PV204 – Semester Project

Warning! This was project design submitted in first deadline of project. By the time, design changed - see presentation.

Team members:

Andrea Turiaková - 422387, Michal Hajas - 422190, Andrej Staruch - 422437

Open-source application:

c't SESAM Password Manager

https://github.com/ola-ct/Qt-SESAM

Actual situation:

QT-SESAM is an application that enables user generate and manage passwords. Passwords are generated using master password and some service specific settings. Whole password generation is nicely described in documentation page: http://ola-ct.github.io/Qt-SESAM/index.en.html

All settings which was used for creating password are then stored AES-encrypted.

On first start Qt-SESAM asks for user's master password.
This master password is later used for:

1. Generating new passwords for other applications.
2. Only option to get access to generated passwords.

Planned extension:

After our code addition, the app will be working as following:

1. On the first start, Qt-SESAM will ask for user's master password. User can set own password or use offered possibility of really strong password (generated by system call, e.g. `mkpasswd -l 20` - this password user can backup on paper, or on some other safe place). It is not necessary to memorize it, since the user can later use smart card, instead of typing password directly.
2. In the application, there will be menu with card operation: add/delete card.
3. Adding card will be simple copying of public key from the card.
4. Then in following login into the application, user doesn't need to enter master password (strong, secure, 20 chars long, hard to remember) but he/she can login via card. The login consists of following steps:
    a. Application will generate random number (32 bits is enough).
    b. Application encrypts random number by smart card's public key.

   c. Application sends encrypted random number via unsecure channel (attacker can’t decrypt it).
   d. User will be prompted for PIN validation.
   e. After correct PIN validation, smart card will decrypt encrypted random number with private key and sends it back to the application.
   f. If received number is same as original one, user is logged in.
5. There will be possibility to change or delete user card by typing and signing in with the original master password.

Javacard applet:

During installation of applet on card, applet will generate key pair (RSA-2048). Before using card user will be prompted to change default PIN.

Card will be capable of operations:

1. Change PIN
2. Verify PIN
3. Send public key
4. Decrypt data with own private key
5. Send data