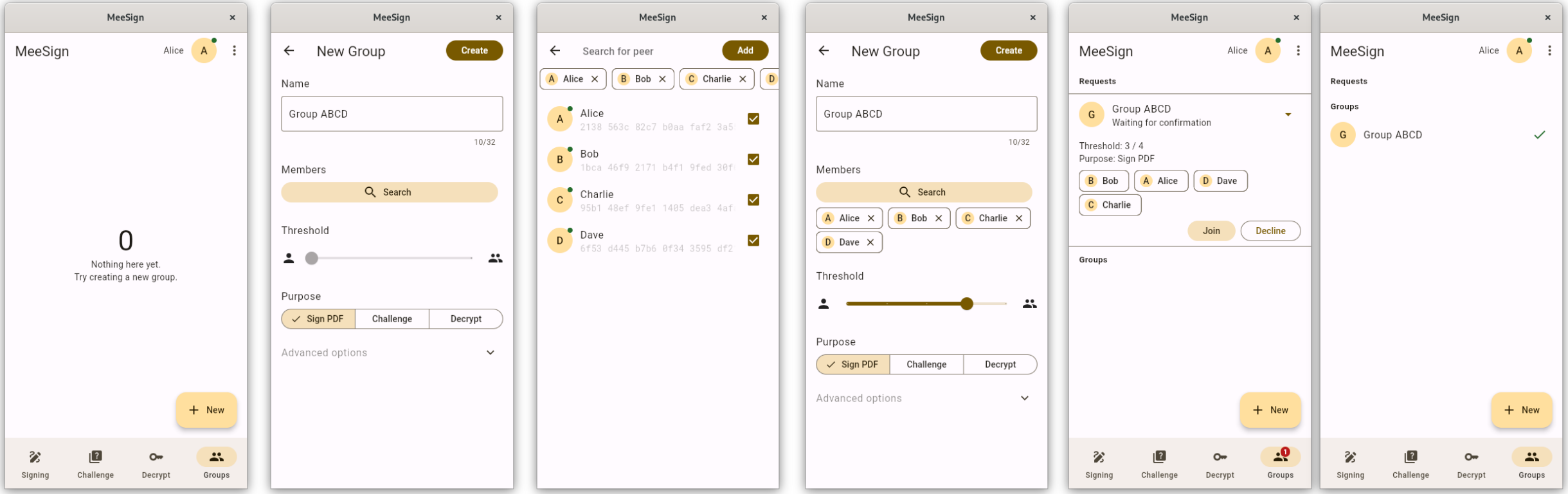
Cryptographic Verification : PASSED
Has the reference data object been found?
Is the reference data object intact?
Is the signature intact?
Signature Acceptance Validation : PASSED
Is the structure of the signature valid?
Is the signed attribute: 'signing-certificate' present?
Is the signed qualifying 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 2: Signing in smaller group

- Groups of 3-4 students (4 devices)
- Create new group with some unique name, add yourself and peers
  - Select purpose as ‘Sign PDF’
  - Try to add peers via qrcode
    - Users display qrcode (upper right corner)
    - Group creator Add member → Scan
  - Set threshold to 3-of-4
- Initiate MPC signing, sign, view document
- Check yourself the resulting MPC signature
  - Adobe Acrobat Reader or https://ec.europa.eu/digital-building-blocks/DSS/webapp-demo/validation

