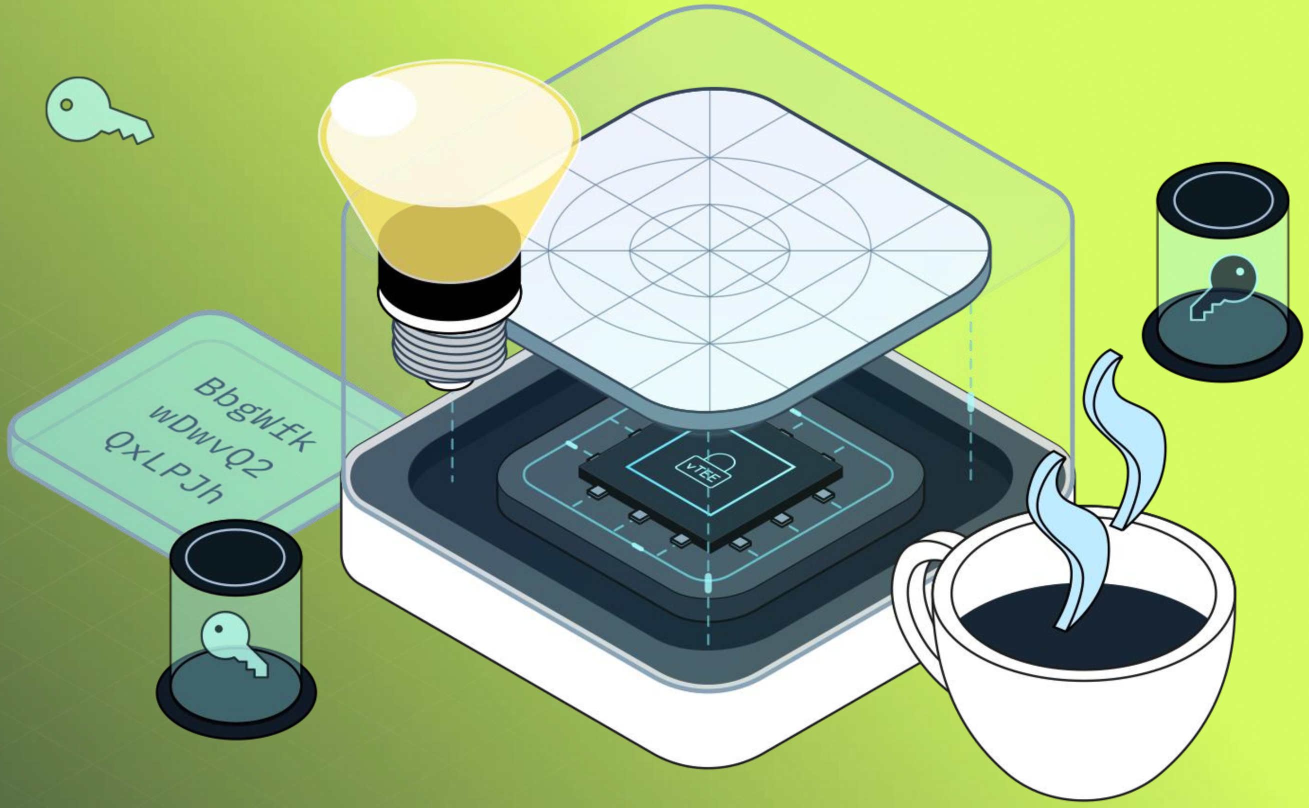
Licel Droidcon Hackathon

Come for the challenge.
Stay for the community!



How will the Hackathon work?

1. ~~Signup up here <QR code>~~
2. Join our webinar (December 2nd)
3. Build your Trusted Application
4. Win Prizes!

