PV204 - Project

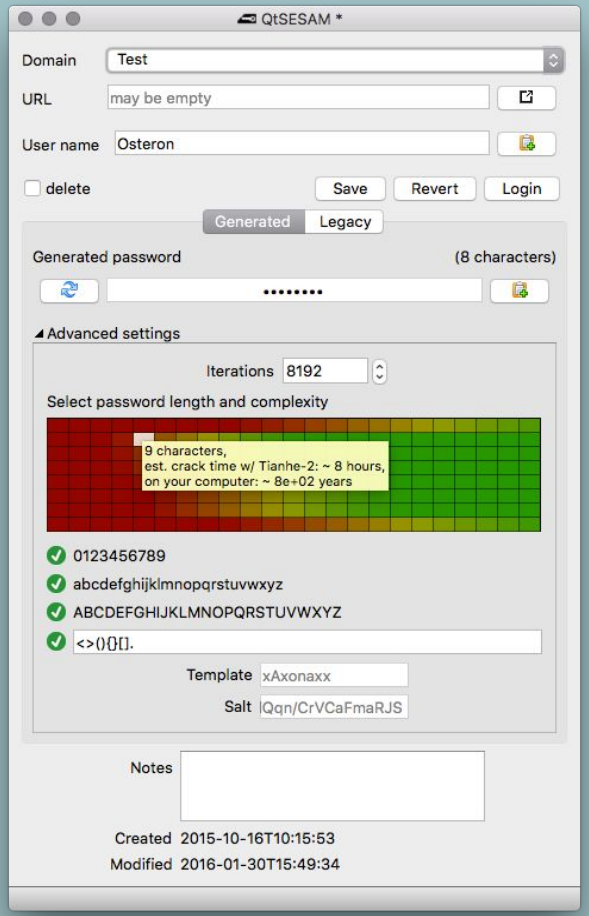
Members: Andrea Turiaková, Michal Hajas, Andrej Staruch

Original application

- https://github.com/ola-ct/Qt-SESAM
- Qt version of c’t SESAM **Password Manager**
- Generate passwords for apps in realtime
- For generating passwords is using custom **user settings**, **salt** and **master password**
- Authentication to app only by master password

Original UI of application

- In very first start user is prompted to enter master password
- User is **only alerted not forced** to have strong password



This is how generating of password looks like

Target scenario

- Add option to **authenticate** user **with JavaCard** - store user’s master password in token
- Communication between app and card must be through secure channel **resistant against attacks** e.g eavesdropping passwords/secrets, man-in-middle-attack ...
- This can **prevent** from **Keylogger attacks** - user enter pin every time, but pin without token is useless.
- There will be possibility to change token in menu after authentication
- Left authentication with master password as **backup option** - in case of card damage or loss

Protocol

- During installation, javacard randomly generates **RSA-1280 keypair**, this key is later used to authenticate javacard during establishing session keys
- The public key of card, is simply send to computer every time, application need it. To prevent man-in-the-middle attack during public key sharing, the user needs to **check**, whether **fingerprint of public key and modulus** is the same as fingerprint, which user obtained during applet installation. Applet installation needs to be done in secure environment (Root of trust).
- Then one-side (only java card is authenticated) authenticated **1024 bits** long **Diffie Hellman** is used to exchange session keys. The configuration is hardcoded for now, but it is easy to generate it randomly, only change is needed on application side.

Protocol

- After diffie hellman we use **SHA-256** as key derivation function for obtaining session key. Communication is then secured with **AES** with **128 bit** long **blocks** and with **session key** obtained by **dh + sha256**. **PKCS7** padding is added manually on both sides (PC, JC).
- When secure channel is successfully established, user can obtain master password from smartcard, but she/he is prompted to input correct PIN.

PIN code restrictions

- Default PIN value is set to value 0000, but user is prompted to change this pin, when she/he is adding token to application. Without changing default pin applet functions cannot be used.
- Functions working with pin or master password in applet require verified PIN

Extension of application code

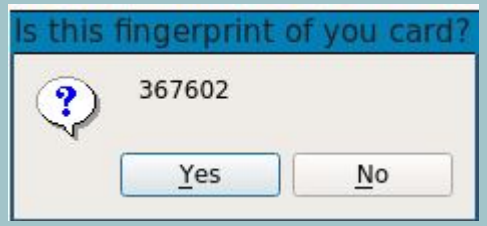
- Available in Qt-SESAM/java_card
- **apdu.cpp** - functions responsible for creating APDU from interface device to card
- **apduresponse.cpp** - functions working with APDU response from card
- **scutils.cpp -** functions responsible for connecting and communicating with card
- **securechanel.cpp -** functions responsible for creating and managing secure channel between card and app

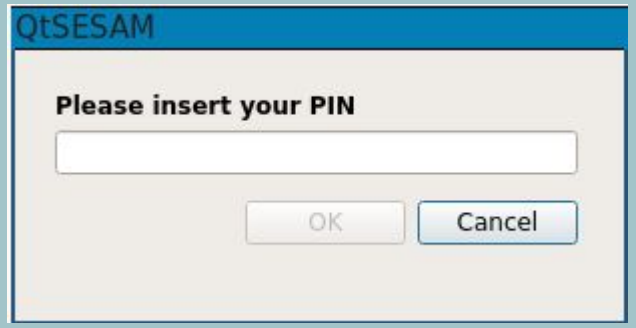
UI changes

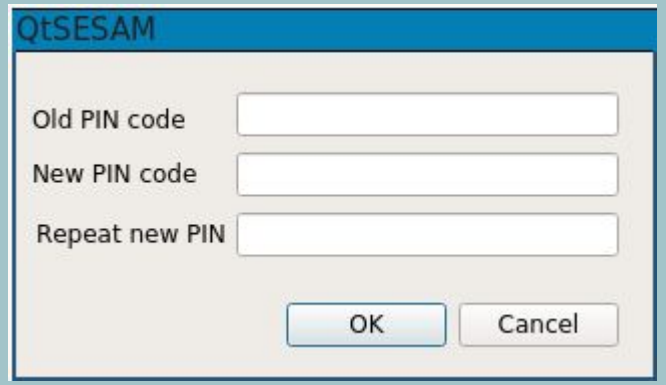

QtSESAM
Enter master password
Password
OK
JavaCard auth


Options
Sync
Password
Backups
Misc
JavaCard
Your JavaCard is connected and ready to use
Remove JavaCard
OK
Cancel

UI changes


Is this fingerprint of you card?
367602
Yes
No


QtSESAM
Please insert your PIN
OK
Cancel


QtSESAM
Old PIN code
New PIN code
Repeat new PIN
OK
Cancel

JavaCard Applet specification

- Available in folder **java_card_application**
- Functions of applet:
  - Get public key/public key modulus
  - Store/get master password (in case secure channel is established and user is authenticated with PIN)
  - Verify/set pin
  - Establish session key using Diffie-Hellman
  - Encrypt/decrypt with session key (in case secure channel is established)