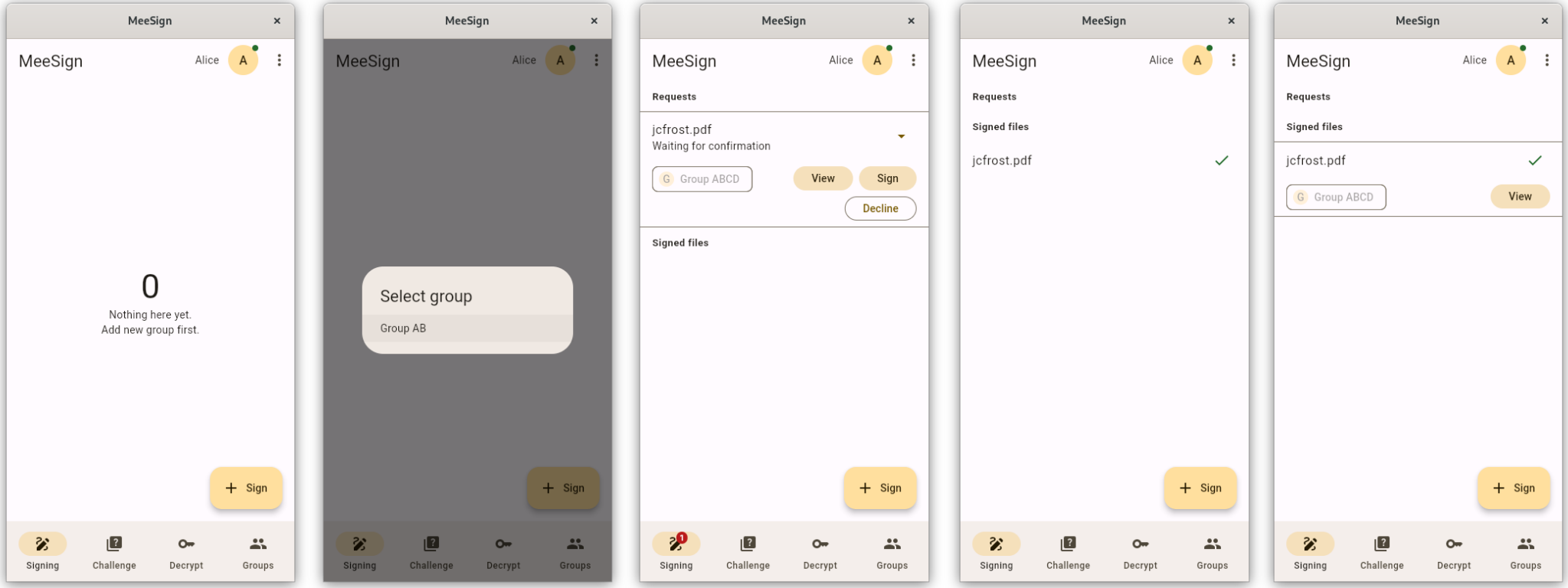
Task 2: Setting up a threshold

- Create a 3-of-4 group for PDF signing



Task 2: Setting up a threshold

- Create a signature of an arbitrary PDF with the group



Task 2: Questions

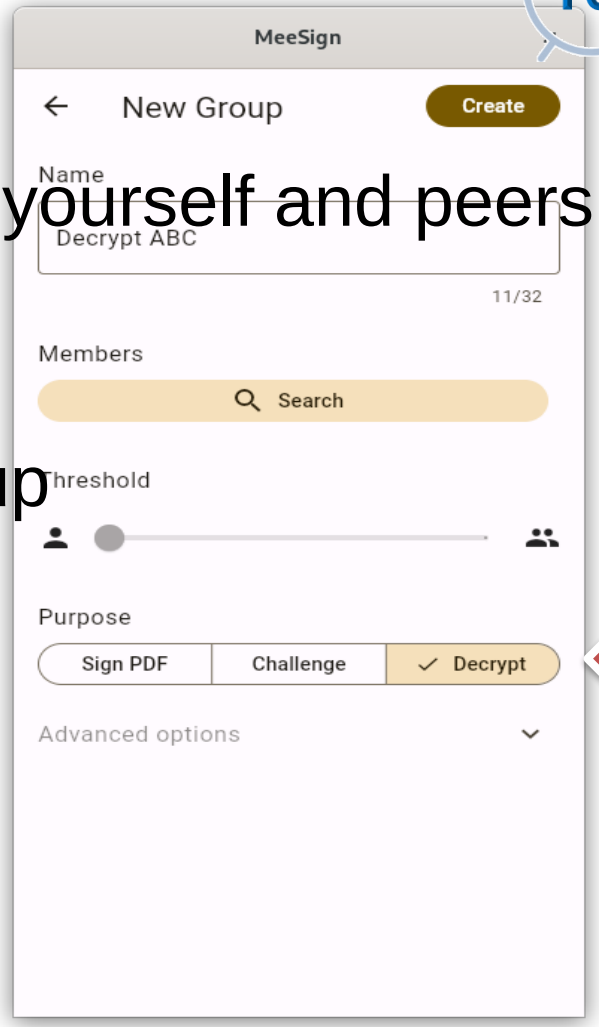
- How did the group creation behave?
  - Did all of you need to approve it? Why?
- How did the signing behave?
  - Did all of you need to approve it? Why?
- What do you think lawyers think about such signature?
  - Accountability?
- Can you set 1-of-n? Why not? What would it mean for the private key?
  - What does it provide beyond a regular single-party signature?
  - Do you have any ideas how it could be used?

