Techniques for Authentication of Mobile Devices

Cybersecurity for Embedded Systems

Track 9 – Group b

Politecnico di Torino

1859

PERNO ANDREA – 296185

SCIARA LORENZO – 303462

SCICOLONE OMAR – 296492

SPINELLO BASILIO – 292856

Goal of the project

- Prove data origin from a specific device to an external verifier
- Utilize unique properties of the device (codes, IDs, etc.)
- Achieve "beyond reasonable doubt" data authentication
- Develop a protocol meeting the goals

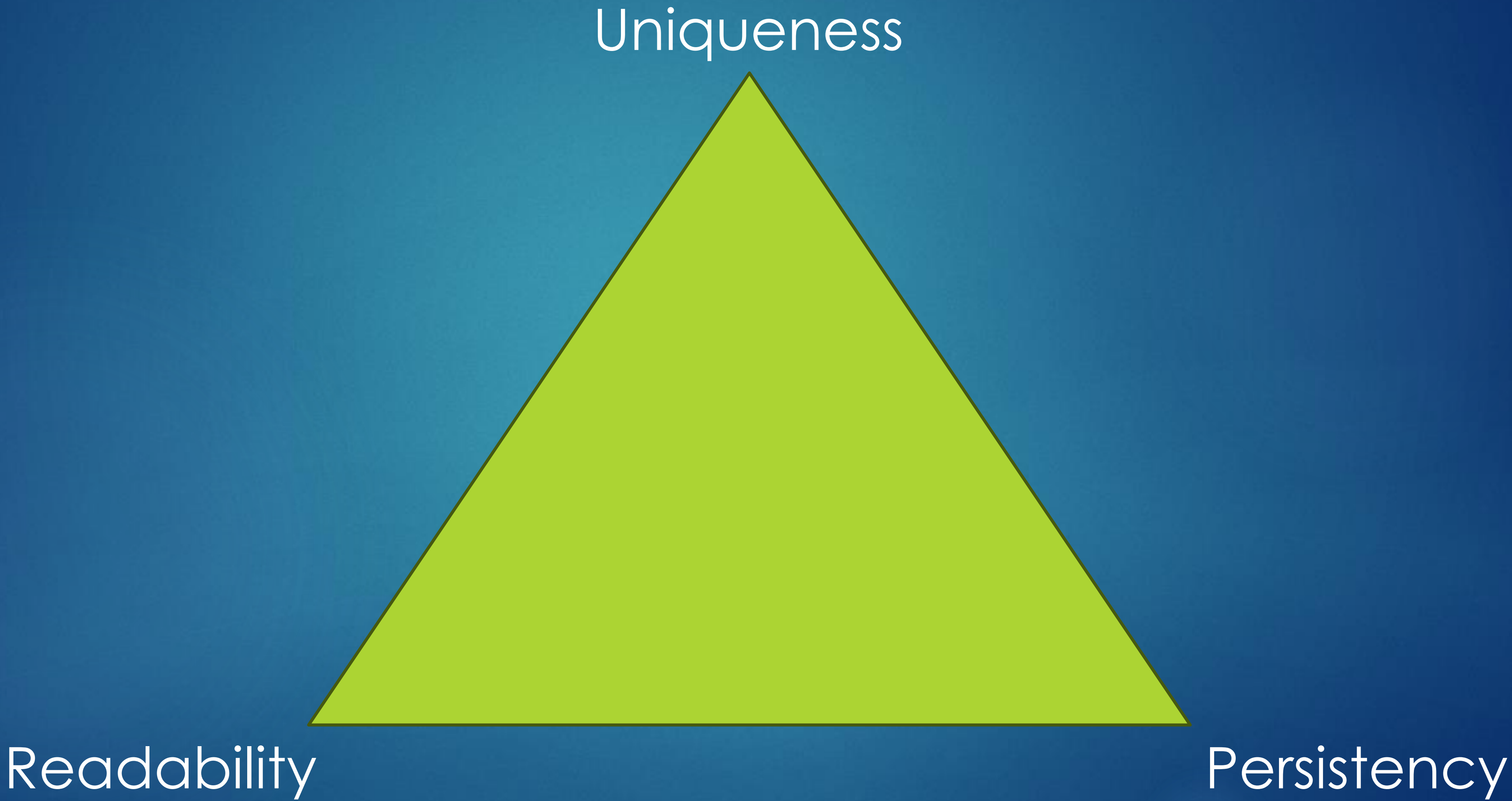
Assumption:

Verifier has access to the device to confirm authenticity.

Unique IDs

Hardware	Software	SIM related
IMEI	Advertising ID	ICCID
Serial Number	UUID	IMSI
MAC	Firebase ID	PhoneNumber
	DRM	