Participation Requirements

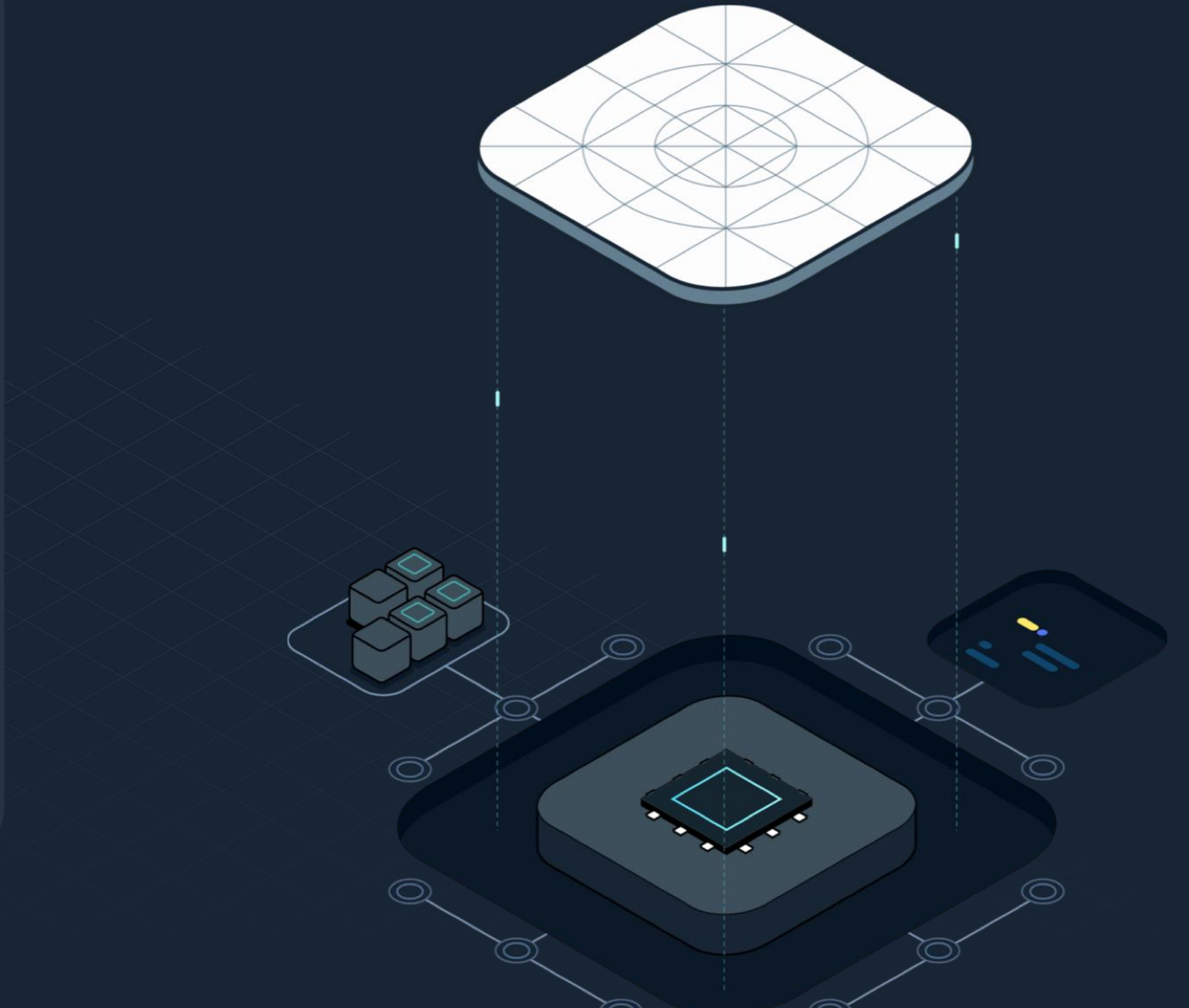
- GitHub Account
- Access to the Licel Hackathon Github Repository - https://github.com/licel/droidcon-hackathon
- All participants must agree to the Rules and Code of Conduct to participate in the Hackathon

Hackathon Rules

- The Hackathon will start on <2nd December 2024 at 6:00 PM UTC> and will end on <13th December 2024 at 11:59 PM UTC>. You will only be able to work on your project during the allotted time
- This Hackathon is exclusive to those who registred during Droidcon London 2024
- You must create a Trusted Application using Java Card Applet technology and integrate it into a mobile application for Android using jCardSim.
- Once the Hackathon begins, you will be able to submit your project. You can submit as many times as you like, the last submission will be considered final
- The intellectual property of your code belongs only to you

How to participate

- Propose a unique idea
- Develop your solution
- Submit it before 13 December 11:59 PM (London Time)



Trusted Application use cases

- Authentication: Enhancing user verification methods
- Payments: Strengthening payment processing security using Trusted Execution Environments
- Data Protection: Safeguarding sensitive information like credentials or personal data
- Other Applications: Any creative and practical use case that showcases the potential of TAs in mobile apps