Task 2: 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?

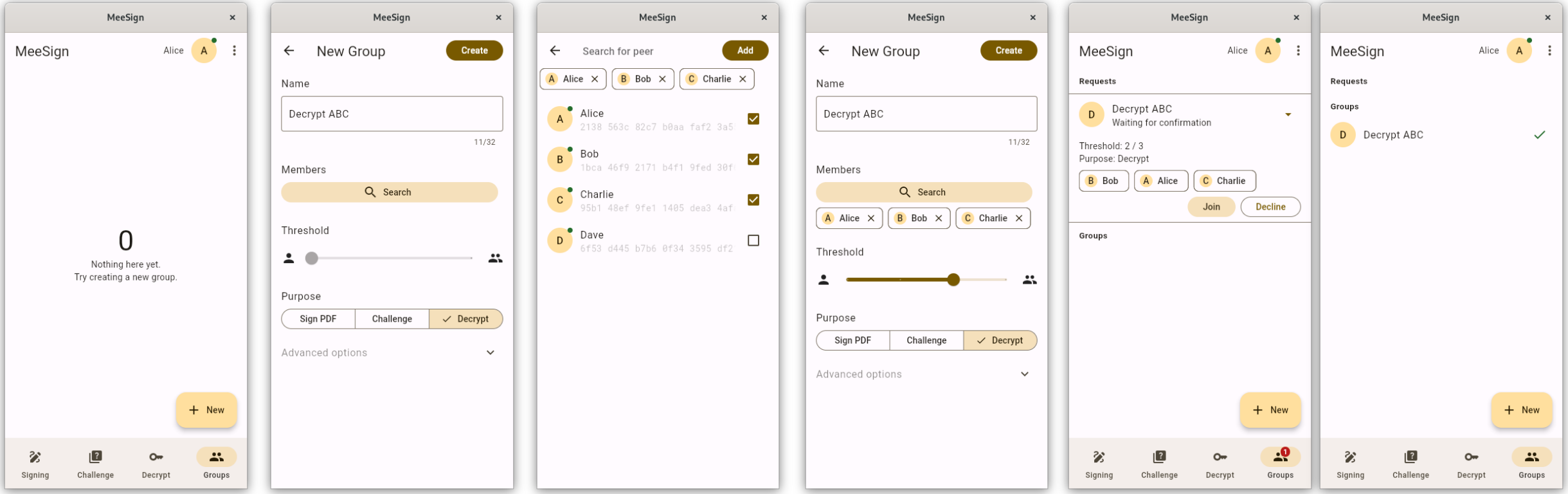
Task 3: Multiparty decryption

- Keep groups of 3-4 students (3 devices)
- Create new group with some unique name, add yourself and peers
  - Select purpose of group to ‘Decrypt’
  - Set threshold 2-of-3
- One sends encrypted image or message to group
- Others decrypt and view result