Decision Criteria

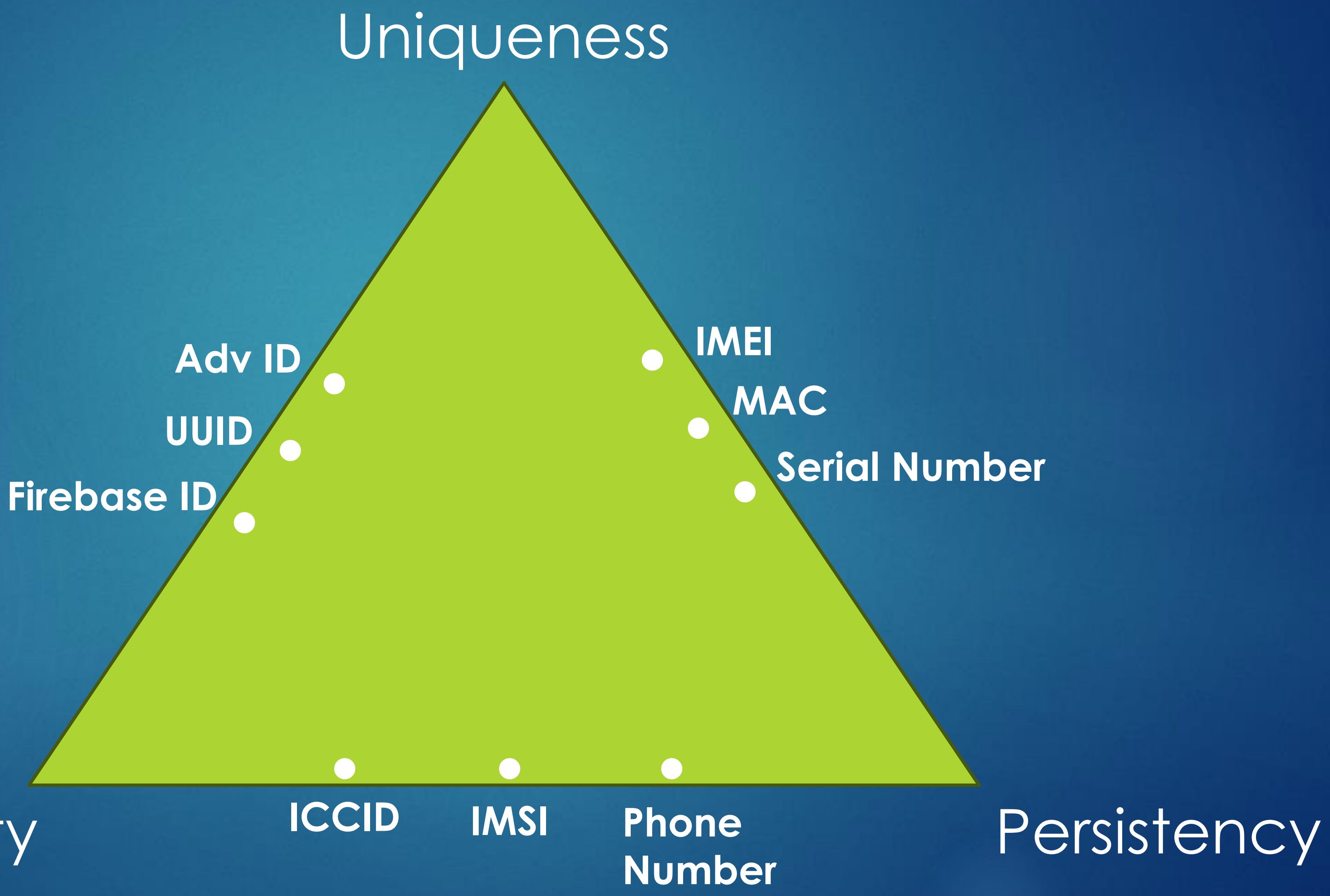


Uniqueness

Readability

Persistency

Decision Criteria



Best practices for unique identifiers

«Avoid using hardware identifier, choose instead user-resettable identifiers whenever possible»

(from Android Documentation)

