Cothority Mobile: a mobile application to perform distributed tasks using the cothority-framework

Bachelor Project – Spring 2017

Romerio Lucio
Index

- Project Overview and Motivation
- Introduction
- Architecture
- Implementation
- Results Analysis
- Future work
- Demo
Project Overview

- Develop a Mobile Application using a cross-platform framework
- Implement a ProtoBuf library
- Implement a Cryptographic library
Project Motivation

• Existing application uses JSON-interface

• Make application available both for Android and iOS

• Libraries needed to implement Cothority services
Introduction - PhoneGap

• CSS3, HTML5 and JavaScript

• Hybrid application:
  – Access to most of the native APIs
  – Layout rendered via web views

• PhoneGap Build
Introduction - CISC

- Add device to access-control-list
- Make proposition
- Vote proposition
Architecture - Libraries

• Implementation coordinated by Gaylor
  – Cothority ProtoBuf
  – CryptoJS

• Extended to satisfy needs of my application
  – CISC messages
  – hashConfig and SchnorrSign
Architecture - Database

- Need to store:
  - Private keys
  - Conode information
Architecture - Setup

- Create PhoneGap project
- Import libraries
- Sidebar
Implementation - Database

- API described into PhoneGap Documentation
- Traditional SQL statements
- Asynchronous task: handlers
Implementation - Conode Status

• Test device - conodes communication

• Procedure:
  1. Send empty message
  2. Show conode status to the user

<table>
<thead>
<tr>
<th>Name</th>
<th>Conode_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>192.33.210.8</td>
</tr>
<tr>
<td>Connection</td>
<td>tcp</td>
</tr>
<tr>
<td>Port Number</td>
<td>8002</td>
</tr>
<tr>
<td>Uptime</td>
<td>23h45</td>
</tr>
<tr>
<td>Traffic [Bps]</td>
<td>1.36</td>
</tr>
<tr>
<td>Services</td>
<td>CoSi,Guard,Identity,Skipchain,Status</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Implementation - WebSocket

- Reuse same connection
- Send message: asynchronous task
- Use handlers
Implementation - Register Device

1. Scan QR-code encoding conode ID
2. Parse scanned string
3. Create and send proposition
4. Show informative message

```
"threshold": 2,
"device": {
    "icsil1-conode1": {
        "point": /*public key*/
    }
},
"data": {
    "ssh:icsil1-conode1.test1": /*ssh key*/
}
```
Implementation – Key Idea

- One html element for each phase
- Change elements style programmatically
- Wrap related JavaScript in a single file
Implementation – Check Config

• Tab menu

• Phases
  – Conode list
  – Send correct message
  – Display Config
  – Vote (if update)
Implementation - Vote

- Procedure:
  1. Compute Config hash
  2. Sign resulting hash
  3. Send ProposeVote message

- Use the CryptoJS library
Implementation - Register SSH

1. List conodes
2. Send ConfigUpdate
3. Create new key pair
4. Make and vote proposition
5. Show informative message
Results Analysis - Encountered Issues

- React and ES6
- From PoP to CISC
- Server not sending error messages
Results Analysis - Limitations

• Vanilla PhoneGap:
  – Only script tag to import files/libraries
  – No official automated testing tool

• Mobile application:
  – Stuck in case of server error
  – Cannot create SSH keys yet
  – Layout compromise
Future work

• Handle server errors
• Allow user to create SSH keys
• Test PoP messages
• Extend the two libraries
• Extend the application with other services