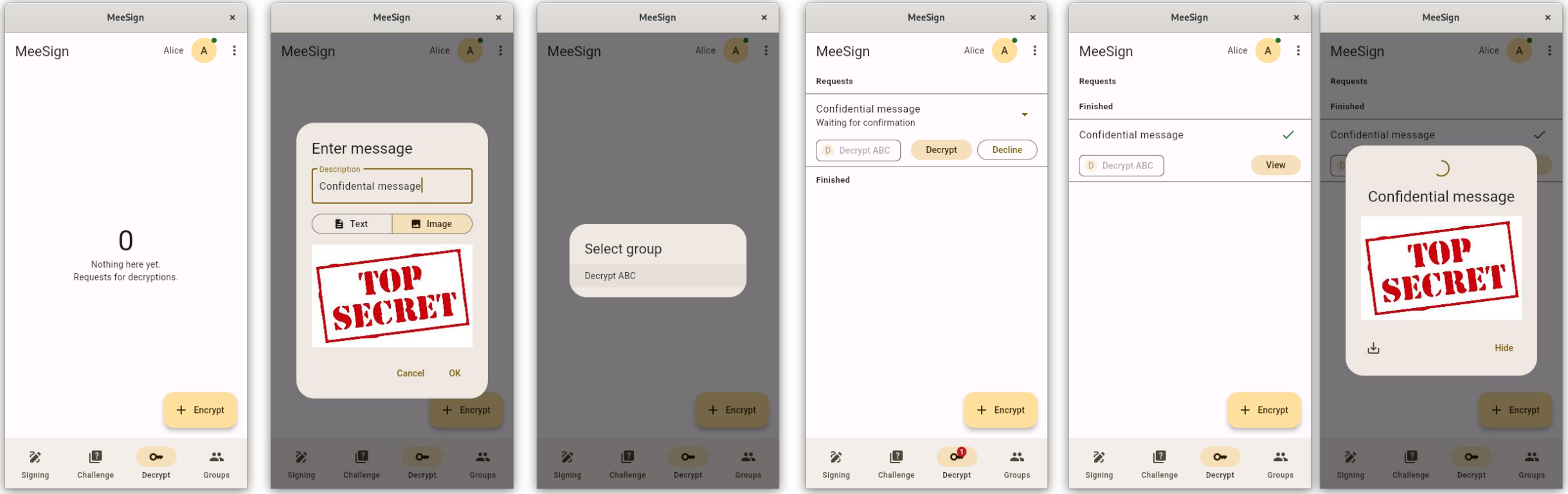
Task 3: Multi-party decryption

- It is not just signing – create a decryption group



Task 3: Multi-party decryption

- Send an encrypted image or message to the group



Task 3: Questions - Multi-party decryption

- Can everyone in the group see the decrypted message?
- Who can see the decrypted picture?