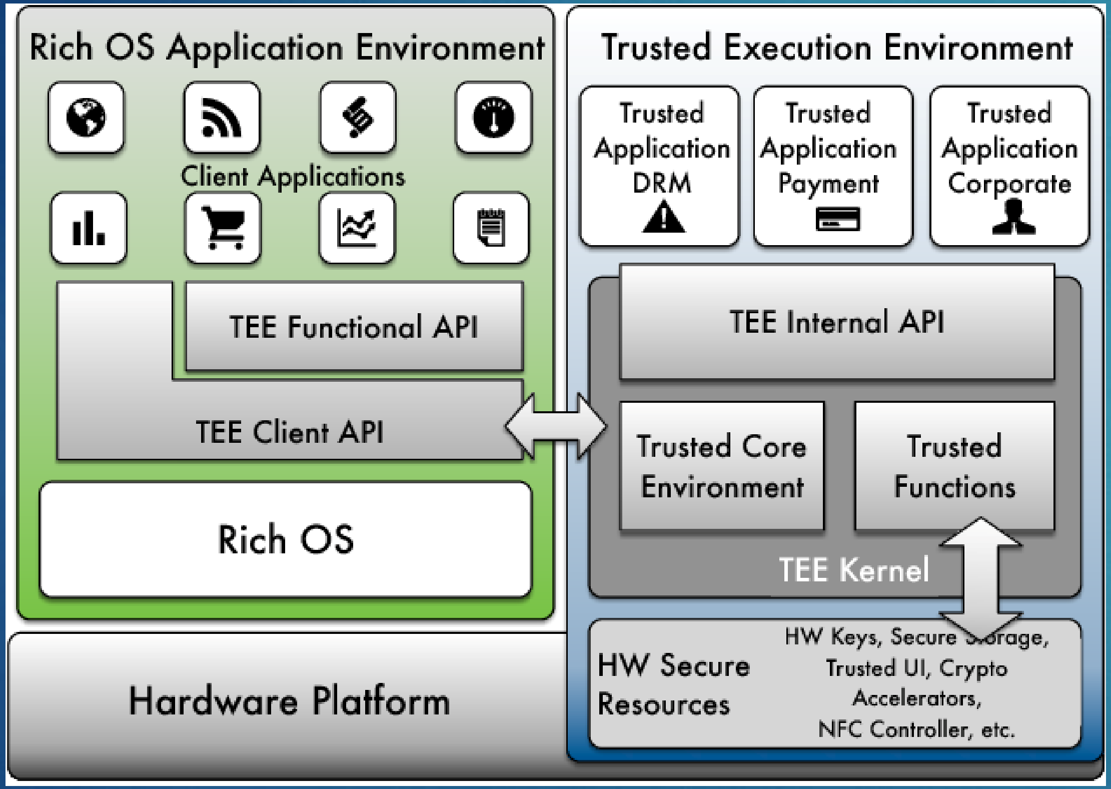
READ_PRIVILEGED_PHONE_STATE

Our approach

- Cryptographic operations of the smartphone
- Trusted Execution Environment (TEE) and Android Keystore System
- TEE-based Hardware-backed Key Attestation

Management and analysis of cryptographic keys and certificates

Trusted Execution Environment



Rich OS Application Environment
Client Applications
TEE Functional API
TEE Client API
Rich OS
Hardware Platform
Trusted Execution Environment
Trusted Application DRM
Trusted Application Payment
Trusted Application Corporate
TEE Internal API
Trusted Core Environment
Trusted Functions
TEE Kernel
HW Secure Resources
HW Keys, Secure Storage, Trusted UI, Crypto Accelerators, NFC Controller, etc.

Main properties:

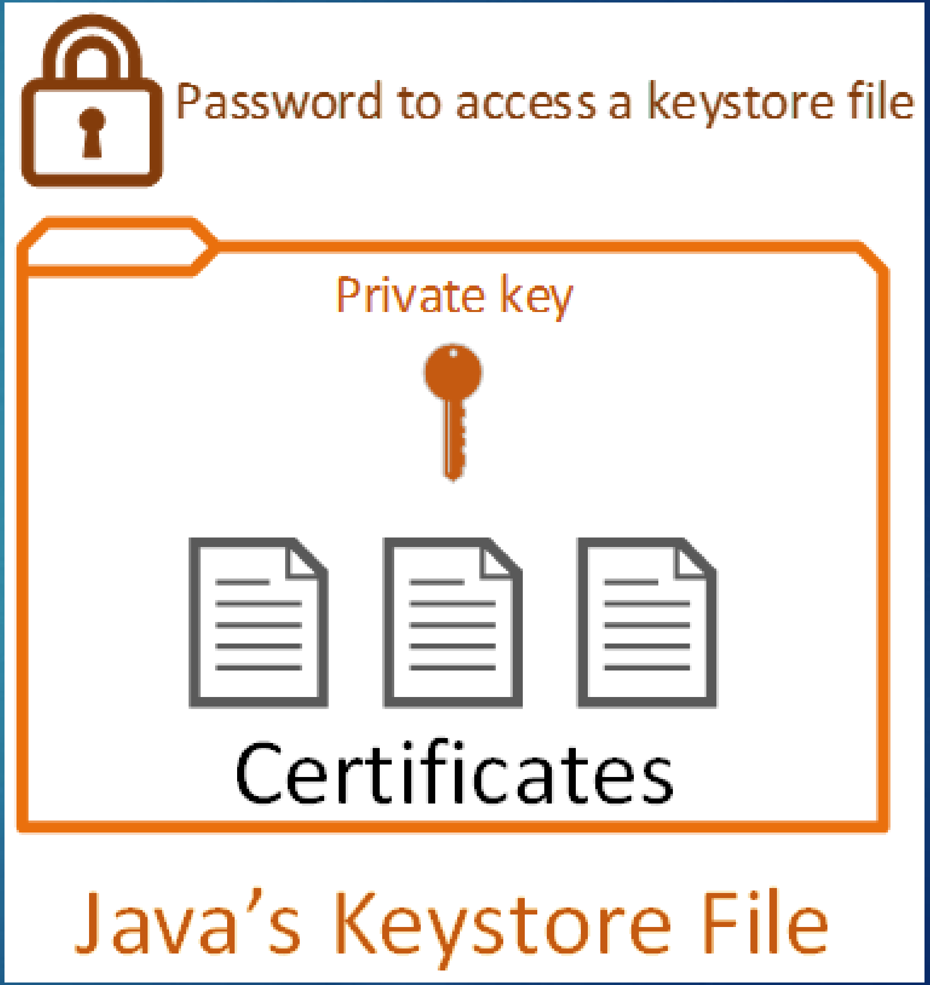
- Hardware isolation
- Secure Data Storage
- Cryptographic engine
- TEE-backed unique Key pair