HOTP Authenticator app

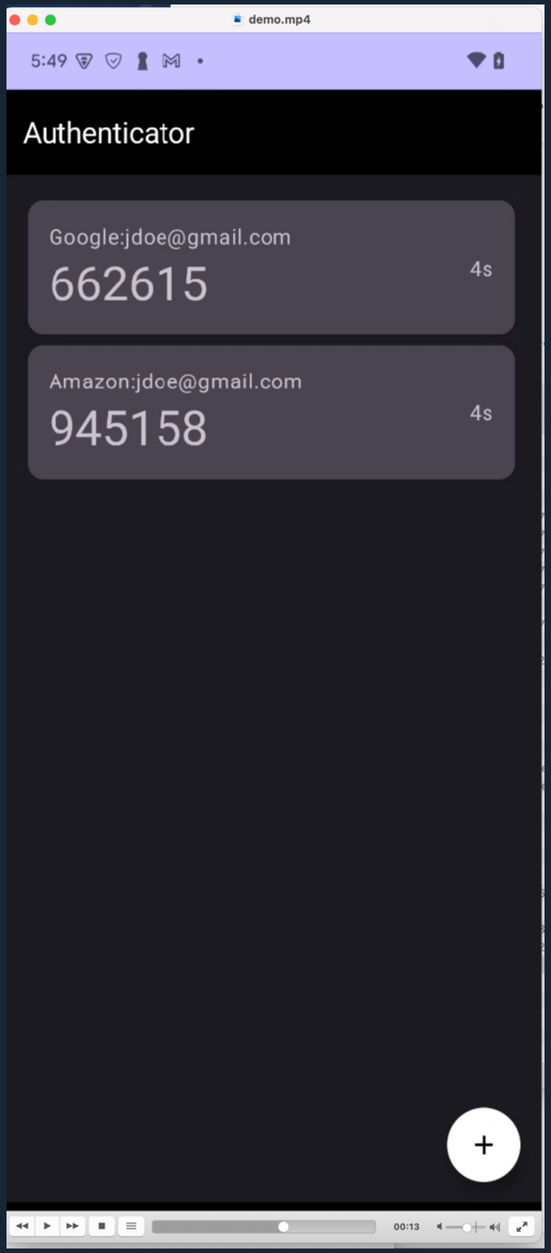
- GitHub Reposiotry

  - https://github.com/licel/hotp-authenticator

- HOTP Generation: Implements the HMAC-based One-Time Password (HOTP) algorithm as defined in RFC 4226.

- Java Card Simulation: The Trusted Application is developed and tested using jCardSim.

- Mobile Integration: The project demonstrates embedding the TA into an Android application for secure authentication workflows.