Task: Brainstorm interesting usages for MPC

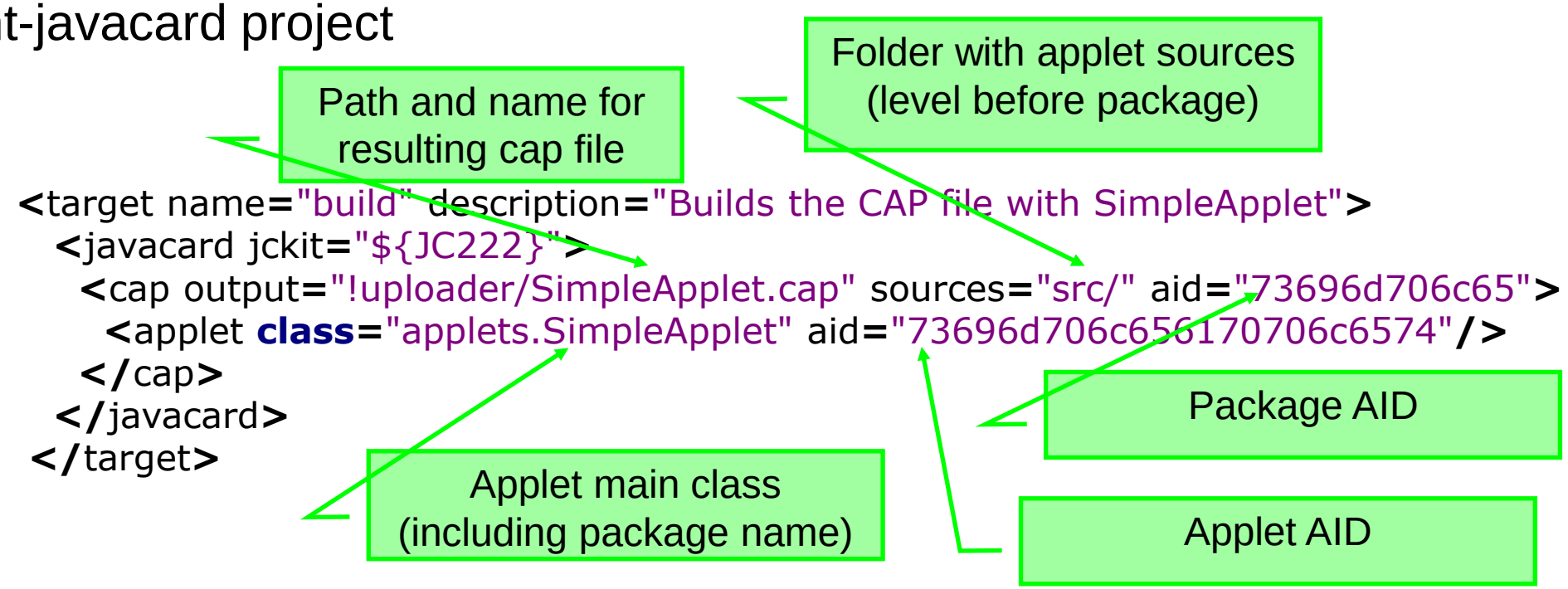
- Form groups of 3 students
- Brainstorm and write into Miro at least three concrete usage scenarios utilizing asymmetric cryptography where MPC can be used (be creative!)
  - https://miro.com/app/board/uXjVMf66usg=/?share_link_id=424368693160
  - Goals achieved, threshold configuration
- Pick the most 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, only machine participants, asynchronous participants, timelocks…

CONVERSION AND UPLOAD TO REAL CARD

We will compile, convert and install SimpleApplet.cap

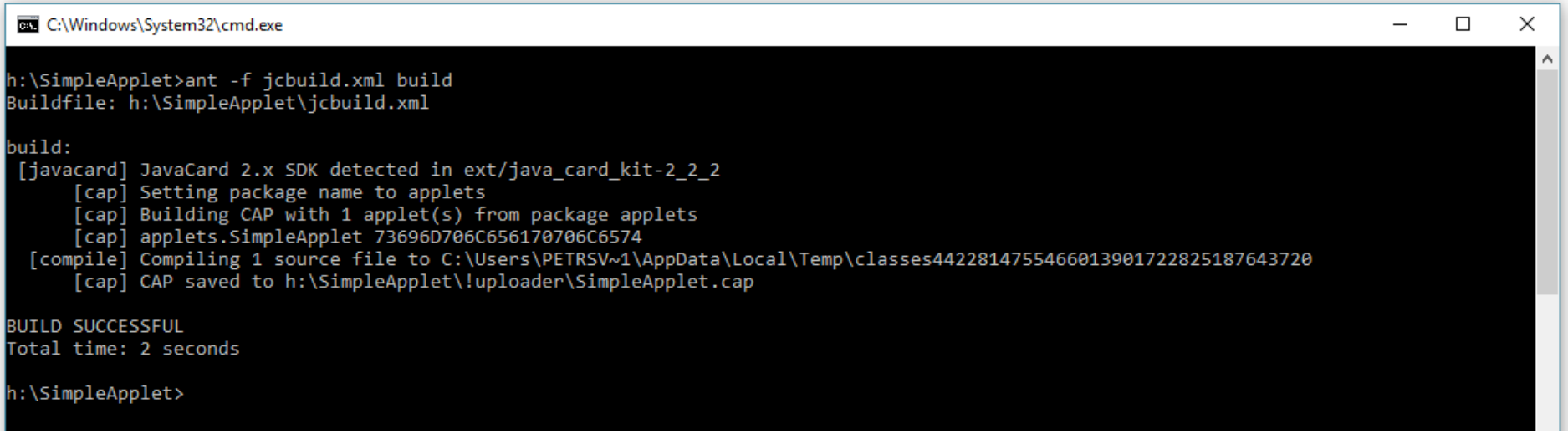
Task: Create cap file and upload to card