Android KeyStore

Android abstraction of TEE

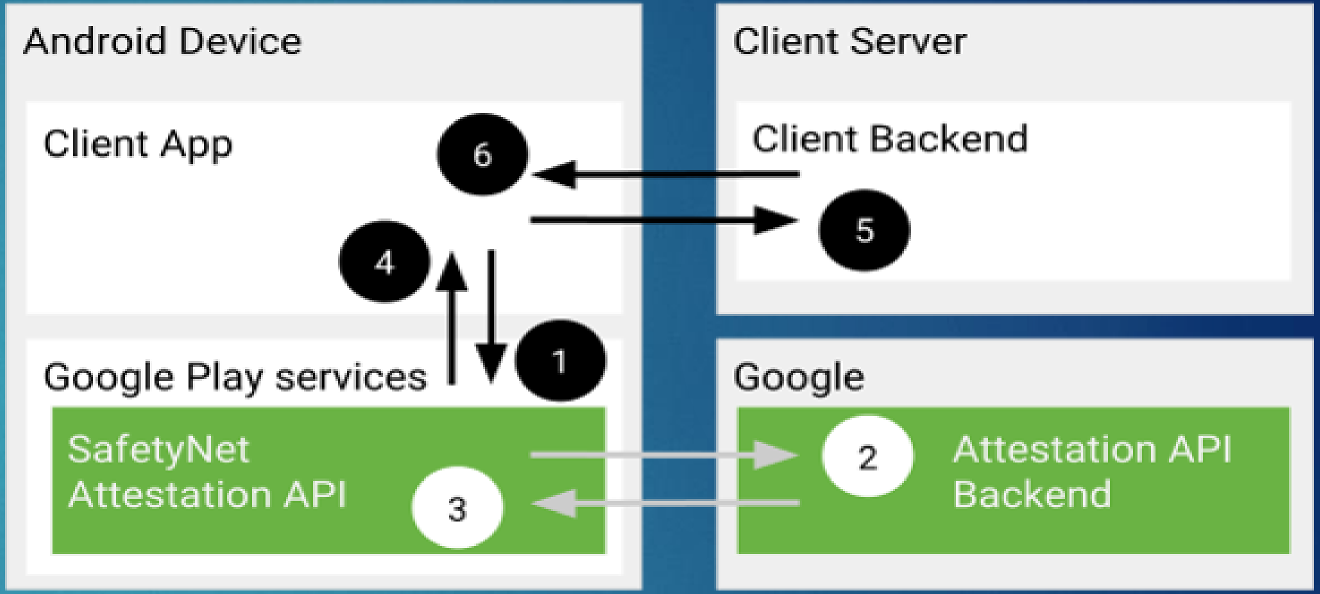
Main properties:

- Secure key matherial storage
- Prevention from unauthorized usage of the keys
- Cryptographic process separation
- Bounded to the TEE



Android Key Attestation

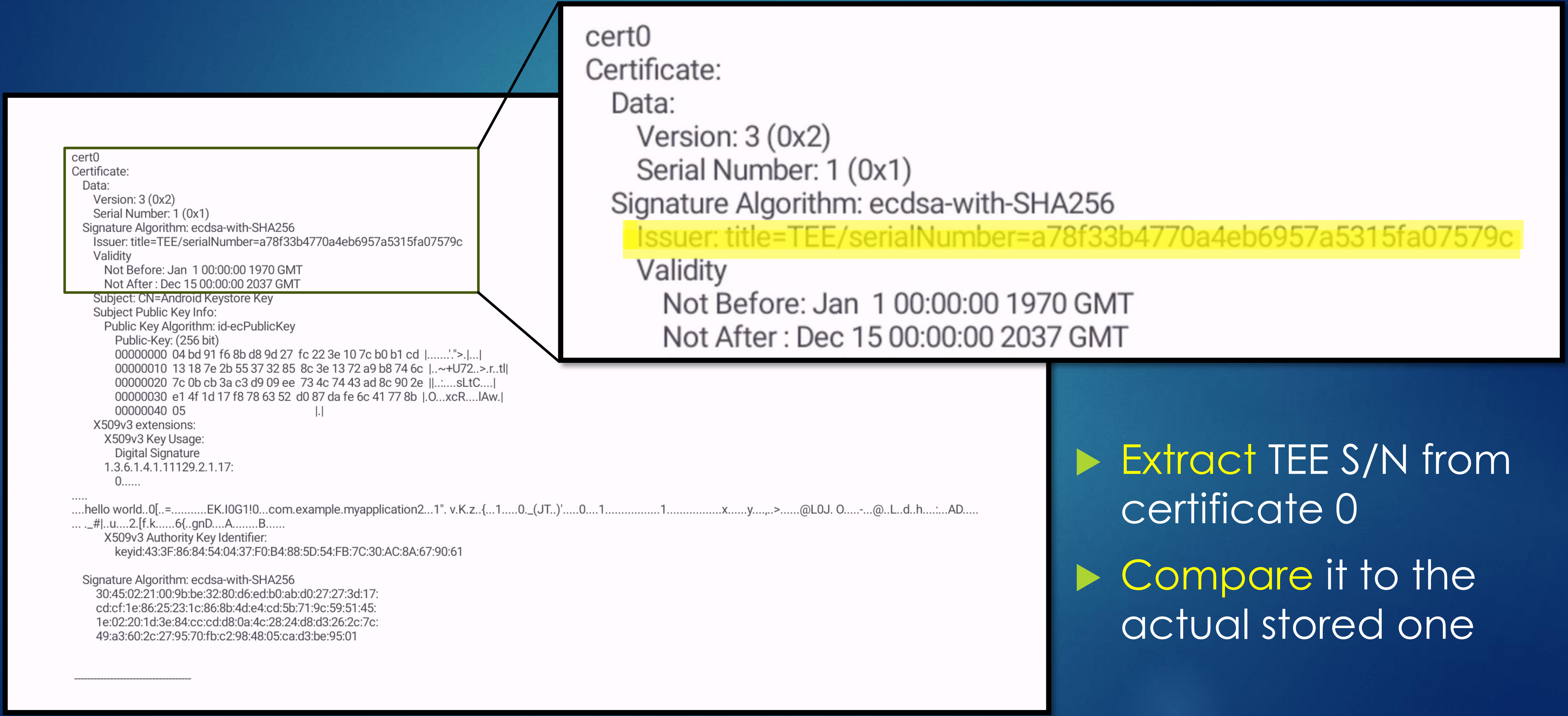
- Each key pair have a Certificate Chain
- Export certificate chain to the verification server



