

# Project Specification

## Working Title

Lukasz's Password Manager.

## Project Aim

This project is to create a web-based, free and open-source password manager for the privacy focused user.

## Project Expectations

This is a product focused security project. At the end of the project, the code will be made free and open-source, as well as the specification for the container. Making the project open-source will give it the ability to be expanded, more secure and more available. The main expectations are:

- Create a cryptographic container for the password data based on previous works and own research, like Linux LUKS (cryptsetup, 2020). This would allow for:
  - A way to recover the encrypted data without a password by using a recovery phrase.
  - A way to distribute the secret recovery data into pieces, with thresholds by implementing Shamir's scheme (Adi Shamir, 1979).
  - Multiple passwords to be used to open the container.
- Make a password manager that:
  - Implements the container described with all of its features.
  - Implements basic password manager features:
    - Add/Remove/Update passwords.
    - Allow exporting passwords.
    - Change encryption specific settings.
    - Generate secure passwords.
- Make a service that:
  - Provides hosting for the password manager.

## Good practice

Good practice will be shown by using abstraction and solid design patterns as well as unit testing. Version control should also be used as git reverts and git branching are possible to allow better workflow. It also allows room for collaboration in the future.

# Cryptographic Container and Research

The container will be the bulk of the research. The container will be able to recover the encrypted data without a password by using a recovery phrase, generate the secret recovery data into pieces and allow multiple passwords to be used to open the container.

The research will find the best techniques in terms of security and performance to allow the container to exist. Similar approaches like LUKS (cryptsetup, 2020) exist and comparing the container against such approaches will be conducted. The best encryption methods will also be researched and considered.

## Features and Evaluation

Because this is a project focused on security, with a very unique set of features that other standalone password managers don't possess, most of the marking should come from the evaluation of the product's features from a user's standpoint as well as individual anonymity and security of data. Another evaluation metric would be the security and performance of the program.

## Novel Features

For example, ability to safely and securely decode data without a password is an entire novelty when it comes to password managers, as well as designing and implementing a container for all the encrypted data that makes it happen. Further features like the ability to have multiple passwords unlock the encrypted container and shared secret recovery option also should be evaluated by implementation and completion.

The ability to unlock the data without the password could be very beneficial if the main user has a trusted person that would be able to gain access to the data in the container. For example, if a person has a trusted family member, the person can give the recovery token to the trusted family member for them to unlock the container and gain access to the passwords.

The password manager will be able to allow users to sign in with multiple passwords. Multiple passwords would allow multiple users to use the same container for passwords and this would allow multiple users to sync the passwords and account details. This could be used by a company which has multiple employees that have shared accounts on several websites.

Shared secret recovery is a function where access to the container requires a certain amount of pieces of information to unlock. For example, a user can set to create 9 pieces of information and if 6 pieces of information are combined, it would allow for the unlocking of the container containing all of the passwords. The numbers can be adjusted by the user. This approach can be beneficial if the user doesn't want a singular entity to gain access to the container. An example of this in practice could be a person distributing one piece of information to each of the 9 family members and adjusting the number of pieces required to open the container to 6. This would mean that to open the container, any 6 out of 9 members of the family that have a piece of information would be required. These members would have full access to the container and the encrypted data.

## **Standard Features**

Many other features that are usually present in password managers will also exist, such as the ability to export login details to a file, and a password generator.

## Bibliography

Adi Shamir (1979) 'How to Share a Secret', *Massachusetts Institute of Technology*, 22(11), pp. 612–613.

cryptsetup (2020) 'Cryptsetup and LUKS - open-source disk encryption', *What the ...?*, 21 December. Available at: <https://gitlab.com/cryptsetup/cryptsetup> (Accessed: 14 January 2021).