- 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

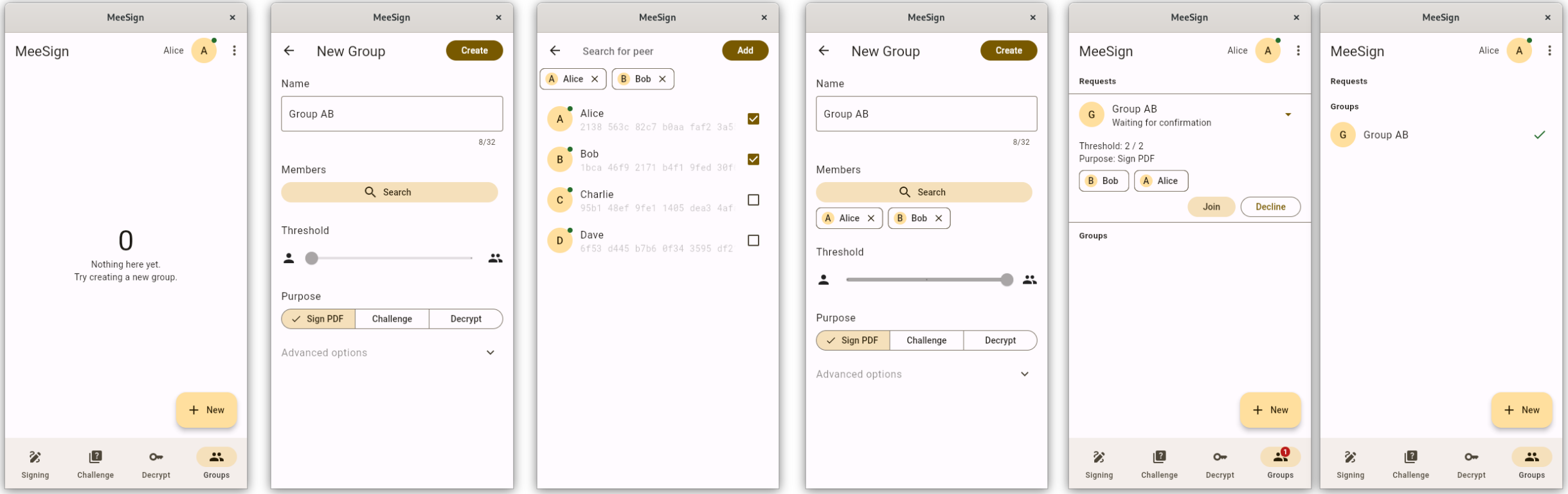
- Implement on-card Increment() method
  - All payload bytes from incoming apdu are incremented by one (separately)
  - Resulting array is returned back to host
- Add new constant for instruction INS_INC
- Add new method void Increment(APDU apdu) and its implementation
  - setIncomingAndReceive(), for 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 ☺