Verify Certificate Chain

Verify Digital Signature for sender authentication

TEE Serial Number

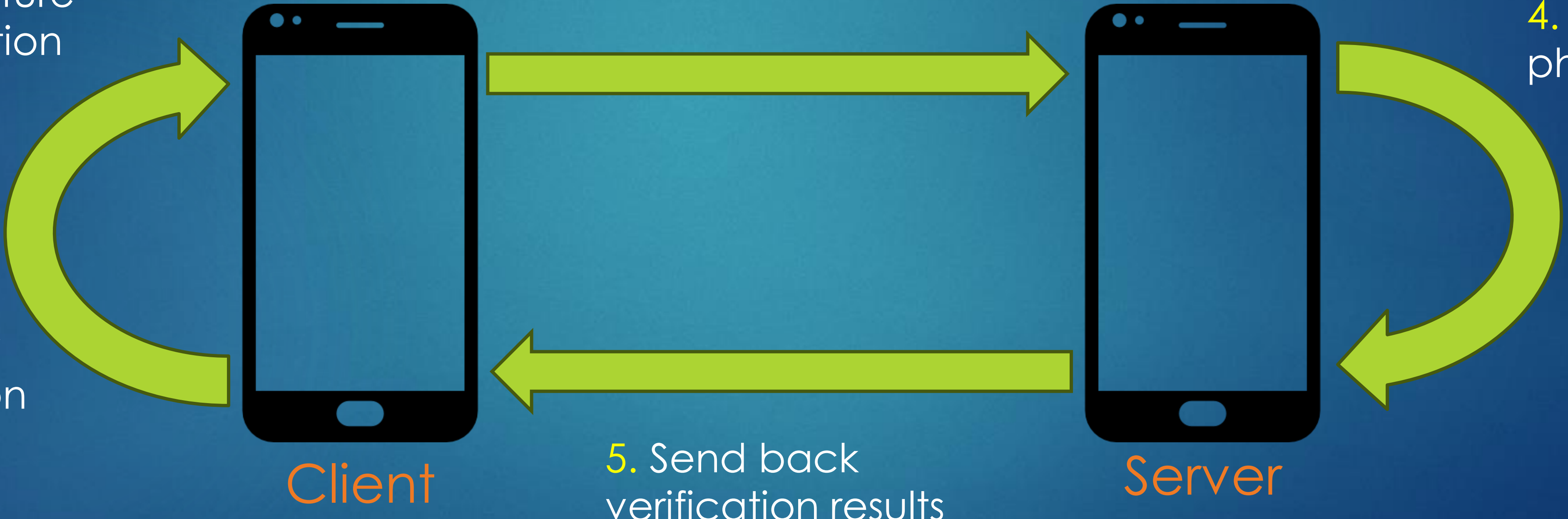


```
cert0
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 1 (0x1)
  Signature Algorithm: ecdsa-with-SHA256
    Issuer: title=TEE/serialNumber=a78f33b4770a4eb6957a5315fa07579c
    Validity
      Not Before: Jan  1 00:00:00 1970 GMT
      Not After : Dec 15 00:00:00 2037 GMT
    Subject: CN=Android Keystore Key
    Subject Public Key Info:
      Public Key Algorithm: id-ecPublicKey
        Public-Key: (256 bit)
        00000000  04 bd 91 f6 8b d8 9d 27  fc 22 3e 10 7c b0 b1 cd  |.......'.">.|...|
        00000010  13 18 7e 2b 55 37 32 85  8c 3e 13 72 a9 b8 74 6c  |..~+U72..>.r..tl|
        00000020  7c 0b cb 3a c3 d9 09 ee  73 4c 74 43 ad 8c 90 2e  ||..:....sLtC....|
        00000030  e1 4f 1d 17 f8 78 63 52  d0 87 da fe 6c 41 77 8b  |.O...xcR....lAw.|
        00000040  05                                               |.|
    X509v3 extensions:
      X509v3 Key Usage:
        Digital Signature
      1.3.6.1.4.1.11129.2.1.17:
        0......
.....
....hello world..0[..=..........EK.I0G1!0...com.example.myapplication2...1". v.K.z..{...1.....0._(JT..)'.....0....1................1................x......y....,.>......@L0J. O.....-...@..L..d..h....:...AD.....
... ._#|..u....2.[f.k......6{..gnD....A........B......
      X509v3 Authority Key Identifier:
        keyid:43:3F:86:84:54:04:37:F0:B4:88:5D:54:FB:7C:30:AC:8A:67:90:61

  Signature Algorithm: ecdsa-with-SHA256
     30:45:02:21:00:9b:be:32:80:d6:ed:b0:ab:d0:27:27:3d:17:
     cd:cf:1e:86:25:23:1c:86:8b:4d:e4:cd:5b:71:9c:59:51:45:
     1e:02:20:1d:3e:84:cc:cd:d8:0a:4c:28:24:d8:d3:26:2c:7c:
     49:a3:60:2c:27:95:70:fb:c2:98:48:05:ca:d3:be:95:01


  --------------------------------
```

```
cert0
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 1 (0x1)
  Signature Algorithm: ecdsa-with-SHA256
    Issuer: title=TEE/serialNumber=a78f33b4770a4eb6957a5315fa07579c
    Validity
      Not Before: Jan  1 00:00:00 1970 GMT
      Not After : Dec 15 00:00:00 2037 GMT
```