Authenticator

Google:jdoe@gmail.com
662615 4s

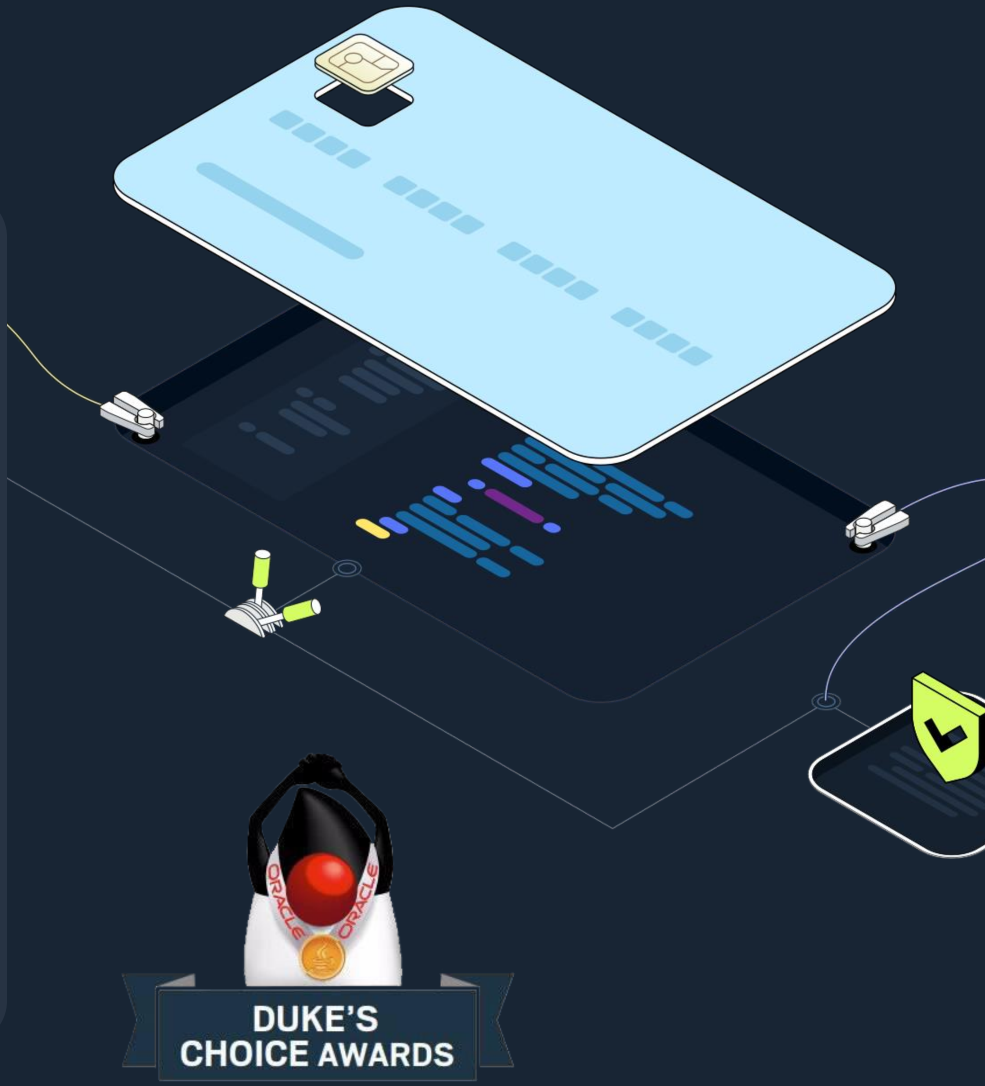
Amazon:jdoe@gmail.com
945158 4s

Develop your Trusted Application with jCardSim

- First public release in December 2011
- jcardsim.org launched in June 2012
- Oracle Duke’s Choice Award 2013 Winner

Functionality

- Full Support for Java Card 3.0.4 (Classic) API
- Rich Simulation API (including vTEE support)
- Cross-platform



jCardSim API Demo

```
// 1. Create simulator
CardSimulator simulator = new CardSimulator();

// 2. Install applet
AID appletAID = AIDUtil.create(”a00000006203010c0101");
simulator.installApplet(appletAID, HelloWorldApplet.class);

// 3. Select applet
simulator.selectApplet(appletAID);

// 4. Send APDU (process)
CommandAPDU commandAPDU = new CommandAPDU(0x80, 0x01, 0x00, 0x00);
ResponseAPDU response = simulator.transmitCommand(commandAPDU);

// 5. Check response status word
assertEquals(0x9000, response.getSW()
```

Java Card Applet Constructor

```
OTPApplet(byte[] bArray, short bOffset, byte bLength) {
    hmacKeys = new HMACKey[HMAC_KEYS_SIZE];
    transientHmacValue = JCSystem.makeTransientByteArray(MAX_ALLOWED_HMAC_SIZE_BYTES,
                                                         JCSystem.CLEAR_ON_DESELECT);
    transientData = JCSystem.makeTransientByteArray(MAX_ALLOWED_DATA_SIZE_BYTES,
                                                    JCSystem.CLEAR_ON_DESELECT);
    register();
}
```

Java Card Applet Process

```
public void process(APDU apdu) throws ISOException {
    byte[] buffer = apdu.getBuffer();
    switch (buffer[ISO7816.OFFSET_INS]) {
        case INS_CALCULATE_OTP:
            calculateOtp(apdu);
            break;
        default:
            ISOException.throwIt(ISO7816.SW_INS_NOT_SUPPORTED);
    }
}
```

Java Card Applet Logic

```
private void calculateOtp(APDU apdu) {
    // parse APDU command
    byte[] buffer = apdu.getBuffer();
    short dataLength = apdu.setIncomingAndReceive();
    byte algorithmType = buffer[ISO7816.OFFSET_P1]; byte hmacKeyIndex = buffer[ISO7816.OFFSET_P2];
    // calculate HOTP
    HMACKey hmacKey = getHmacKey(hmacKeyIndex);
    byte[] hmacValue = hmacValue(hmacKey, getData(apdu, dataLength), dataLength, algorithmType);
    byte[] truncateHash = truncate(hmacValue, (byte) (getByteOutput(algorithmType) - 1));
    byte length = (byte) truncateHash.length;
     // return data
     apdu.setOutgoing();
     apdu.setOutgoingLength(length);
     apdu.sendBytesLong(truncateHash, (short) 0, length);
}
```

Judging criteria

Technical Review

- The project repository must include a brief description of your project idea
- The project must build successfully using the Android Gradle Plugin (AGP)
- The compiled APK must run successfully on an Android Emulator

Jury Evaluation

Each project will be scored between 0 and 5 points based on the following criteria:

- Innovation of the idea
- Implementation quality
- Practicality of the Trusted Application integration
- Overall project value

Check your email
on December 18

Good Luck!