SIMPLE EXAMPLE 2-OF-2 GROUP

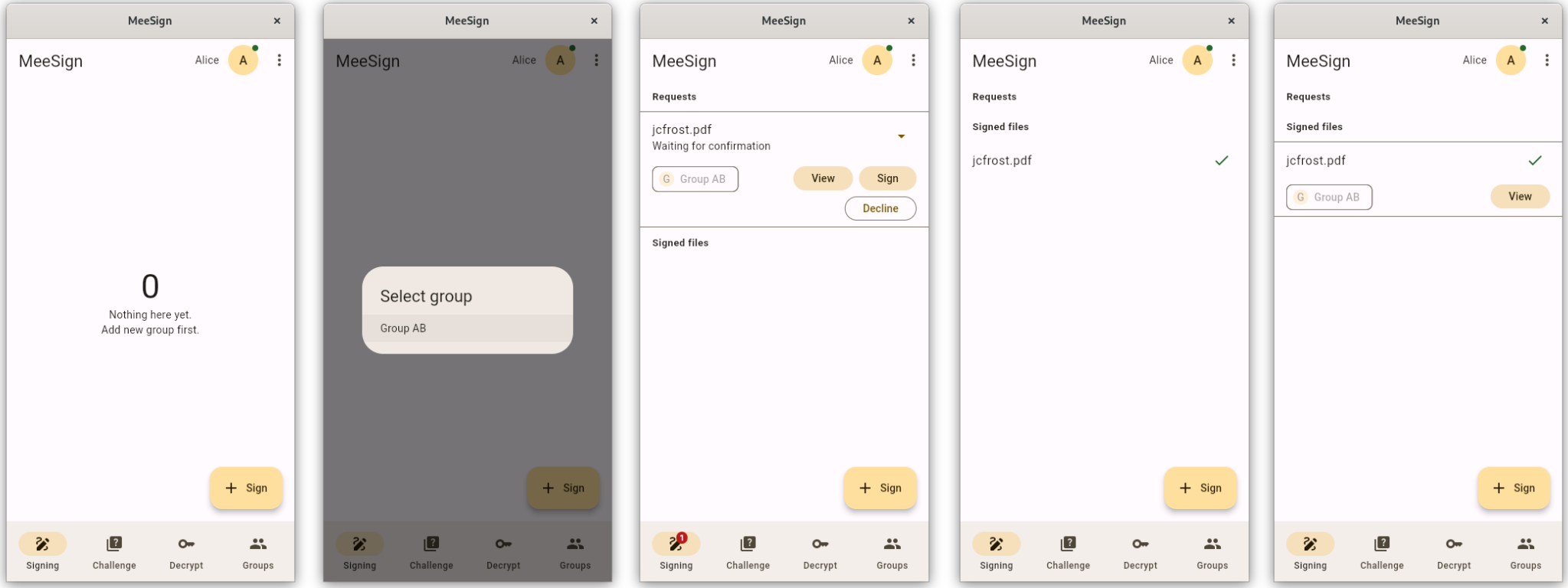
Example: Multi-party PDF signing

- Create a 2-of-2 group for PDF signing



Example: Multi-party PDF signing

- Create a signature of an arbitrary PDF with the group



Example: Multi-party PDF signing

- Verify the signature
  - Why is it not trusted?

```
↳ pdfsig jcfrost.signed.pdf
Digital Signature Info of: jcfrost.signed.pdf
Signature #1:
  - Signature Field Name: Signature1
  - Signer Certificate Common Name: Group AB
  - Signer full Distinguished Name: CN=Group AB
  - Signing Time: Mar 11 2024 14:04:38
  - Signing Hash Algorithm: SHA-256
  - Signature Type: adbe.pkcs7.detached
  - Signed Ranges: [0 - 932569], [951515 - 952103]
  - Total document signed
  - Signature Validation: Signature is Valid.
  - Certificate Validation: Certificate issuer isn't Trusted.
```