- Extract TEE S/N from certificate 0
- Compare it to the actual stored one

Implementation Schema



1. KeyPair generation

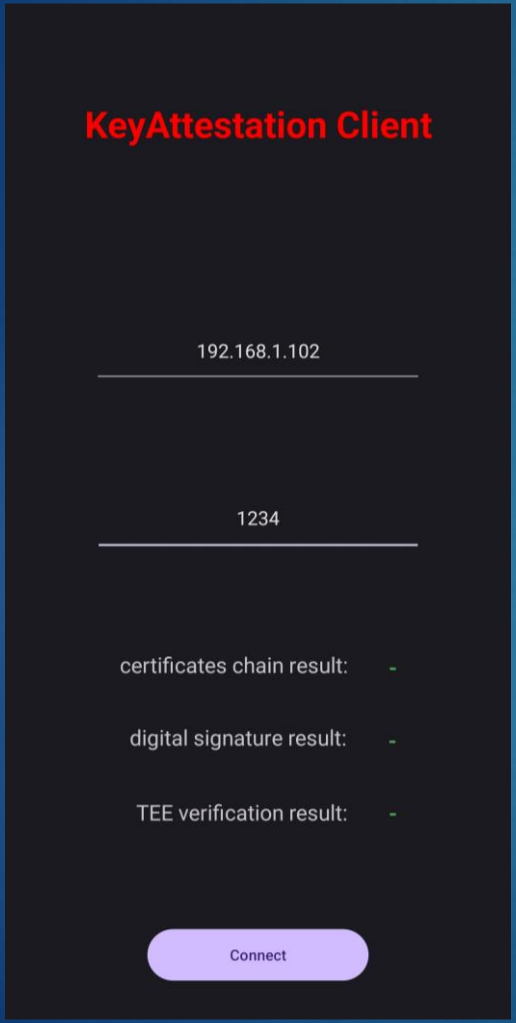
2. Signature generation

3. Send computed Signature and Key certificate chain

4. Verification phase

5. Send back verification results

Client Implementation



KeyAttestation Client

192.168.1.102

1234

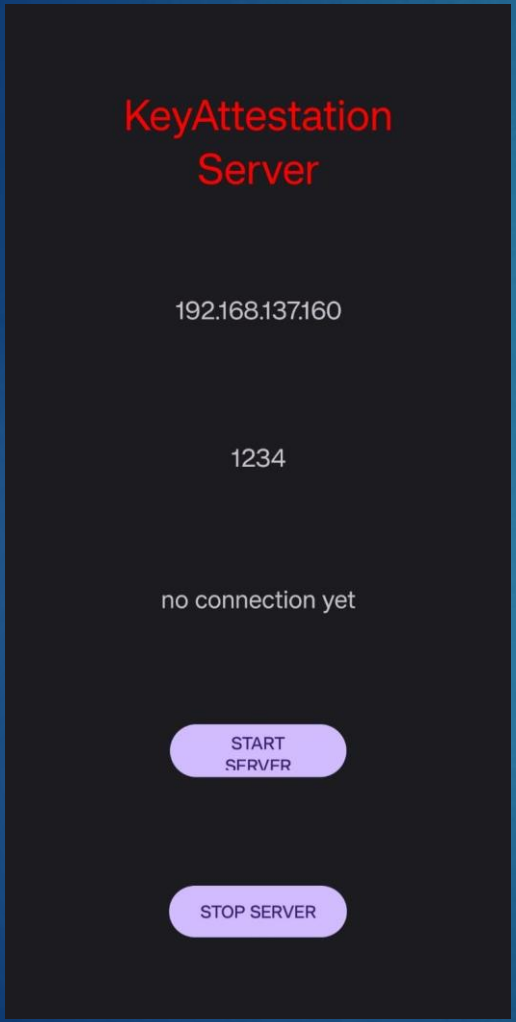
certificates chain result: -

digital signature result: -

TEE verification result: -

1. Generation of the Key Pair
2. Sign data with the private key
3. Send signature and Certificate Chain to the Server
4. Wait for response

Server Implementation



KeyAttestation Server

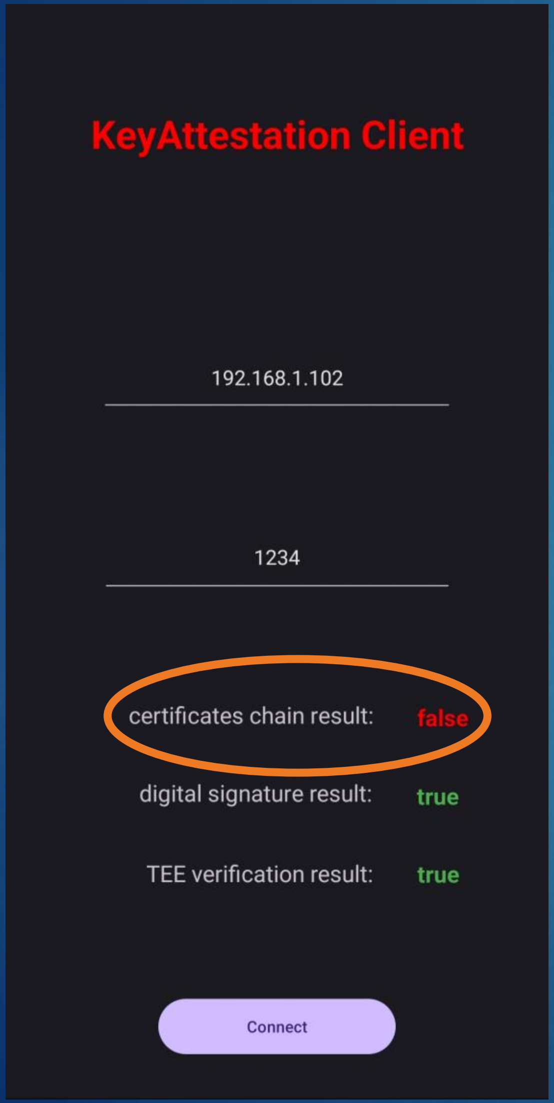
192.168.137.160

1234

no connection yet

1. Receives signature and Certificate Chain
2. For each certificate:
   1. Check validity
   2. Check integrity
   3. Check revocation status
3. Check signature value
4. Compare TEE Serial Number
5. Send results

Demo - Client Errors



KeyAttestation Client

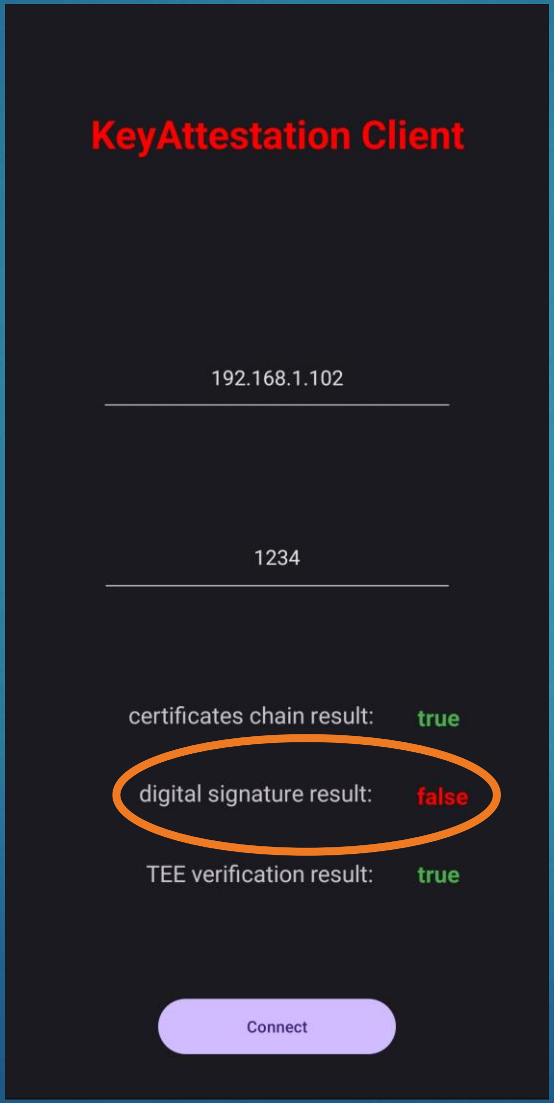
192.168.1.102

1234

certificates chain result: false

digital signature result: true

TEE verification result: true



KeyAttestation Client

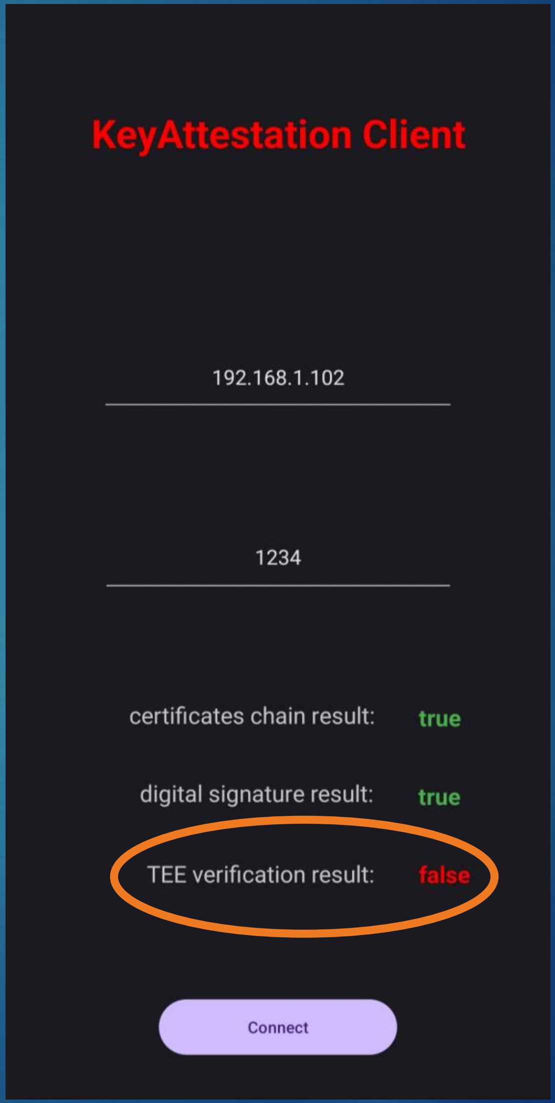
192.168.1.102

1234

certificates chain result: true

digital signature result: false

TEE verification result: true



KeyAttestation Client

192.168.1.102

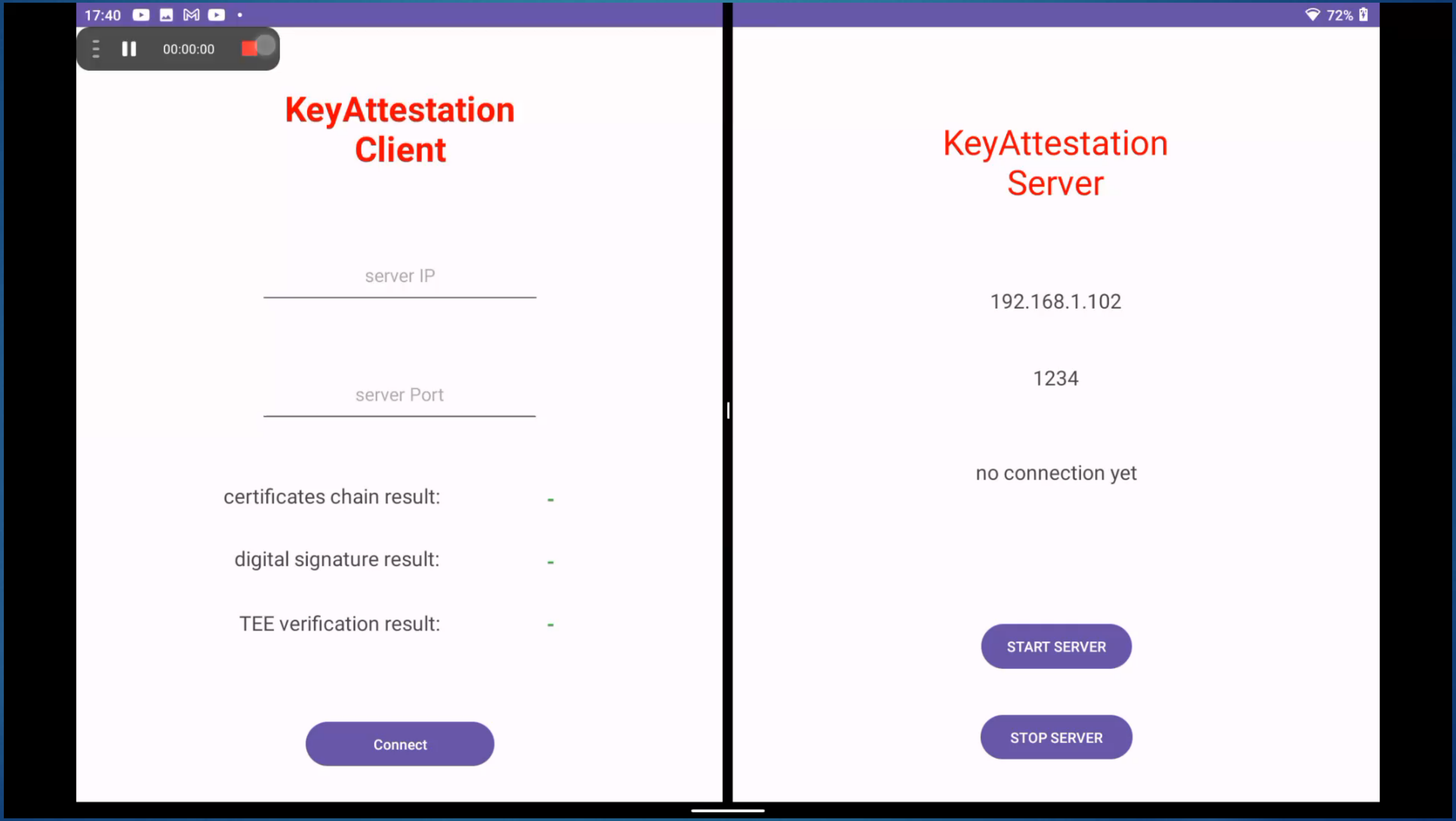
1234

certificates chain result: true

digital signature result: true

TEE verification result: false

Demo - Video



17:40

00:00:00

72%

KeyAttestation Client

server IP

server Port

certificates chain result: -

digital signature result: -

TEE verification result: -

Connect

KeyAttestation Server

192.168.1.102

1234

no connection yet

START SERVER

STOP SERVER

Summary

We are able to:

- Client side:
  - Generate a Key Pair for asymmetric digital signature
  - Retrieve Public Key certificate chain
  - Sign some data
- Server side:
  - Verify the integrity of data and certificate chain
  - **Authenticate the sender device with TEE S/N**

Possible Extensions

- TEE Serial Number database
- Signature of generic data
- Automatic identification of the Server (private network)
  - Extension to public IP network

Bibliography

- Android Identifiers:

  https://en.proft.me/2017/06/13/how-get-unique-id-identify-android-devices/

  https://ehsanet.medium.com/android-unique-device-id-history-and-updates-7667b38e4ee2
- Best Practices for Android unique identifiers:

  https://developer.android.com/training/articles/user-data-ids
- Android KeyStore:

  https://developer.android.com/training/articles/keystore
- Android Key Attestation:

  https://developer.android.com/training/articles/security-key-attestation

Thanks for the attention!

Politecnico di Torino

1859

PERNO ANDREA – 296185

SCIARA LORENZO – 303462

SCICOLONE OMAR – 296492

SPINELLO BASILIO – 292856